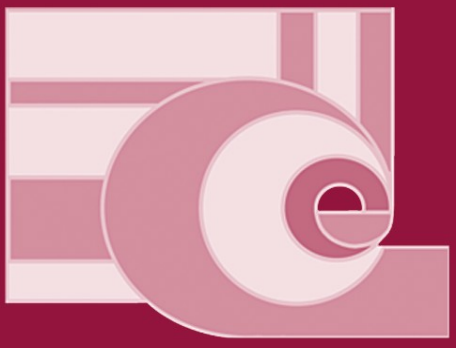
Alexander Gelbukh (Ed.)

Computational Linguistics and Intelligent Text Processing

LNCS 6008

11th International Conference, CICLing 2010,
Iaşi, Romania, March 2010
Proceedings



Springer

Lecture Notes in Computer Science 6008

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

Alexander Gelbukh (Ed.)

Computational Linguistics and Intelligent Text Processing

11th International Conference, CICLing 2010
Iaşi, Romania, March 21-27, 2010
Proceedings

Volume Editor

Alexander Gelbukh
Center for Computing Research
National Polytechnic Institute
Mexico City, 07738, Mexico
E-mail: gelbukh@gelbukh.com

Library of Congress Control Number: 2010922527

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

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

ISSN 0302-9743
ISBN-10 3-642-12115-2 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-12115-9 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper 06/3180

Preface

CICLing 2010 was the 11th 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 volume contains three invited papers and the regular papers accepted for oral presentation at the conference. The papers accepted for poster presentation were published in a special issue of another journal (see information on the website). Since 2001, the proceedings of CICLing conferences have been published in Springer's *Lecture Notes in Computer Science* series, as volumes 2004, 2276, 2588, 2945, 3406, 3878, 4394, 4919, and 5449.

The volume is structured into 12 sections:

- Lexical Resources
- Syntax and Parsing
- Word Sense Disambiguation and Named Entity Recognition
- Semantics and Dialog
- Humor and Emotions
- Machine Translation and Multilingualism
- Information Extraction
- Information Retrieval
- Text Categorization and Classification
- Plagiarism Detection
- Text Summarization
- Speech Generation

The 2010 event received a record high number of submissions in the 11-year history of the CICLing series. A total of 271 papers by 565 authors from 47 countries were submitted for evaluation by the International Program Committee (see Tables 1 and 2). This volume contains revised versions of 61 papers, by 152 authors, selected for oral presentation; the acceptance rate was 23%.

The volume features invited papers by:

- Nicoletta Calzolari, Istituto di Linguistica Computazionale, Italy
- James Pustejovsky, Brandeis University, USA
- Shuly Wintner, University of Haifa, Israel

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

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

Country or region	Authors	Papers[1]		Country or region	Authors	Papers[1]	
	Subm.	Subm.	Accp.		Subm.	Subm.	Accp.
Argentina	6	4.67	0.67	Macao	5	2	1
Australia	6	1.75	–	Mexico	13	7	2.33
Austria	9	2.80	1	Moldova	15	3	1
Belgium	4	2	2	Netherlands	2	1.50	0.50
Brazil	9	4	1	Norway	3	1.17	0.67
Canada	13	7.08	1.75	Pakistan	1	1	–
China	40	16.60	2	Poland	5	4.13	3
Cuba	1	0.25	–	Portugal	13	5.50	–
Czech Rep.	8	4.50	–	Romania	26	14.50	2
Denmark	4	2	–	Russia	11	5	1
Estonia	1	1	–	Saudi Arabia	2	1	–
France	28	13.20	6.20	Slovenia	1	0.50	–
Germany	35	17.67	5.50	Spain	43	17.13	5.88
Greece	4	1.33	1.33	Sweden	7	4.33	1
Hong Kong	12	5.57	1	Taiwan	2	1	–
Hungary	4	1.83	–	Tajikistan	1	0.33	–
India	77	41.50	7.17	Thailand	16	4	–
Indonesia	4	1.50	–	Tunisia	4	1.33	–
Iran	13	8	–	Turkey	14	6.67	1
Israel	2	1	–	Ukraine	2	2.20	–
Italy	32	12.88	3.83	UK	9	4.17	0.83
Japan	11	5.75	3	USA	32	18.67	3.33
Korea (South)	10	5	–	Venezuela	3	1	–
Lithuania	2	2	1	*Total:*	565	271	61

[1] By the number of authors from a country.

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

1st Place: "An Experimental Study on Unsupervised Graph-Based Word Sense Disambiguation," by George Tsatsaronis, Iraklis Varlamis, and Kjetil Nørvåg;

2nd Place: "Cross-Lingual Alignment of FrameNet Annotations Through Hidden Markov Models," by Paolo Annesi and Roberto Basili;

3rd Place: "A Chunk-Driven Bootstrapping Approach to Extracting Translation Patterns," by Lieve Macken and Walter Daelemans;

Student: "Integer Linear Programming for Dutch Sentence Compression," by Jan De Belder and Marie-Francine Moens.

Table 2. Statistics of submissions and accepted papers by topic[2]

Accepted	Submitted	% accepted	Topic
12	52	23	Statistical methods (mathematics)
12	46	26	Machine translation and multilinguism
12	45	27	Text mining
11	50	22	Information extraction
11	40	28	Semantics and discourse
10	27	37	Syntax and chunking (linguistics)
9	59	15	Practical applications
9	54	17	Lexical resources
9	43	21	Clustering and categorization
9	39	23	Information retrieval
8	33	24	Other
7	21	33	Named entity recognition
6	32	19	Acquisition of lexical resources
5	24	21	Symbolic and linguistic methods
5	22	23	Formalisms and knowledge representation
4	20	20	Summarization
4	12	33	Noisy text processing and cleaning
3	24	12	Natural language interfaces
3	22	14	Word sense disambiguation
3	16	19	Morphology
3	13	23	POS tagging
3	12	25	Parsing algorithms (mathematics)
3	9	33	Emotions and humor
3	5	60	Speech processing
2	11	18	Question answering
1	12	8	Text generation
1	9	11	Textual entailment
1	7	14	Spell checking
1	6	17	Cross-language information retrieval
1	4	25	Computational terminology
–	11	–	Opinion mining
–	8	–	Anaphora resolution

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

The authors of the awarded papers 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.

With CICLing 2010—the first CICLing event held in Europe—the computational linguistics community paid tribute to Romania, the nation that gave the world probably the greatest number of wonderful computational linguists per capita—of which quite a few have been CICLing PC members, keynote speakers, or authors: Rada Mihalcea, Daniel Marcu, Vasile Rus, Marius Paşca, Constantin Orăsan, Dan Cristea, Vivi Nastase, Diana Inkpen, Roxana Girju, to name just a few. To further honor Romania and Romanian language (do some

languages cause people to become computational linguists?) the conference was accompanied by a three-day post-conference satellite event, PROMISE 2010: Processing ROmanian in Multilingual, Interoperational and Scalable Environments. The CICLing conference program included two PROMISE keynote talks, presented by Dan Cristea and James Pustejovsky.

With CICLing 2010 we also celebrated 150 years of the Alexandru Ioan Cuza University of Iaşi, the oldest higher education institution in Romania.

Besides its high scientific level, one of the success factors of CICLing conferences is their excellent cultural program. The attendees of the conference had a chance to see the castles of Braşov County (the Bran—or Dracula's—Castle, Peleş, and Pelişor), the painted Orthodox monasteries of Northern Bucovina (Voroneţ, Moldovita, and Humor), as well as the most important sights in Neamt County (the Neamt fortress of Steven the Great, a bison reservation, the Bicaz Canyon, and the Red Lake), among other (see photos on www.CICLing.org).

I would like to thank everyone involved in the organization of this conference and its satellite event, PROMISE 2010. In the first place are the authors of the papers constituting this book: it is the excellence of their research work that gives value to the book and sense to the work of all the other people involved. I thank the Program Committee members and additional reviewers for their hard and very professional work. Very special thanks go to Manuel Vilares and his group, Nicolas Nicolov, Dan Cristea, Ted Pedersen, Yasunari Harada, and Fuji Ren, for their invaluable support in the reviewing process.

I express my most cordial thanks to the members of the local Organizing Committee for their enthusiastic and hard work. I especially thank Corina Forăscu, who amongst her innumerable teaching, scientific, administration, and family obligations always—starting from CICLing 2009—found the time and strength to devote herself to the organization of the conference, frequently at the cost of sleepless nights and lost weekends.

I thank the Faculty of Computer Science (FCS) of the Alexandru Ioan Cuza University of Iaşi (UAIC), Romania, for hosting the conference. With deep gratitude I acknowledge the support of Vasile Işan and Henri Luchian, two of the leaders of the UAIC, whose helpful hand, advice, and support were given unconditionally whenever we needed it. Without the help, support, and guidance kindly offered by Gheorghe Grigoraş, the FCS dean, many aspects of the conference would not have been handled so well and smoothly. We were lucky and happy to collaborate with the Media and Administrative Departments of the UAIC, who provided us with the best conference facilities. The conference would not have been a success without the kind and joyful help of all the student volunteers, especially those from the Association of Students in Informatics in Iaşi (ASII), and all others involved in the organization of the conference. I also greatly appreciate the support of the Romanian Academy, especially concerning the PROMISE conference co-organized under its aegis together with the UAIC.

Very special thanks go to Rada Mihalcea, one of the greatest and oldest friends of CICLing, who, like many times before, guided and helped the organizing team in many activities, behind the scene but always present.

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

February 2010 Alexander Gelbukh

Organization

CICLing 2010 was organized by the Faculty of Computer Science (FCS) of the Alexandru Ioan Cuza University of Iaşi (UAIC), Romania, in collaboration with the Natural Language Processing Laboratory (nlp.cic.ipn.mx) of the Center for Computing Research (CIC) of the National Polytechnic Institute (IPN), Mexico, and the Mexican Society of Artificial Intelligence (SMIA).

Program Chair

Alexander Gelbukh

Program Committee

Eneko Agirre
Sivaji Bandyopadhyay
Roberto Basili
Christian Boitet
Nicoletta Calzolari
Dan Cristea
Alexander Gelbukh
Gregory Grefenstette
Eva Hajičová
Yasunari Harada
Graeme Hirst
Eduard Hovy
Nancy Ide
Diana Inkpen
Alma Kharrat
Adam Kilgarriff
Igor Mel'čuk
Rada Mihalcea
Ruslan Mitkov
Dunja Mladenić
Masaki Murata
Vivi Nastase
Nicolas Nicolov
Kemal Oflazer
Constantin Orasan
Maria Teresa Pazienza
Ted Pedersen
Viktor Pekar
Anselmo Peñas
Stelios Piperidis
James Pustejovsky
Fuji Ren
Fabio Rinaldi
Roracio Rodriguez
Vasile Rus
Franco Salvetti
Serge Sharoff
Grigori Sidorov
Thamar Solorio
Juan Manuel Torres-Moreno
Hans Uszkoreit
Manuel Vilares Ferro
Leo Wanner
Yorick Wilks
Annie Zaenen

Award Committee

Alexander Gelbukh
Eduard Hovy
Rada Mihalcea
Ted Pedersen
Yorick Wiks

Additional Referees

Rodrigo Agerri
Muath Alzghool
Javier Artiles
Bernd Bohnet
Ondřej Bojar
Nadjet Bouayad-Agha
Luka Bradesko
Janez Brank
Julian Brooke
Miranda Chong
Silviu Cucerzan
Lorand Dali
Víctor Manuel Darriba Bilbao
Amitava Das
Dipankar Das
Arantza Díaz de Ilarraza
Kohji Dohsaka
Iustin Dornescu
Asif Ekbal
Santiago Fernández Lanza
Robert Foster
Oana Frunza
René Arnulfo García Hernández
Ana García-Serrano
Byron Georgantopoulos
Chikara Hashimoto
Laura Hasler
William Headden
Maria Husarciuc
Adrian Iftene
Iustina Ilisei
Ikumi Imani
Aminul Islam
Toshiyuki Kanamaru
Fazel Keshtkar
Jason Kessler
Michael Kohlhase
Olga Kolesnikova
Natalia Konstantinova
Valia Kordoni
Hans-Ulrich Krieger
Geert-Jan Kruijff
Yulia Ledeneva
Yang Liu
Oier Lopez de Lacalle
Fernando Magán-Muñoz
Aurora Marsye
Kazuyuki Matsumoto
Alex Moruz
Sudip Kumar Naskar
Peyman Nojoumian
Blaz Novak
Inna Novalija
Tomoko Ohkuma
Bostjan Pajntar
Partha Pakray
Pavel Pecina
Ionut Cristian Pistol
Natalia Ponomareva
Marius Raschip
Luz Rello
Francisco Ribadas
German Rigau
Alvaro Rodrigo
Franco Salvetti
Kepa Sarasola
Gerold Schneider
Marc Schroeder
Yvonne Skalban
Simon Smith
Mohammad G. Sohrab
Tadej Štajner
Sanja Štajner
Jan Strakova
Xiao Sun
Masanori Suzuki
Motoyuki Suzuki
Motoyuki Takaai
Irina Temnikova
Zhi Teng
Nenad Tomasev
Eiji Tomida
Sulema Torres
Mitja Trampus
Diana Trandabat
Stephen Tratz
Yasushi Tsubota
Hiroshi Umemoto

Masao Utiyama
Andrea Varga
Tong Wang
Ye Wu
Keiji Yasuda
Zhang Yi
Daisuke Yokomori
Caixia Yuan
Zdeněk Žabokrtský
Beñat Zapirain
Daniel Zeman
Hendrik Zender

Organizing Committee

Lenuţa Alboaie
Ciprian Amariei
Gheorghiţă Cătălin Bordianu (Student Chair)
Sabin Buraga
Dan Cristea (Honorary Chair)
Corina Forăscu (Chair)
Vlad Rădulescu
Manuel Şubredu
Dan Tufiş (PROMISE Chair)

Web Support:
Alin Alexandru
Ciprian Ciubotariu
Sergiu Hriscu
Diana Pojar
Vlad Săndulescu

Website and Contact

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

Table of Contents

Word Sense Disambiguation and Named Entity Recognition

Best Paper Award – First Place

Semantics and Dialog

Invited Paper

Humor and Emotions

Machine Translation and Multilingualism

Best Paper Award – Third Place

Information Extraction

Information Retrieval

Text Categorization and Classification

Plagiarism Detection

Text Summarization

Best Student Paper Award

Speech Generation

Planning the Future of Language Resources: The Role of the FLaReNet Network

Nicoletta Calzolari and Claudia Soria

CNR-ILC, Pisa, Italy
{glottolo,claudia.soria}@ilc.cnr.it

Abstract. In this paper we analyse the role of Language Resources (LR) and Language Technologies (LT) in today Human Language Technology field and try to speculate on some of the priorities for the next years, from the particular perspective of the FLaReNet project, that has been asked to act as an observatory to assess current status of the field on Language Resources and Technology and to indicate priorities of action for the future.

Keywords: Language Resources and Technology, strategic initiatives, priorities.

1 Why are Strategic Initiatives Necessary?

Language Technologies (LT), together with their backbone, Language Resources (LR), provide an essential support to the challenge of Multilingualism and ICT of the future. The main task of language technologies is to bridge language barriers and to help creating a new environment where information flows smoothly across frontiers and languages, no matter the country, and the language, of origin.

To achieve this, we need to act as a community able to join forces on a set of shared priorities.

Currently, however, the field of LR&Ts suffers from an excess of individuality and fragmentation: there is no substantial sharing of what are the priorities for the field, where to move, not to mention a common timeframe.

This lack of coherent directions is partially also reflected by the difficulty with which fundamental information about LR&Ts is reachable: basically, it is very difficult, if not impossible, to get a clear picture of the current situation of the field in simple terms such as who are the main actors, what are the available development and deployment methods, what are the "best" language resources, what are the areas for which further development and investment would be most necessary, etc. Substantial information is not easily reachable not only for the producers but also for policy makers and funding agencies.

The field is active, but it needs a coherence that can only be provided by sharing common priorities and endeavours. Under this respect, since some time large groups have been advocating the need of a LR&T *infrastructure*, which is increasingly recognised as a necessary step for building on each other achievements, integrating resources and technologies and avoiding dispersed or conflicting efforts. A large

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 1–11, 2010.

range of LRs and LTs is there, but the infrastructure that puts LR&Ts together and sustains them is still largely missing; interoperability of resources, tools, and frameworks has recently come to be understood as perhaps the most pressing current need for language processing research. Infrastructure building is thus indicated by many as the most urgent issue and a way to make the field move forward. Time is ripe for going beyond individual research interests and recognise the infrastructural nature of LRs by establishing an Open Resource Infrastructure (ORI). This will allow easy sharing of data, corpora, language resources and tools that are made interoperable and work seamlessly together, as well as networking of language technology researchers, professionals, users. At the same time, however, this is an endeavour that represents a true cultural turnpoint in the LRs field and therefore needs a careful preparation, both in terms of acceptance by the community and thoughtful investigation of the various technical, organisational and practical aspects implied.

While there has been considerable progress in the last decade, there remains a significant challenge to overcome current fragmentation and imbalance inside the LR&T community. To this end, it is of utmost importance that *strategic activities* are put into place so as to ensure that the LRs community is made aware of the current status of the field, and at the same time so that new directions of development are indicated in a coherent and clear way.

The entire community behind Language Resources (organizations, institutions, funding agencies, companies, and individuals) needs guidance and assistance in planning for and addressing the needs of the language resources and technologies of the future. Together, and under the umbrella of a shared view of actual priorities, a future can be shaped in which a common market for Language Resources and Technologies is created through coordination of programs, actions and activities.

In order to provide such a strategic view of the future directions, a number of preliminary steps are necessary:

- To gather, consolidate and sustain a **community**: LR&T stakeholders need to be identified and convinced that they are part of a larger body.
- To **facilitate interaction** among LR&T stakeholders, so that exchange of opinions and views is ensured
- To promote and sustain **international cooperation**
- To initiate and carry out a community-wide effort to **analyse the sector** of LR&Ts. The analysis should cover along all the relevant dimensions, technical and scientific, but also organisational, economic, political and legal;
- To identify short, medium, and long-term **strategic objectives** and **provide consensual recommendations** in the form of a plan of action targeted to a broad range of stakeholders, from the industrial and scientific community to funding agencies and policy makers.

In this paper we illustrate these steps from the particular standpoint of the FLaReNet[1] project, whose mission is to act as an observatory to assess current status of the field

[1] FLaReNet – Fostering Language Resources Network, www.flarenet.eu– is a Network of Excellence funded under the EU eContent program that aims at developing the needed common vision and fostering a European strategy for consolidating the sector, thus enhancing competitiveness at EU level and worldwide.

on Language Resources and Technology and to indicate priorities of action for the future.

2 An Inventory of Possible Strategic Actions for Language Resources and Technology

2.1 Create and Mobilise a Unified and Committed Community of Language Resources and Technologies players

(Re)creating a network of experts around the notion of Language Resources and Technologies is a challenging task. To this end, FLaReNet is bringing together leading experts of research institutions, academies, companies, consortia, associations, funding agencies, public and private bodies both at European and international level, users and producers alike, with the specific purpose of creating consensus around short, medium and long-term strategic objectives. It is of foremost importance that such a community be composed of the as widest as possible representation of experiences, practices, research lines, industrial and political strategies. This will allow to derive an overall picture of the field of Language Resources and Technologies that is not limited to the European scene, but can also be globally inspired.

In order to constantly increase the community of people involved, as well as to ensure their commitment to the objectives of the Network, FLaReNet runs an active permanent **recruiting campaign**. The FLaReNet Network is open to participation by public and private, research and industrial organizations. Invitation to join, either personal or by means of mailing lists are used in order to enlarge the community as much as possible.

The Network is currently composed of more than 200 individuals and 79 organisations from 31 different countries. FLaReNet affiliates belong to academia, research institutes, industries and government, and their number is steadily enlarging through new subscriptions. Such a community needs to grow not only in number, but also with reference to the type of disciplines involved, from the core ones (Natural Language Processing, computational linguistics, Language Engineering) to "neighboring" ones, such as cognitive science, semantic web, etc. Participants are expected and encouraged to express their views individually as experts but also their organizations views and concerns.

Meetings are the primary means for attracting new members and to reinforce participation of existing ones, but participation is expected and encouraged also by means of online discussions, forum threads, and collaborative documents.

Other general ways for sensitizing and attracting people, as well as for making former members aware of the Network activities, is a massive use of **advertising material, publishing of the Newsletter, and participation in conferences and major events** related to Language Resoruces and Technologies.

Apart from actions for enlarging the FLaReNet community, those aimed at **consolidating** it are especially important. Participants to the community need to feel they belong to a group of people that is actually shaping the field of Language Resources and Technologies by delineating its direction for the next future. The User

Forum, the creation of Thematic Group and "think-tanks" of experts and the launch of closed meetings are the privileged ways for creating close and connected groups of people.

2.2 Define Priorities, Formulate Strategies and Recommendations

Language technologies and language resources are the necessary ingredients for the development of applications that will help bridging language barriers in the global and unified information space, in a variety of means (the Web as well as other communication devices) and for a variety of channels/media (spoken and written language alike but also other associated modalities e.g. gesture). It is of utmost importance, however, to identify a set of priority themes as well as short, medium, and long-term strategic objectives in order to avoid scattered or conflicting efforts. The major players in the field of Language Resources and Technologies need to consensually work together and indicate a clear direction and priorities for the next years, under the form of a roadmap for Language Resources and Technologies. This is the kind of results at which meetings are especially targeted. Actions foreseen to this end are centred around the activity of thematic, general and liaison meetings (see Deliverable 1.4 for further details).

FLaReNet has the challenging goal to act as a "sensor" of current and future trends in Language Resources and Technologies. In order to do this, it must be able to make most pressing issues emerge from its community of players. A number of actions globally converge toward this goal:

- thematic meetings;
- encouragement to propose discussion themes (e.g. through our wiki site);
- requests for topic proposals for Thematic meetings / provoking issues;
- link with major (new) projects & initiatives.

Activities belonging to this category broadly share a common workflow: meetings and events are the privileged places where important issues emerge from the community. These issues are broadly discussed, both at the events themselves and through on-line discussion. Major topics are then distilled and delivered to the community and to the EC under the form of recommendations.

To date, FLaReNet has published two sets of recommendations, the first issued after the FLaReNet Launching Event ("First FLaReNet Forum Highlights"), and the other coming from a consultation of the community. The latter, the "*Blueprint for Actions and Infrastructures*" (D8.2a) gathers the recommendations collected around the many meetings, panels and consultations of the community, as well as the results of the surveying activities carried out under FLaReNet workpackages. The Blueprint encompasses a preliminary Plan for Actions and Infrastructures targeted at HLT players at large, policy-makers and funding agencies.

2.3 Analyse and Survey the LR&T Sector at Large

The definition of a clear and coherent roadmap that identifies priority areas of LRs and LT that need public funding to develop or improve clearly presupposes the availability of an accurate map of Language Resources and Technologies, under many

different respects: the methods and models for production, use, validation, evaluation, distribution of LRs and LTs, their sharing and interoperability; different types and modalities of LRs; the applications and products for LR&Ts; the advantages and limitations of standardisation; the different needs and priorities of academy vs. industry and commerce, of data providers vs. users; the traditional and new areas of interest for LRs; the cultural, economic, societal, political issues, etc.

To this end, FLaReNet is involved in surveying the sector of LR&Ts from many different perspectives. A survey was dedicated to existing language resources and current status of HLT market, mostly from player profile perspective. This survey, which resulted in D2.1, tried to focus on some of the major features that would help understand all issues related to LRs from descriptive metadata to usability in key application, to the composition of various BLARKs for important technologies, to the legal/ethical/privacy issues, etc.

Another study was about the identification of the problems occurring in using language resource and language technology standards and to identify emerging needs for future LRT standards (D4.1). Here, the approach chosen is based on studying existing documents related to LRT standards, to study existing LRT standards, to evaluate current implementations of these standards, to ask implementers about the problems they have identified in using such standards and to ask all LRT stakeholders about missing standards or other problems they see in this respect.

Finally, a survey of automatic production methods for LRs was produced. This comprises a survey of the most demanded resources that are used as the core element of some NLP applications and an overview of the current techniques for automatic construction of LRs. The last academic proposals for automatic acquisition and production of LRs have been also reviewed, in order to confirm the interest that these topics raise in the community of researchers, and as the basic information to start a classification of methods and resources addressed.

2.4 Provide an Assessment of the Current Status of the Field

Work conducted so far in FLaReNet has contributed to draft a first portrait of the current situation in the LR&T sector, in particular for what concerns the types of players and resources (WP2), the various needs for standardisation according to the different communities and the obstructing factors to adoption of standards (WP4), an overview of current practices in evaluation and validation of LR&Ts (WP5), and a review of the innovative methodologies being implemented for the automatic development/processing of LRs (WP6). In addition to the activity of the work packages, input has been collected from a number of events, either organised or co-organised by FLaReNet.

The following is a shortlist of facts that concisely hint at the situation of the LR&T sector as it has emerged from FLaReNet observation.

- Re-use and re-purposing of data is hindered by lack of common data representation
- Documentation of language resources is generally poor
- Clear and easy-to-reach information about resources and related technologies is lacking
- There are too few initiatives around the BLARK concept for European languages
- Little concern is given to the issue of data preservation

- The legal framework is far too complex, and in particular:
  - License models especially suited to LRs are lacking
  - Legal protection modes are different across Europe
  - There are different strata of intellectual property rights
- Sustainability for linguistic tools and language resources needs to be increased
- LRs need to be maintained, in terms of bug reporting, updates and improvements
- More efforts are needed to solve the problem of how to automate the production of the large quantity of resources required, and at the same time how to ensure the necessary quality to get acceptable results in industrial environments
- The evaluation of automatic techniques for LR production is of variable quality. Comparisons among techniques should also be carried out to better assess each of them and their strengths and weaknesses, fostering a greater development in the research on these fields
- Much of the research on automatic acquisition of LRs has focused on small-scale experiments and therefore their usability in applications is largely yet to be demonstrated
- It is very difficult to find information about the characteristics of the language resources that industrial applications use, as well as about the size and granularity of the information contained
- Standardisation is at the core of interoperability. Standardisation issues currently show substantial convergence of opinion and practice, which needs now to be supported to become operational
- LR standards are:
  - too much oriented towards academic/research purposes, not yet mature enough for industrial applications
  - too difficult to understand
  - too abstract, lack concrete examples for implementation, lack of user scenarios or user guides
  - too isolated, existing only on paper but not integratable in digital workflows,
  - too cumbersome to implement, no return on investment in sight for implementers
- Industry-born standards are:
  - too much driven only by specific needs and lack long-term vision
- Given the breadth of current landscape of LR&Ts, a “cultural” change is needed in the sense that there is the need to find ways to monitor how resources are used, to register the resources used or created, to introduce the notion of “publishing” resources and to get academic credit for resources that have been made available.

3 Recommendations for Actions in the HLT Field

The following recommendations are intended both for HLT stakeholders (producers, users and developers of Language Resources and Technologies, both academic and industrial) on the one side and funding agencies and policy makers on the other.

Infrastructure building is the most urgent issue. An Open Resource Infrastructure, which allows easy sharing of language resources and tools that are made interoperable and work seamlessly together, is felt essential. Infrastructures and repositories for tools and language data, but also for information on data (documentation, manuals, metadata, etc.) should be established that are universally and easily accessible by everyone.

3.1 Resource Production and Use

- Provide documentation of the produced resources, covering at least the following aspects (metadata): owner/copyright holder, format and encoding issues of the data and the files, languages(s) covered, domains, intended applications, applications in which the data was used, formal parameters that have to be verified, reliability of any annotation that is included
- For documentation, adherence to practices followed by major data centers is advisable
- Ensure quality of language resources (LRs), for instance by performing a basic quality assessment, to ensure that the minimal critical dimensions are documented: availability/reliability of information on technical, formal issues such as media, number of files, file structure(s), file names etc.
- Annotated resources should be provided with a detailed documentation describing the annotation procedures which have been developed in the annotation process of the LR
- Promote the development of new methodologies for assessing the annotation quality of LRs, in particular for semantic annotation
- Information about whether the resources acquired are actually used or, the other way around, of what are the particular characteristics of the actually used resources, needs to be made public.
- Use of best practices or standards in new projects must be enforced, to facilitate data re-use. Projects developing LRs should be requested to adhere to standards for encoding and representation of data and associated annotations
- Policy makers should enforce documentation of resources, including annotation formats
- Priorities in the development of core LRs for languages should be driven by BLARK-like initiatives: support them and encourage countries to develop their own BLARK matrices
- The creation of LRs must be tied to the development of technologies. It is mandatory to produce the basic tools to process the 'raw' data
- Support the development of LRs for less-resourced languages
- Invest in the production of parallel corpora in multiple languages
- Support the development of resources and technologies for processing non-verbal, and more generally contextual information encompassed in speech-based interaction

- Actual industrial needs have to be to addressed: information about whether the resources acquired are actually used or, the other way around, of what are the particular characteristics of the actually used resources, needs to be made public. The involvement of industries in the research on automatic methods must be supported
- Public procurement, especially at the EU level, should be used as one of the instruments to boost production and adoption of language technologies.

3.2 Interoperability Issues

- It is important that commonly accepted practices (best practices, de-facto standards or standards, when available) are used for the representation and documentation of data
- Not only are data formats to be standardised, but also metadata
- Standards need tools that support them , to promote and ensure their adoption
- LR standards have to be made *more operational* (both, existing ones and those under preparation), with a specific view on different user communities – most users should not or do not want to know that they are using standards, they should operate in the background and they should be "inherent" to the language technology tools or more generic tools they use
- A crucial step towards consistency and interoperability for a global information exchange is the definition of a work environment for *data category definition and management*
- Aim at new forms and manifestations of standards, as *embedded standards*
- For each standard, *return on investment* and possible *motivations* of users should be elaborated together with potential or real users (early adopters)
- Focus in the *short term planning* on those areas where there is enough consensus so that chances are high that a widely accepted standard can be published in a short period of time
- Increase the *acceptance* of LR standards (and the need for them) in different communities, both research and industry communities, and to directly involve user communities in creating standards
- Analyse the needs and requirements for harmonisation of existing standards
- Develop a *strategy* for LR *standards creation*, taking into account aspects such as: bottom-up vs top-down approaches with an interactive process model needed, and modular component standards rather than a single monolithic standard for all of LR
- Standards maintenance should be a process of *change management*, ideally *in real time*
- Inform more pro-actively on best practices in implementing standards and in successful corporate language standards.

- Try to solve the "standard divide" by which a few languages are very well equipped with language resources and consequently with LR standards needed
- Have an *integrative view* on LR standards: an European Interoperability Framework (EIF) for LR has to be developed (cross-domain, cross-purpose, cross-cultural, etc.)
- Contribute to expand the EIF, e.g. in the context of eGovernment, eHealth, eLearning, etc. where many of the existing LR standards can already contribute effectively to enhance data interoperability
- Bring together research communities and industrial application communities for developing a joint vision on LR standards in general
- Foster cooperation between MT industry and CAT-oriented translation and localization industry, for well-balanced and more integrative LR standards industrially usable yet based on pre-normative research
- Develop a broader vision of LR standards with the inherent inclusion of *multimedia*, *multimodal* and *speech* processing applications
- Create an operational ecology of language resource standards that are easily accessible, re-usable, effective, and that contribute to semantic interoperability
- Aim to a global standardization effort on the well-known line of EAGLES-LIRICS-ISO, a long-term strategy which brings together US-experts with their standards and best practices with the European traditions of EAGLES etc. and with East Asian best practices in the field.

3.3 Licensing, Maintenance and Preservation

- Prevent loss of data along the years, by ensuring appropriate means for data archiving and preservation
- Avoid "home-made" licensing models. When drafting a distribution license, carefully think of making it suitable for subsequent re-use and re-distribution of the resource. Adhere to practices used by distribution agencies whenever possible
- Whenever possible, ensure appropriate means for maintenance of Lrs.
- It is important to ensure that publicly funded resources are made publicly available at very fair conditions. Public agencies should impose that resources produced with their financial support are made available free of charge for academic R&D activities. It is also important to encourage language resource owners to donate them to data centres to be distributed free of charge
- Enforce/sustain initiatives for data archiving and preservation: it should be ensured that the data produced by a certain project/initiative/organisation will survive any possible change of media for distribution

- When funding new LRs, funding agencies should request a plan for their maintenance
- Ensure sustainability of funded resources, e.g. by requesting accessibility and usability of resources for a given time frame
- Sustain initiatives offering legal recommendations/guidelines for the reuse of Language Resources, and investigating appropriate licensing models allowing for re-use and re-distribution.

3.4 Evaluation and Validation

- Work on common and standard evaluation procedures, taking into account normalization for comparison. Techniques should not only be evaluated on scientific grounds, but also by their impact in real scenarios of NLP applications
- Develop tools for automatic validation (fault detection (clipping, noise...), detection of segmentation errors, of weak annotations, confidence measures of speech transcriptions)
- Investigate different solutions for addressing the problem of task- vs. application-oriented, such as:
    - A general evaluation framework, including both kinds of evaluation, such as the ISLE Framework for Evaluation in Machine Translation (FEMTI) approach
    - An integrated evaluation platform
    - In the same framework, remote evaluation distributed over the Internet, which permits to interchange components, allowing comparing various approaches, while also examining the influence of the component on the whole system, and which could be organized as Web services.
- Evaluation of the results of automatic techniques must also foresee complex scenarios where the quality of the final results depends on the quality of the partial results.
- The definition of appropriate evaluation frameworks for automatic acquisition methods is needed. The development of evaluation methods that cover the different automatic techniques is fundamental, in order to allow for a better testing of existing and newly discovered methods. Beyond the evaluation on scientific grounds, it is also recommended that techniques are measured by their impact in real scenarios of NLP applications
- Promote a permanent effort framework to take care of language technology evaluation in Europe.

3.5 Directions for Research and Development

- Invest in the development of resources and technologies for processing non-verbal, and more generally contextual information encompassed in speech-based interaction

- As many of the automatic evaluation measures in the style of BLUE and its descendant are still highly controversial, active research into other types of metrics and other ways of evaluating is desirable
- More efforts are needed to solve the problem of how to automate the production of the large quantity of resources required, and at the same time how to ensure the necessary quality to get acceptable results in industrial environments
- Standards need to *co-evolve* at high speed together with rapid change in science, technology, commerce
- Support the involvement of industries in the research on automatic methods, so as to allow a more precise assessment and evaluation of automatic methods for the development of LRs for real-scale applications
- Support transfer of Human Language Technology to SMEs: instruments should be established to transfer language technologies from projects to the SME language technology community in order to stimulate the availability of new technologies and increase the language coverage.
- New languages that joined recently the Union should be considered as a higher priority in coming EU programs
- Human language resources need to be "de-globalized" and focus on local languages and cultures despite today's "global" village
- Copyright law should be harmonised at the European and national level in such a way to permit the free use of copyrighted works for academic purposes
- Favour multidisciplinary integration of different communities.

References

1. Calzolari, N.: Approaches towards a "Lexical Web": the role of Interoperability. In: Ide, N., Fang, A.C. (eds.) ICGL 2008, The First International Conference on Global Interoperability for Language Resources, City University of Hong Kong, pp. 34–42 (2008)
2. Calzolari, N.: Initiatives, Tendencies and Driving Forces for a "Lexical Web" as Part of a "Language Infrastructure". In: Tokunaga, T., Ortega, A. (eds.) LKR 2008. LNCS (LNAI), vol. 4938, pp. 90–105. Springer, Heidelberg (2008)
3. Calzolari, N., Baroni, P., Bel, N., Budin, G., Choukri, K., Goggi, S., Mariani, J., Monachini, M., Odijk, J., Piperidis, S., Quochi, V., Soria, C., Toral, A. (eds.): Proceedings for the FLaReNet Forum, The European Language Resources and Technologies Forum: Shaping the Future of the Multilingual Digital Europe. Istituto di Linguistica Computazionale del CNR, Pisa (2009)
4. Calzolari, N., Soria, C.: The FLaReNet Thematic Network: a Global Forum for Cooperation. In: 7th Workshop on Asian Language Resources in conjunction with ACL-IJCNLP 2009, Singapore (2009)
5. Ide, N., Pustejovsky, J., Calzolari, N., Soria, C.: The SILT and FLaReNet International Collaboration for Interoperability. In: 3rd Linguistic Annotation Workshop in conjunction with ACL-IJCNLP 2009, Singapore (2009)

Cross-Lingual Alignment of FrameNet Annotations through Hidden Markov Models

Paolo Annesi and Roberto Basili

University of Roma Tor Vergata, Roma, Italy
{annesi,basili}@info.uniroma2.it

Abstract. Resources annotated with frame semantic information support the development of robust systems for shallow semantic parsing. Several researches proposed to automatically transfer the semantic information available for English corpora towards other resource-poor languages. In this paper, a semantic transfer approach is proposed based on Hidden Markov Models applied to aligned corpora. The experimental evaluation reported over an English-Italian corpus is successful, achieving 86% of accuracy on average, and improves on the state of the art methods for the same task.

1 Introduction

The development of semantic parsing systems targeted to languages for which annotated corpora, such as FrameNet [1], are not available has a limited and slower development. Statistical learning methods trained over annotated *resources* can not be effectively optimized [2]. For this reason, parallel or aligned corpora are particularly interesting. Annotations for resource-poor languages, e.g. Italian, can in fact be projected out from texts aligned with a language, like English.

In [3], several projection and transfer algorithms are proposed for acquiring monolingual tools from aligned multilingual resources. The study in [4] estimates the degree of syntactic parallelism in dependency relations between English and Chinese. Nevertheless direct correspondence is often too restrictive and syntactic projection yields good enough annotations to train a dependency parser. A bilingual parser that comes with a word translation model is proposed in [5]. In the frame semantics research, Chinese FrameNet is built up in [6] by mapping English FrameNet entries to concepts listed in HowNet[1], an on-line ontology for Chinese, however without exploiting parallel texts.

Recent work explored the possibility of the cross-linguistic transfer of semantic information over bilingual corpora in the development of resources annotated with frame information for different European languages ([7,2,8]). In [2] an annotation projection by inducing FrameNet semantic roles from parallel corpora is presented, where investigation on whether semantic correspondences can be established between the two languages is discussed. The presented methods automatically induce semantic role annotations for a target language whereas a

[1] http://www.keenage.com

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 12–25, 2010.

general framework for semantic projection that can incorporate different knowledge sources is introduced. This work distinguishes predicates alignment from roles alignment, relying on distributional models of lexical association for the first task and on the linguistic information encoded in the syntactic bracketing for the latter one. Results are characterized by higher-precision projections even over noisy input data, typically produced by shallow parsing techniques (e.g. chunking). These approaches have a significant complexity in devising the suitable statistical models that optimize the transfer accuracy. Moreover, they can be effectively used to develop Semantic Role Labeling (SRL) systems in a resource poor language. SRL is first applied to English texts and this makes it possible to label the English portion of a bilingual corpus with a significant accuracy. The large volumes of information can be thus derived, in a relatively cheap way, through cross-language transfer of predicate and role information. A method that avoids complex alignment models to determine more shallow and reusable approaches to semi-supervised SRL has been presented in [9]. It defines a robust transfer method of English annotated sentences within a bilingual corpus. This work exploits the conceptual parallelism provided by FrameNet and a distributional model of frame instance parallelism between sentences, that guarantees a controlled input to the later translations steps. It also employs a unified semantic transfer model for predicate and roles. The result is a light process for semantic transfer in a bilingual corpus. Even if this approach provides a simple process for semantic transfer, it is based on heuristic rules about word alignments and role segmentation.

The aim of this paper is to investigate a more robust method based on statistical principles, namely Hidden Markov Models (HMMs), aiming to map the semantic transfer problem into a sequence labeling task. The objective is to carry out semantic role transfer between aligned texts in a bilingual corpus with a very high accuracy. In section 2, we discuss the Markov model adopted in this study by providing the overview and the formal definitions of the proposed process. The experimental evaluation on a bilingual English-Italian corpus is discussed in Section 3.

2 An Hidden Markov Model of the Semantic Transfer

The semantic transfer task consists in mapping the individual segments of an English sentence expressing semantic roles, i.e. target predicates or Frame Elements [1], into their *aligned* counterparts as found within the corresponding Italian sentence. In Fig.1 an example of a semantic transfer task is shown. In this case the predicate, i.e. the *Lexical Unit* (LU), also called the *target* hereafter, for the frame EVENT) is *happen*. The semantics of the sentence also defines the TIME role, through the segment *after 2006*. The semantic transfer task here is to associate "*happen*" with the verb "*accadrá*" and "*after 2006*" with the fragment "*dopo il 2006*" in the Italian sentence. Given a parallel corpus with the English component labeled according to semantic roles, we aim at detecting, for

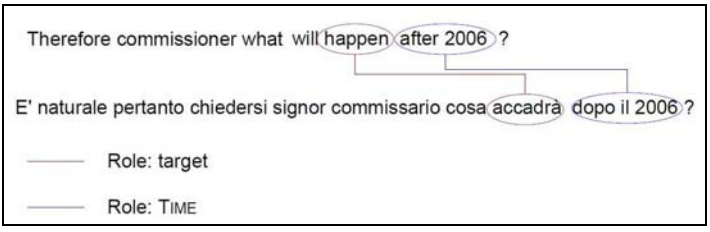


Fig. 1. Cross-language transfer: an example

each segment in an English sentence expressing the role X, the substring in the corresponding Italian sentence that exactly defines X.

For this reason, we will assume hereafter that the English component is always labeled through a SRL software that supplies all the (valid) semantic roles according to FrameNet database.

Let us define an English sentence f as $es_f = (e_1 \dots e_m)$, i.e. a sequence of words e_j, and the corresponding set of indexes j as $EnI = \{1, \dots, m\}$. Analogously, we define an Italian sentence f as the sequence of words $is_f = (w_1 \dots w_n)$, and the set of indexes as $ItI = \{1, \dots, n\}$. Giza [10] is a publicly available machine translation tool based on HMM word alignment models that provides the alignments between individual Italian and English word pairs: these are produced by Giza according to translation probabilities as estimated across an entire bilingual parallel corpus. Notice how all the English words related by Giza with an Italian word can be seen as the emissions of the latter word, given the corpus. Hereafter, the set of the emissions for each $i - th$ Italian word is defined as $E_i = \{e_1, \dots, e_n\}$ whereas every e_j is a possible translation of the Italian word w_i. Now, in the perspective of the semantic role transfer task sketched above, every Italian word can be characterized by one of the following *three* states:

1. It is *inside* the semantic role, as it translates one or more English words that are part of the role, as in the case of *dopo* in the Fig. 1 example
2. It appears *before* any other Italian words translating any part of the semantic role, i.e. it is *out of the role on its left* like *commissario*
3. It appears *after* any other Italian words translating any part of the semantic role, i.e. it is *out of the role on its right* like the token "?".

In this view, the semantic role transfer problem can be always mapped into a sequence labeling task, as for every role in an English sentence we need to tag individual words in the Italian sentence with the three labels corresponding to the above states. In Figure 1, the English words $\{after, 2006\}$ compose the substring of es that defines the TIME semantic role, namely α: this substring will be hereafter denoted by $es(\alpha)$. In analogy with the English case, we will denote by $is(\alpha)$ the analogous substring of the Italian sentence is that expresses the same role α.

Notice that for the translations E_i of an Italian word w_i to be acceptable for any role α they must also appear in the segment $es(\alpha)$. In the example of

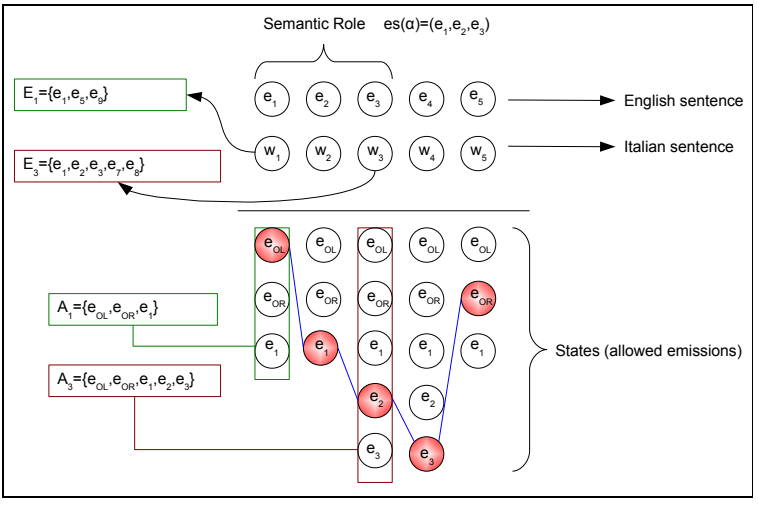


Fig. 2. Cross-language transfer: states and resolution

Fig. 2, the set of words E_i must belong to the set[2] $E_i \cap es(\alpha)$, that defines the useful potential translations of the word w_i for the segment corresponding to the semantic role α. Notice how, in a generic sentence pair (es, is), every translation maps an Italian word $w_i \in is$ into one of its valid translations. The members of the set $E_i \cap es(\alpha)$ can be seen thus as possible state labels referring to individual English words e_k, whenever these latter appear in the English segment $es(\alpha)$ expressing a role α.

On the contrary, whenever an Italian word is not involved in the role α, i.e. it appears *before* or *after* the segment $is(\alpha)$, we will use the alternative tags e_{OL} and e_{OR}. These latter define that the i-th Italian word w_i does not belong to the targeted semantic role $is(\alpha)$. The set of valid labels for every Italian word w_i are thus defined as $A_i = (E_i \cap es(\alpha)) \cup \{e_{\text{OL}}, e_{\text{OR}}\}$. An example of these labels is reported in Figure 2 as columns under every Italian word w_i.

Let us introduce the function $\theta(i)$ that, given is and $es(\alpha)$, couples each Italian word $w_i \in is$ with an English word $e_j \in A_i$, i.e. a possible translation or the special labels e_{OL} or e_{OR}. This function can be defined as follow

$$\theta(i) = j \quad \text{with} \quad e_j \in A_i \tag{1}$$

In Fig. 2, every i-th state is related with an emission in A_i. In this example the Italian word w_1 has its own set of emissions consisting of the English words $\{e_1, e_5, e_9\}$. Notice that as e_5 and e_9 do not belong to the English role subsequence es(α), they are not included in set A_1, that consist only of e_1, e_{OL} and e_{OR} indeed. The darker states define the resolution path that retrieves the Italian semantic role that is the words sequence (w_2, w_3, w_4). On the contrary the

[2] It should noticed here that the sequence $es(\alpha)$ is in fact used as a set, with an odd but still understandable use of the corresponding membership function.

words w_1 and w_5 are labeled as outside the role on the left and on the right, respectively.

The selection of the state sequence as the best labeling for a role $es(\alpha)$ is a *decoding task*: it requires to associate probabilities to all the possible transfer functions $\theta(\cdot)$, so that a transfer can be more likely than another one. Every state $e_j \in A_i$ is tied with the observation of the Italian word w_i: it represents the specific j-th translation as shown in the English sentence es. States $e_j \in A_i$ establish the correspondence between word pairs (w_i, e_j): for all the words $e_j \in es(\alpha)$ the state sequence provides the correspondence with the Italian segment $is(\alpha)$. The resulting HMM, given an Italian sentence is of length n, provides the most likely sequence of states $S = (e^1_{j_1}, e^2_{j_2}, \dots e^n_{j_n})$, whereas every $e^k_{j_k} \in A_k$: S identifies the Italian words w_i whose "translations" are inside the English segment $es(\alpha)$, i.e. $e_{j_i} \notin \{OL, OR\}$. Figure 2 reports an example where $es(\alpha) = (e_1, e_2, e_3)$ and the state sequence suggests $is(\alpha) = (w_2, w_3, w_4)$.

2.1 States, Emissions and Transitions for Semantic Transfer

In order to develop our Markov model of a semantic transfer task, let us discuss it through an example. Given the Italian sentence "*É naturale pertanto chiedersi signor commissario cosa accadrá dopo il 2006?*" and the corresponding English one "*Therefore commissioner what it will happen after 2006?*", we want to transfer the semantic role of TIME represented by the words "*after 2006*". FrameNet define the English sentence as follow

Therefore commissioner what it will ***happen*** [*after 2006* TIME]?

We suppose that this correct labeling is proposed by a existing SRL software and every role label is given as input. In order to analyze the role TIME, Fig. 3 shows how es(TIME) influences the set of possible states A_i for every Italian word.

In Table 1, the emissions supplies by Giza for each Italian word are shown. Therefore each Italian word w_i is associated to a set of English emissions in A_i. Notice how even if many less likely alignments have been neglected in Table 1,

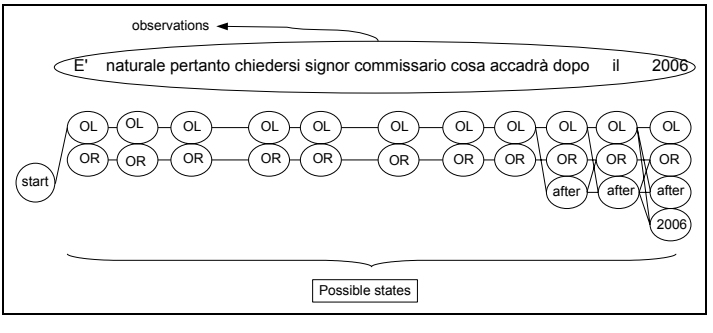


Fig. 3. States concerning the semantic transfer example and possible transition

Table 1. Emissions supplied by Giza for the Italian words concerning the sentence in the example of Fig. 1. Note that the English segment *es*(TIME) is the substring, i.e. "*after 2006*"

Position i	**Italian word** $\mathbf{w}_i$	**English translations** $\mathbf{E}_i$
1	É	*was, this, is, that, therefore, ...*
2	naturale	*water, environmental, quite, natural, ecosystem, ...*
3	pertanto	*conclusion, changed, upon, elapsed, cautious, ...*
4	chiedersi	*know, request, now, days, think, asking, ...*
5	signor	*he, commissioner, echo, barroso, ...*
6	commissario	*frattini, dimas, chile, gentlemen, ...*
7	cosa	*topics, uphold, what, ...*
8	accadrá	*happen, supplied, go, prospects, ...*
9	dopo	*from, had,* ***after****, next, ...*
10	il	*when, basis, until,* ***after****, ...*
11	2006	***2006, after****, vienna, current, period, ...*

the number of candidate translations supplied by Giza is still large. However, the set of the useful word alignments for the role TIME is restricted due to the parallelism between *es*(TIME) and *is*.

In Fig. 3 all the possible states for this specific semantic transfer task are shown. States are those derived from all A_i sets for every observation i. In this way the available states for the first 8 observations are just e_{OL} and e_{OR}, since Giza align all the first 8 Italian words with English words not in the "*after 2006*" segment. Notice that some connections between states are not allowed. The *out right* state is not reachable from the *start* state, as we first have to pass trough an *out left* state or an English word emission state (the latter is not available in this particular case). The *out right* state can not reach an *out left* state obviously. Finally a state with a role English word can not be connected with an *out left* state.

In Fig. 4 all the possible solution paths are shown. The darker path is the selected most likely one. Our task is to derive the best function $\widehat{\theta}(i)$ in terms of its overall probability among all the possible alternative $\theta(i)$. In this example, the best path associates the input English substring "*after 2006*" with the Italian substring "*dopo il 2006*".

Using an Hidden Markov Model for the semantic role transfer task means to define a Bayes inference that consider all the possible state sequences given the observable emissions. Associating a probability to each transfer functions $\theta(i)$ we select the most likely sequence $\widehat{\theta}(i)$ that solve our transfer task as follows:

$$\widehat{\theta}(i) = \underset{\theta(i)}{\operatorname{argmax}} P\big(\theta(i) | es(\alpha), is\big) \tag{2}$$

By applying the Bayes rule to Eq. 2, we reduce it as follows:

$$\widehat{\theta}(i) = \underset{\theta(i)}{\operatorname{argmax}} P\big(is, es(\alpha) \mid \theta(i)\big)\ P\big(\theta(i)\big) \tag{3}$$

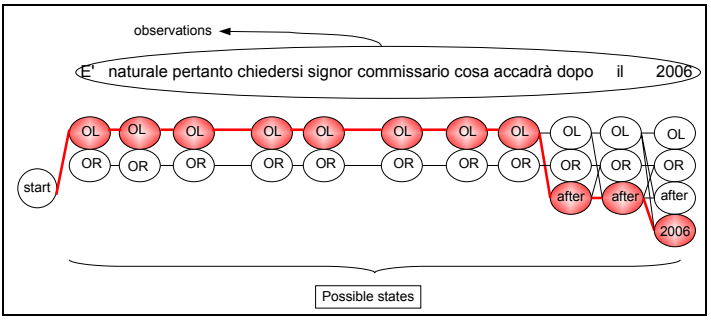


Fig. 4. Possible solutions and best solution

In Eq. 3, we distinguish two probabilities: the left one, $P\big(is, es(\alpha) \mid \theta(i)\big)$, i.e. the *emission* probability, and the right one, $P\big(\theta(i)\big)$, that is the *transition* probability. The emission probability is the probability that links a word $w_i \in is$ with its English counterpart through the selection of the state in A_i. The transition probability is the probability to cross a path between states, i.e. entering into a role and exiting correspondingly after having consumed some valid translations e_j. A first assumption about the emission probability is that the probability of an Italian word depends only on its own emissions. So we can retype this probability as follow

$$P\big(is, es(\alpha) \mid \theta(i)\big) \approx \prod_{i=1}^{n} P(w_i \mid e_{\theta(i)}) \tag{4}$$

in which the emission probabilities do not depend on previous states in a path, so that the product of the emission probability can be used. A second assumption about the transition probability is that the state at step i only depends on the state $i-1$, so that the transition probability is given by

$$P\big(\theta(i)\big) \approx \prod_{i=2}^{n} P\big(\theta(i) \mid \theta(i-1)\big) \tag{5}$$

Finally replacing Equation 4 and 5 into Equation 3, we have

$$\widehat{\theta}(i) \approx \underset{\theta(i)}{\operatorname{argmax}} \prod_{i=1}^{n} P(w_i \mid e_{\theta(i)}) \prod_{i=2}^{n} P\big(\theta(i) \mid \theta(i-1)\big) \tag{6}$$

where the first one is the emission probability and the second one is the transition probability.

Estimating Emission probabilities. The emission probability expressed in Eq. 6 can be retyped using Bayes rule as:

$$P(w_i|e_{\theta(i)}) = \frac{P(e_{\theta(i)}|w_i)\ P(w_i)}{P(e_{\theta(i)})} \tag{7}$$

The probability $P(e_{\theta(i)}|w_i)$ defines the coupling between an Italian word w_i and an English one e_j supplied by the mapping $\theta(i)$. For $j \neq OL$ and $j \neq OR$ this probability is given by Giza. $P(w_i)$ defines the probability to extract w_i randomly from our corpus. Similarly $P(e_{\theta(i)})$ is the probability to extract $e_{\theta(i)}$ randomly from our corpus, that is the English word chosen by our transfer function. Given $P(w_i) = \frac{C(w_i)}{N_{it}}$ where $C(w_i)$ is the function that counts all the w_i occurrences in our corpus and N_{it} is the Italian corpus size, we define $P(w_i) = \frac{C(w_i)+1}{N_{it}+|D_{it}|}$ by applying a smoothing where $|D_{it}|$ is the size of the Italian vocabulary. Analogously, $P(e_{\theta(i)}) = \frac{C(e_{\theta(i)})+1}{N_{en}+|D_{en}|}$. Equation 7 can be thus rewritten as

$$P(w_i|e_{\theta(i)}) = P(e_{\theta(i)}|w_i)\ \frac{C(w_i)+1}{C(e_{\theta(i)})+1}\ \frac{N_{en}+|D_{en}|}{N_{it}+|D_{it}|} \tag{8}$$

in which three emission probabilities, depending on the value of $\theta(i)$ are represented. When an Italian word w_i is part of the semantic role the emission probability of an English word is defined as

$$P(w_i|e_j) = P(e_j|w_i)\ \frac{C(w_i)+1}{C(e_j)+1}\ \frac{N_{en}+|D_{en}|}{N_{it}+|D_{it}|} \tag{9}$$

where $P(e_j|w_i)$ is given by Giza.

When an Italian word is outside a semantic role (i.e. $\theta(i) = OL$ or $\theta(i) = OR$) the corresponding emission is estimated as

$$P(w_i|e_{\text{OL}}) = \frac{\sum_{is}\sum_{\alpha\in is}\delta_{OL}(w_i, is, \alpha)}{\sum_{is}\sum_{\alpha\in is}\sum_{w_i\notin is(\alpha)}\delta_{OL}(w_i, is, \alpha)} \tag{10}$$

whereas the function $\delta_{OL}(w_i, is, \alpha)$ as well is given by

$$\delta_{OL}(w_i, is, \alpha) = \begin{cases} 1 & \text{if } w_i \text{ is on the left of } is(\alpha) \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

Notice that $\delta_{OL}(w_i, is, \alpha)$ counts the occurrences of w_i on the left of a semantic role α and it has a counterpart in the function $\delta_{OR}(w_i, is, \alpha)$ that counts the right co-occurrences.

As for this kind of emission probabilities, we apply smoothing so that Eq. 10 becomes

$$P(w_i|e_{\text{OL}}) = \frac{\left(\sum_{is}\sum_{\alpha\in is}\delta_{OL}(w_i, is, \alpha)\right)+1}{\left(\sum_{is}\sum_{\alpha\in is}\sum_{w_i\notin is(\alpha)}\delta_{OL}(w_i, is, \alpha)\right)+|D_{it}|} \tag{12}$$

Finally, $P(w_i|e_{\text{OL}})$ has its obvious counterpart $P(w_i|e_{\text{OR}})$ for the words w_i on the right of any semantic role.

Estimating transition probabilities. The transition probability constraints the overall likelihood of a path through the Markov model of a semantic transfer task. Every transition depends only on the current, i.e. k-th, state and on the next

$k-1$-th state. The type of a state is defined by the attributes in A_i as defined in Section 1. A transition is determined by the choice of a mapping function $\theta(i)$ that decides how to map an incoming words w_i. $\theta(i)$ clearly depends on the word itself w_i as it characterizes the best possible translations of w_i in the targeted English sentence es. However computing a lexicalized estimate of the probability $P\big(\theta(i) \mid \theta(i-1)\big)$ is problematic as data sparseness would limit the suitable treatment of rare and unseen phenomena (i.e. unigrams and bigrams absent from the training corpus).

The model presented hereafter departs from the hypothesis of a lexical estimate and generalizes it according to the three macro labels (the syntactic states of being before, within or after a semantic role). This gives rise to a lower number of transition types between states and not words, that are depicted in Table 2.

Table 2. Transition probabilities between states. States $\mathbf{e}_i$ (with $i > 0$) characterize transitions from two pairs or Italian words both internal (i.e. members of the sequence) to $is(\alpha)$.

	$\mathbf{e_{i+1}}$	$\mathbf{e_{OL}}$	$\mathbf{e_{OR}}$
$\mathbf{e_i}$	+	0	1
$\mathbf{e_{OL}}$	1	+	0
$\mathbf{e_{OR}}$	0	0	+

Note that the transitions that enter (or exit) in (from) a semantic role (i.e. from the OL state to a word) are only allowed **once** in a legal solution of the semantic transfer task. Other particular transitions are also not allowed, as for example the one from an "out right" position (OR) back to an "out left" one (OL), as it is not possible to restart the role tagging process when it has been already accomplished on the left. The remaining transitions are all allowed several times. The transition probability can be thus defined as follows:

$$P\big(\theta(i) \mid \theta(i-1)\big) = \frac{\frac{C^b_{i,i\text{-}1}}{C^b}}{\frac{C_{i\text{-}1}}{C^b+1}} = \frac{C^b_{i,i\text{-}1}}{C_{i\text{-}1}} \frac{C^b+1}{C^b} \tag{13}$$

where the notation $C^b_{i,i\text{-}1} = C\big(\theta(i) \mid \theta(i-1)\big)$ is used for bigrams and the notation $C_{i\text{-}1} = C\big(\theta(i-1)\big)$ for unigrams, respectively. The counts used in the estimates of Eq. 13 are summarized in Table 3.

Table 3. Counts involved in the different transition probabilities

	$\mathbf{e}_{i+1}$	$\mathbf{e_{OL}}$	$\mathbf{e_{OR}}$
$\mathbf{e_i}$	$C^b_{i,i+1}$	na	$C^b_{i,\mathrm{OR}}$
$\mathbf{e_{OL}}$	$C^b_{\mathrm{OL},i+1}$	$C^b_{\mathrm{OL,OL}}$	na
$\mathbf{e_{OR}}$	na	na	$C^b_{\mathrm{OR,OR}}$

3 Evaluation

In this section the Markov model for the semantic transfer will be evaluated over an English-Italian bilingual parallel corpus. The sentences used for this purpose have been also employed in [9,11]. As reported in Table 5, it consists in a set of 984 sentences split into 788 training sentences and the remaining 196 ones used for testing. The bilingual corpus is an excerpt of the European parliament data [12], available online. More precisely the about 200 sentences employed in testing were annotated in English and Italian according to their own predicates and semantic roles. The sentences do not share all their annotations as they have been manually labeled according to different FrameNet versions. In Table 4 the number of semantic roles manually annotated in both the English and Italian sentences are shown. Basically, all the LUs (i.e. target predicates) are shared between the two corpus, while only half of the frame elements use the same labels. The statistical data used to build up the model are supplied by Giza and computed over the sentences used as the training corpus.

The emission probabilities are computed by Eq. 8 and can be divided in two main classes. The first one is the probability of the translation of an Italian word w_i into an English word e_k, that is part of the known targeted semantic role, $es(\alpha)$. It is estimated as in Eq. 9 in terms of the Giza probabilities. The second one is the probability of an Italian word w_i to be part of segments that are outside $es(\alpha)$. It is computed according to Eq. 12 by estimating counts over the training corpus, whereas observations about Italian words occurring on the left or on the right of a semantic role could be collected.

The transition probabilities are not lexicalized and can be thus computed for every kind of transition within those depicted in Table 2. The model described allows only a unique solution, that is a set of one or more contiguous Italian words expressing a semantic role, highlighted between the two labels *out left* (OL) and *out right* (OR). The system is evaluated according to the usual metrics of precision, recall and F-measure as computed over the involved semantic transfer information. First the partial or perfect matching of the individual role segments is computed: an output segment is partially (or fully) detected if it has a partial (or perfect) overlap with the corresponding segment defined in the oracle. Percentage is obtained as the ratio with respect the set of all targeted segments. Token-based measures are also considered. Token-recall or precision are obtained considering as individual decisions the tokens belonging to the targeted roles. A(n Italian) token labeled by a semantic role α is a true positive iff it is part of the segment for α as defined in the oracle. Similarly, tokens are considered false negatives and positives if they belong only to the oracle or only to the system output. In Table 6 the overall system results are reported. The percentages are referred to the targets and to the semantic role (or FEs in the FrameNet jargon).

Baselines and previous work can be described according to the example shown in Figure 5. The upper part of Fig. 5 represents the word-level alignment as proposed by the Giza tool. The baselines are reported in the bottom part (B) of the figure. The first alignment derives from the Moses alignment ([13]): it selects

Table 4. Semantic roles in the bilingual corpus. The roles in common between English and Italian corpus are those for which labels are identical.

Semantic Roles	English corpus	Italian corpus	In common
Lexical Units	998	984	984
Frame Elements	1713	1661	842
Total	2711	2645	1826

Table 5. The training/test set splitting adopted for the training of the HMM

	Sentences	Semantic Roles	Targets	Frame Elements
Training	788	1438	788	650
Testing	196	388	196	192

Table 6. Accuracy of the role alignment task over the Gold Standard

Model	Perfect Matching (FE only)	Partial Matching (FE only)	Token Precision (FE only)	Token Recall (FE only)	Token F1 (FE only)
baseline	66.88% (28,37%)	71.78% (41,13%)	.7 (.59)	.31 (.14)	.4 (.23)
Cicling09	59% (45.3%)	**80,6**% (**81**%)	.80 (.80)	**.86** (**.87**)	.83 (.84)
HMM system	**60.3**% (**56.7**%)	78,8% (80.2%)	**.86** (**.87**)	.85 (.86)	**.86** (**.87**)

among the partial segments suggested by the Moses phrase-translation tables the maximal segment (i.e. the longest translation of tokens internal to the targeted role $es(\alpha)$). The row named Cicling09 reports the results obtained by the system discussed in [9] over the test set adopted in this work. That system takes as input the Moses phrase-translation tables. It then performs a boundary detection phase, where possibly useful translation subsequences are merged: all the collected Italian segments are here processed and the best boundary is selected. Pruning of some unsuited solutions is then obtained through a post-processing phase. Here the computed boundaries are refined by applying heuristics based on the entire sentence, i.e. according to candidate solutions for all the different semantic roles in a sentence.

In Figure 5 the comparison of the three semantic transfer methods is presented in the last three rows of the B) part. In the example a role α (e.g. THEME) is characterized by $es(\alpha)$=["*the European Union*"]. The third one is the one proposed in this study. As we can see the Moses baseline method, i.e. the second one, suggests the maximum boundary among those proposed by the alignment shown in A). The Cicling09 system, i.e. the first one, refines this result by applying some heuristics. Although more precise, it does not retrieve all the tokens. The HMM system retrieves all tokens without relying on post processing. The Viterbi algorithm in fact is applied for each role to the entire sentence. The resulting

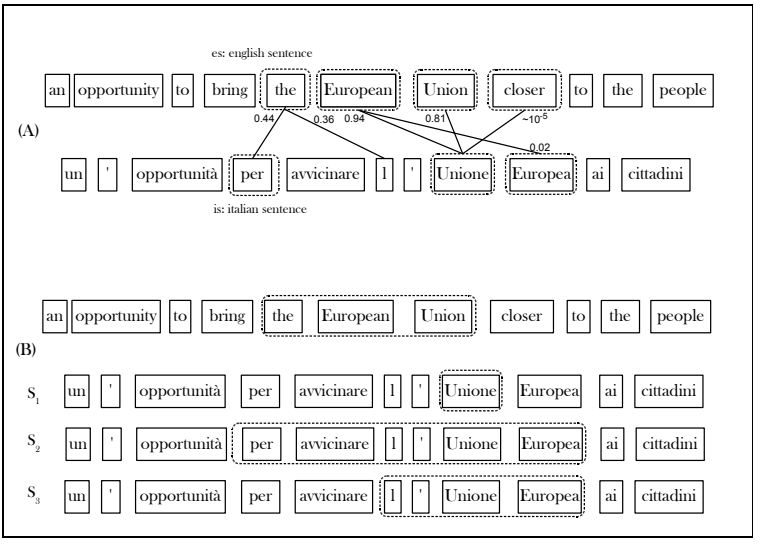


Fig. 5. Comparison among three semantic transfer methods based on the Moses alignments shown in A). The results over the argument "*the European Union*" are shown in the last row of the B) part, as compared with the Moses baseline and the Cicling09 system (first and second row in B), respectively).

best path through the trellis states receive all the syntagmatic constraints from the available translations and from the transitions that characterize the closed labeling as shown in Fig. 4. This method is thus more robust achieving higher precision scores without post processing.

Results in Table 6 show that the HHM defined in this paper produces an improvement in precision and recall over the previously proposed methods. Although the percentage of partially matched roles is in line with the Cicling09 system (i.e. 78.8% vs.80.2%), the perfectly matched phenomena are many more. At the Frame Element level (i.e. over the set of most complex phenomena) a striking performance increase is obtained, that raise accuracy from 45% to 56% (i.e. about 25% increment). Results are also very good for what concerns token-based measures, this suggesting that the HMM-based model approximates the perfect labeling in a much more accurate way. It is worth noticing that Cicling09 results reach a good level of token recall (i.e. ∼86%) at the expense of a lower precision (80%), while the precision reachable by the HMM-based system is higher (∼87%) with a corresponding 5% increase in token F1. In Table 6, results reported in brackets refer to the set of frame elements. These are more complex as for their length and grammatical structure (target predicates are usually expressed by one-token segments, e.g. simple verbs or nouns). On this phenomena the HMM-based system is almost always better with more stable and precise labeling. A further error analysis was carried out in line with [14]. It showed that the most frequent mistakes in the HMM output are basically due to missing/wrong Translations of Target-Words. In the adopted test set in

fact no Double Annotation nor Unexpressed Frame information were found as for its original design. In general, the HMM-based system is more robust and independent from complex heuristics that characterize instead previous works.

4 Conclusions

Unsupervised models for semantic role transfer in bilingual corpora have been recently presented in [9]. Improving these models means making the boundary detection algorithms more robust and introducing new grammatical and syntactic rules. However, this research line may also lead to weaker models that may be not fully applicable to real cases. In this work, an Hidden Markov Model is introduced in order to increase robustness and generalize the semantic transfer system to a larger set of phenomena. First of all, the model should not depend too much on the language pair, in order for it to be adopted in a larger set of cases (i.e. generic semantic role transfer tasks between any language pair and aligned corpus). The model strictly relies on the Giza statistical alignment capabilities and on emission and transition probability estimates, robustly derived from a small corpus. Each sentence is mapped into his own model where semantic roles in the target language are states and the source roles are the observations. Models are just solved at a statistical level (i.e. multiple applications of Viterbi decoding, one for each role to be detected): no rule-based boundary detection or post processing is applied.

The proposed supervised model has been shown to be trainable using a small corpus. In this paper, on the set of 984 sentences taken from European Parliament corpus, an 80% was used for the training phase. Results obtained on the remaining 20% of the sentences allowed to compare the proposed HMM-based approach with the unsupervised system described in [9]. The increase in token-based precision confirms the superiority of the new method. Although they are relative to a subset of the European Parliament used for the evaluation in [9], they are representative of a large set of lexical and grammatical phenomena. Wider experimentation is already in progress to confirm these results over the entire European Parliament corpus: the performance of a Semantic Role Labeling system will be used as an indirect measure of the quality reachable by the approach here proposed.

References

1. Baker, C.F., Fillmore, C.J., Lowe, J.B.: The Berkeley FrameNet project. In: Proc. of COLING-ACL 1998, pp. 86–90 (1998)
2. Pado, S.: Cross-lingual annotation projection models for role-semantic information. PhD Thesis Dissertation, University of Saarlandes, Saarbrucken, Germany (2007)
3. Yarowsky, D., Ngai, G., Wicentowski, R.: Inducing multilingual text analysis tools via robust projection across aligned corpora. In: HLT 2001: Proceedings of the first international conference on Human language technology research, Morristown, NJ, USA, pp. 1–8. Association for Computational Linguistics (2001)

4. Hwa, R., Resnik, P., Weinberg, A., Kolak, O.: Evaluating translational correspondence using annotation projection. In: Proceedings of the 40th Annual Meeting of the ACL, pp. 392–399 (2002)
5. Smith, D.A., Smith, N.A.: Bilingual parsing with factored estimation: Using english to parse korean. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 49–54 (2004)
6. Fung, P., Chen, B.: Biframenet: Bilingual frame semantics resource construction by cross-lingual induction, pp. 931–937 (2004)
7. Pado, S., Pitel, G.: Annotation precise du francais en semantique de roles par projection cross-linguistique. In: Proc. of TALN 2007, Toulouse, France (2007)
8. Tonelli, S., Pianta, E.: Frame information transfer from english to italian. In: Proc. of LREC Conference, Marrakech, Marocco (2008)
9. Basili, R., Cao, D.D., Croce, D., Coppola, B., Moschitti, A.: Cross-language frame semantics transfer in bilingual corpora. In: Gelbukh, A. (ed.) CICLing 2009. LNCS, vol. 5449, pp. 332–345. Springer, Heidelberg (2009)
10. Och, F.J., Ney, H.: A systematic comparison of various statistical alignment models. Computational Linguistics 29, 19–51 (2003)
11. Tonelli, S., Pianta, E.: Three issues in cross-language frame information transfer. In: Proceedings of the RANLP 2009 (2009)
12. Koehn, P.: Europarl: A parallel corpus for statistical machine translation. In: Proc. of the MT Summit, Phuket, Thailand (2005)
13. 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: Annual Meeting of the Association for Computational Linguistics (ACL), Demonstration Session, Prague, Czech Republic (2007)
14. Trandabăţ, D., Husarciuc, M.: Romanian semantic role resource. In: Proc. of the 6th LREC Conference (LREC 2008), Marrakech, Morocco (2008)

On the Automatic Generation of Intermediate Logic Forms for WordNet Glosses

Rodrigo Agerri[1] and Anselmo Peñas[2]

[1] Universidad Politécnica de Madrid (UPM),
Vicomtech Research Centre*
Donostia-San Sebastián, Spain
ragerri@vicomtech.org
[2] NLP & IR Group UNED
Madrid, Spain
anselmo@lsi.uned.es

Abstract. This paper presents an automatically generated Intermediate Logic Form of WordNet's glosses. Our proposed logic form includes neo-Davidsonian reification in a simple and flat syntax close to natural language. We offer a comparison with other semantic representations such as those provided by Hobbs and Extended WordNet. The Intermediate Logic Forms are straightforwardly obtained from the output of a pipeline consisting of a part-of-speech tagger, a dependency parser and our own Intermediate Logic Form generator (all freely available tools). We apply the pipeline to the glosses of WordNet 3.0 to obtain a lexical resource ready to be used as knowledge base or resource for a variety of tasks involving some kind of semantic inference. We present a qualitative evaluation of the resource and discuss its possible application in Natural Language Understanding.

1 Introduction

Ongoing work on text understanding has made clear the need of readily available knowledge and lexical resources that would help systems to perform tasks that involve some type of semantic inference (e.g., [1,2,3]). For example, 21 of 26 teams participating in PASCAL RTE-3 [4] used WordNet as a knowledge resource to support reasoning. It has also been pointed out that we may need to develop deep language understanding techniques if we are to consistently obtain very high performance results in tasks such as RTE [5]. Some work has therefore been done trying to improve the utility of WordNet (notably [6,7]) for semantic inference, by augmenting it with syntactic analysis and logic formalisation of its glosses. This paper reviews previous work aiming to identify those points which could be improved. The result is the development of a new freely available resource consisting of the generation of Intermediate Logic Forms (ILFs) for WordNet 3.0 glosses: ILF-WN.[1]

* Currently at Vicomtech, http://www.vicomtech.org
[1] Freely available to download at http://nlp.uned.es/semantics/ilf/ilf.html

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 26–37, 2010.

The 'intermediate' character of ILF comes from the fact that rather than generating a semantic representation in first-order logic (or other type of standard logic), we provide a formal representation that aims to be as close as possible to natural language by performing strict neo-Davidsonian reification [8,9] and reducing to a minimum the syntax complexity. The objective is to provide a flat, syntactically simple formal representation suitable to perform various types of semantic inference (e.g., as in Recognizing Textual Entailment [1]), avoiding the excessive brittleness caused by first-order approaches, as well as being able to tackle difficult semantic problems such as co-reference, anaphora resolution, etc.

Our representation is based on two main predicates, one denoting the existence of a discourse entity *e(Id1,x1)*, and another to identify the existence of a direct relation between two discourse entities *rel(Id1,Id2,x1,x2)*. Both, entities and relations are indexed to easily add semantic information related to the discourse entities (e.g. lexical information: *w(Id1,Word:Pos:Cat), syn(Id1,Synset-offset)*), but also to to the relations (e.g. syntactic dependency types *dep(Id1,Id2,nsubj)*, semantic roles, etc.) in a structure suitable to treat discourse-related problems. For example, co-reference is denoted by the unification of variables in two different discourse entities (e.g. *e(Id1,x1), e(Id3,x1)*).

Next section discusses previous related work. Section 3 describes the main characteristics of Intermediate Logic Forms. Section 4 describes the development of ILF-WN. A qualitative comparison or evaluation with respect to previous approaches to formalise WordNet's glosses can be found in section 5 and section 6 concludes and points out to any future improvements to ILF-WN.

2 Previous Related Work

Our proposal, both the logic forms and the formalization of WordNet's glosses is inspired by neo-Davidsonian formalisms used in computational semantics such as [10,11,12]. However, ILF-WN is a flat and simple syntax closer to the output of dependency parsers. The syntax also contemplates that every relation between words is a predicate instead of introducing first-order logical operators. Two approaches have previously offered a logic form of WordNet's glosses.

2.1 Extended WordNet

Extended WordNet 2.0-1.1 (XWN 2) provides a logical form and sense disambiguation for the glosses of WordNet 2.0 [13,6,12]. A very important feature of XWN is the expansion of WordNet's relations by taking into account the disambiguated info they extract from the glosses. This is something that current version of ILF-WN does not offer. The overall procedure of building XWN 2 consists of pre-processing the glosses and perform syntactic parsing, logical form transformation and word sense disambiguation (WSD) of adjectives, adverbs, verbs and nouns. They use various methods to perform WSD on the glosses. They disambiguate 64% words of WordNet glosses with 75% accuracy. The rest of the words are tagged with the first sense.

The pre-processing of glosses aims to include the *definiendum* in the *definiens* adding several other terms to make glosses more suitable for syntactic parsing. For example, "the adjective glosses were extended with the adjective and 'is something' in front of the gloss and a period at the end of it" [6]. Take the adjective 'bigheaded': The gloss was transformed from (1) "used colloquially of one who is overly conceited or arrogant; "a snotty little scion of a degenerate family"-Laurent LeSage; "they're snobs–stuck-up and uppity and persnickety", to (2) "bigheaded is something used colloquially of one who is overly conceited or arrogant". The pre-processed gloss is then parsed using Charniak's parser [14] and an in-house parser [6], and its result in a treebank form is included.

The parse results were classified into GOLD (parses manually checked), SILVER (agreement between the two parsers without human intervention) and NORMAL (disagreement between parsers, in-house parser is given priority) qualities (but for formatting reasons, the last element of the tree should be indented to the right of 'VBZ is').

The transformation to logical form [12] is inspired by the eventuality logic proposed by Hobbs [10]. Depending of the part-of-speech of the synset, they use a number of rules for the assignment of variables. In the case of adjectives, "the first word representing the synset is taken and assigned the argument 'x1'. In the gloss of the synset on the right hand size, the argument 'x1' refers to the same entity as the one described by the first word in the synset."

The glosses are included in their original format in English whereas both the parse (Example and logic form (Example 1) elements are performed on the pre-processed versions of the glosses. Furthermore, in the original glosses synsets' definitions, examples and other information is offered as a unit.

Example 1. Logic Form of "bigheaded.s.01" in XWN 2.

```
<lft quality="SILVER">
 bigheaded:JJ(x1) -> use:VB(e1, x6, x1) colloquially:RB(e2)
 of:IN(e1, e2) one:JJ(x3) be:VB(e2, x1) overly:RB(x4)
 conceited:JJ(x4) arrogant:JJ(x4)
</lft>
```

Perhaps due to the glosses pre-processing, some parses result in overly complex structures where in most cases the most important part of the gloss namely, the *definiens*, is buried among a number of subordinating clauses with respect to the phrase 'overly conceited or arrogant'. This problem is fairly frequent for long glosses (usually in nouns), and it seems to degrade the quality of the final logic form. Leaving aside issues such the inclusion of the *definiendum* in the *definiens*, we can see in example 1 that there are variables that do not belong to anything (e.g., *x6*), and others that are left free (not related in any way with the rest of the formula), such as *overly:RB(x4) conceited:JJ(x4) arrogant:JJ(x4)*. Other issues related to the absence of the coordinating disjunction 'or' in the logic form and the assignment of the same variable *x4* for 'overly', 'conceited' and

'arrogant', renders some glosses' logic forms of XWN 2 difficult to understand and use.

2.2 ISI/Boeing WN30lfs

A second project, WN30-lfs, consists of the logical forms for the glosses of WordNet 3.0, except where the parsing failed [7], in XML format, using eventuality notation [10]. It was generated by USC/ISI [7,15]. Every synset is an element consisting of the gloss (without examples, etc.) and its logical form. They preprocessed the glosses to obtain sentences of the form "word is gloss". They parsed them using the Charniak parser [14], and the parse tree is then converted into a logical form by a tool called LFToolkit, developed by Nishit Rathod. In LFToolkit, lexical items are translated into logical clauses involving variables. Finally, as syntactic relations are recognized, variables in the constituents are unified [7]. Furthermore, predicates are assigned word senses using the WordNet semantically annotated gloss corpus [16]. Example 2 shows the logical form for the gloss of *bigheaded*.

Example 2. Logic Form in "bigheaded.s.01" in WN30-lfs.

```
<entry word="bigheaded#a#1" status="partial">
  <gloss>used colloquially of one who is overly conceited or
  arrogant</gloss>
  <lf>bigheaded#a#1'(e0,x0) -> colloquially#r#1'(e5 ,e4) +
     of'(e9,x11,x12) + one'(e11,x12) + excessively#r#1'(e8,e10)
     + conceited#a#1/arrogant#a#1'(e10,x10)</lf>
    <sublf>conceited#a#1'(e10,x10) ->
    conceited#a#1/arrogant#a#1'(e,x10)</sublf>
    <sublf>arrogant#a#1'(e10,x10) ->
    conceited#a#1/arrogant#a#1'(e,x10)</sublf>
</entry>
```

WN30-lfs also includes the sense to be defined in the definition (as in XWN 2) linked by a (seemingly) first-order conditional operator (see Example 2). Furthermore, it is difficult to understand the fact that the logical forms of WN30-lfs often contain free variables and/or predicates without any relation with any other predicates in the definition. As in XWN 2, the predicates for the phrase *overly conceited or arrogant* in Example 2 are left isolated from the rest of the definition.

Summarizing, inspired by XWN 2 and WN30-lfs and acknowledging the many merits of both XWN 2 and WN30-lfs, we believe that there is still some need for providing lexical and/or knowledge resources suitable for computational semantics tasks that required formalized knowledge. In particular, we aim at providing a simple, clear and easy to use logical forms for WordNet's glosses. We also aim at making as transparent as possible the steps taken to obtain the logical forms

from the original glosses, and how this information can be offered in a XML structured resource: ILF-WN (Intermediate Logic Form for WordNet glosses).

3 ILF Representation

Our representation consists mainly of two main predicates, one denoting the existence of a discourse entity, and another establishing a direct relation between two discourse entities. Entities and relations are indexed to easily add semantic information related to the discourse entities. In the following subsections we explain this representation in more detail.

3.1 Discourse Entities

Each word introduce a discourse referent denoted by a variable. This variable, together with its index conform the predicate for discourse entities, *e(Id,x)*. The word itself is only a single piece of information associated to the discourse entity among other information obtained during the linguistic processing (e.g. part of speech, lemma, sense, offset in a ontology, similar words, etc). In ILF-WN, we illustrate this with two predicates for lexical information: *w(Id,Word:Pos:Cat), syn(Id,Synset-offset)* (the latter only for monosemous words).

It can be seen that indexes are important to link the lexical information associated to a word with the role of that word in discourse, independently of the variable unification that further reference resolution may produce. In this sense, two discourse entities that denote the same referent will be expressed as *e(Id1,x),e(Id2,x)*. For example, consider the following text from TAC RTE 2009 testset:

> The disappearance of York University chef Claudia Lawrence is now being treated as suspected murder, North Yorkshire Police said. However detectives said **they** had not found any proof that **the 35-year-old**, who went missing on 18 March, was dead. **Her** father Peter Lawrence made a direct appeal to **his** daughter to contact **him** five weeks after **she** disappeared. **His** plea came at a news conference held shortly after a 10,000 reward was offered to help find Miss Lawrence. Crimestoppers said the sum **they** were offering was significantly higher than usual because of public interest in the case.

The pronouns that need to be resolved are highlighted. Using Lingpipe's coreference system [17], we first identify 'Claudia Lawrence' as a named entity of type PERSON in sentence 1, and then link the female pronoun 'her' in the third sentence to 'Claudia Lawrence' in sentence 1 by assigning them the same reference identifier (*refid*):

```
<coref system="Lingpipe 3.8.1">
<namedent="2" refid="1" s_id="1" type="PERSON" w_ind="6" />
<namedent="4" refid="1" s_id="3" type="FEMALE_PRO" w_ind="1" />
</coref>
```

The co-reference expressions are easily included in the ILFs:

w(1,6,'Claudia_Lawrence','n','nnp') e(1,6,S1_6) w(3,1,'she','prp$','prp$') s(3,1,S1_6)

When the pronoun 'her' (3,1) is resolved to a previous named entity (1,6), then it gets assigned the same variable, namely, word 1 in sentence 3 (her) gets the same discourse referent as word 6 in sentence 1 (Claudia Lawrence). This applies to any subsequent pronouns that are linked to the same entity.

3.2 Relations between Discourse Entities

Strict Davidsonian reification allows us to greatly simplify the syntax of ILF. The relations between entities are introduced by the dependencies obtained by the dependency parser [18]. The predicate *rel(Id1,Id2,x,y)* captures this relation. The pair *Id1,Id2* indexes the relation, preserving the governor-dependent structure of *Id1* with respect to the entity associated to *Id2*. By assigning an index to the relation, we can associate to it any information it might be required (e.g. dependency type, preposition sense, type of noun-noun relation, etc.).

The representation of *buy(x,y)* become *e(Id1,e), rel(Id1,id2,e,x), rel(Id1,id3, e,y)* in our notation. We can then add lexical information, dependency types and semantic roles to the ILF: *w(Id1, Buy), syn(Id1, Syn), dep(Id1,id2,nsubj), dep(Id1,id3,dobj), srl(Id1,id2,Buyer), srl(Id1,id3,Bought).*

In the current version of ILF-WN only the Stanford dependency types are considered, and we include them in the *rel* predicate for simplicity.

4 ILFs for WordNet 3.0 Glosses

4.1 Processing Pipeline

We have assembled a pipeline consisting of a gloss preprocessing module, the C&C tokenizer [19], part-of-speech CRFTagger [20], the Stanford dependency parser [18], and our own ILF generator. ILFs are generated from the dependency parser output adding extra semantic information (if available). The pipeline can take a sentence or discourse in English as an input and automatically generate its ILF. Each third-party tool included in the pipeline is used off-the-self.

Gloss pre-processing. A pre-processing of the glosses was performed in order to obtain grammatically sound sentences more suitable for tokenization, part-of-speech (POS) tagging and syntactic parsing. The pre-processing is loosely inspired in [13]:

1. Text between brackets is removed. Text between brackets is usually an explanation related to the use of the sense defined by the gloss. For example, the gloss of the synset '*bigheaded.s.01*' reads "(used colloquially) overly conceited or arrogant."

2. Everything after a semicolon is removed: Text after the semicolon is usually a semi-structured phrase which does not add anything new to the definition itself. For example, the synset '*concrete.a.01*' is defined as "capable of being perceived by the senses; not abstract or imaginary."
3. According to POS category:
   (a) For nouns and adverbs, we capitalize the first word and add a period at the end. For example, the gloss of the noun '*entity.n.01*' is "That which is perceived or known or inferred to have its own distinct existence."
   (b) For the adjective glosses, 'Something' is added it the beginning and a period at the end of the gloss. The gloss of '*bigheaded.s.01*' mentioned above now reads "Something overly conceited or arrogant." whereas the definition of '*concrete.a.01*' has been transformed to "Something capable of being perceived by the senses."
   (c) The verb glosses were modified by adding 'To' at the beginning of the gloss and a period at the end. The definition of '*swagger.v.03*' is transformed from "act in an arrogant, overly self-assured, or conceited manner" to "To act in an arrogant, overly self-assured, or conceited manner."

Tokenization. The pre-processing performed on the glosses makes it easier for tokenization. We use *tokkie*, the tokenizer offered by the C&C tools [21,19]. Tokenization is performed with *removing quotes* option on.

POS CRFTagger. After tokenization we use the CRFTagger, a Conditional Random Field POS tagger for English [20]. The model was trained on sections 01-24 of Wall Street Journal (WSJ) corpus and using section 00 as the development test set (accuracy of 97.00%) on the Penn Treebank [22]. Even though its reported accuracy is similar to those of Stanford [23] and C&C tools [19] POS taggers also trained on the Penn Treebank, we chose CRFTagger due to its speed in processing large collections of documents.

Dependency parser. We feed the POS tagged glosses to the Stanford Parser [24] in order to output a syntactic analysis consisting of *Stanford typed dependencies*, which amount to a kind of grammatical relations between lemmatized words acting as nodes of a dependency graph [18]. We take advantage of the parser's ability to output the dependency graphs in XML format for a better integration in ILF-WN.

Generation of ILFs. We automatically generate Intermediate Logic Forms from the typed dependencies output of the Stanford Parser, enriching its output with any available lexical and semantic information.

4.2 Description of ILF-WN

Version 0.2 of ILF-WN consists a collection of validated XML documents distributed in two separate packages: (1) Four main files, one per part-of-speech;

(2) a set of files, one per synset, each file identified by its *offset* (a unique identification number for each synset). Both formats contain the part-of-speech (POS), syntactic and ILFs annotations for every gloss in WordNet 3.0.

ILF-WN provides a structured annotation of every gloss in terms of their part-of-speech, syntactic analysis using a dependency parser, and the result of transforming the syntax into an Intermediate Logic Form (ILF). Example 3, which shows the structure of the synset *bigheaded.s.01* in ILF-WN, will be used to describe the resource in more detail.

Being a formalization of WordNet's glosses, ILF-WN is structured in synsets, namely, in senses expressed by a set of synonym words and a gloss. As shown in Example 3 every <sense> element in ILF-WN has three attributes: A unique numeric identifier or *offset*, its POS category in WordNet notation ('a' for *adjectives*, 's' for *satellite adjectives*, 'r' for *adverbs*, 'v' for *verbs* and 'n' for *nouns*), and the synset name, which consists of a lemma, its POS category and the sense number. In Example 3 the synset name is *bigheaded.s.01*, which translates to 'the first sense of the satellite adjective bigheaded'. Decomposing the *offset*, the first digit identifies the POS of the synset, followed by an eight digit number (in the format of the Prolog version of WordNet 3.0 [25]). The first digit of nouns is '1', verb is referred by '2', both adjectives and satellite adjectives are collapsed and start with '3'. Finally, adverbs' offsets start with '4'.

Every <sense> element includes two required and one optional sub-elements: <gloss>, <lemma> (at least one), and <examples> (zero or more). The lemma elements contain the different lemmas of words by which a sense is expressed in WordNet (they are considered synonyms). There might also be some examples of sentences including a use of a word expressing this particular sense. In Example 3, 'bigheaded', 'persnickety', 'snooty', 'snot-nosed', 'snotty', 'stuck-up', 'too_big_for_one's_breeches', and 'uppish', are the 7 lemmas of words that characterize the sense glossed as "Something overly conceited or arrogant". There are also two examples conveying this sense by means of some of the synonym words.

The linguistic annotation specific to ILF-WN is performed on the preprocessed glosses' definitions specified in the <text> element. After tokenizing, POS tagging and dependency parsing, the resulting annotation is placed in the <parse> element in XML format. The dependency graph consists of the POS tagged and lemmatized words of the gloss and the grammatical relations between them. From the dependency graph an ILF is generated and placed in the <ilf> element. For easier readability, we also provide a pretty print of ILF in the <pretty-ilf> element.

5 Comparison with Other Approaches

ILF-WN bears a number of similarities with respect to both XWN 2 and WN30-lfs as its aim, providing lexical knowledge to support semantic inference, fully coincides with their purpose. However, ILF-WN offers a number of particularities added in order to improve the final resource. Although our discussion is based

Example 3. Synset *bigheaded.s.01* in ILF-WN.

```
<sense offset="301890382" pos="s" synset_name="bigheaded.s.01">
  <gloss>
    <text>Something overly conceited or arrogant.</text>
    <parse parser="Stanford parser 1.6.1">
      <s id="1">
        <words pos="true">
          <word ind="1" pos="NN">something</word>
          <word ind="2" pos="RB">overly</word>
          <word ind="3" pos="JJ">conceited</word>
          <word ind="4" pos="CC">or</word>
          <word ind="5" pos="JJ">arrogant</word>
          <word ind="6" pos=".">.</word>
        </words>
        <dependencies style="typed">
          <dep type="advmod">
            <governor idx="3">conceited</governor>
            <dependent idx="2">overly</dependent>
          </dep>
          <dep type="amod">
            <governor idx="1">something</governor>
            <dependent idx="3">conceited</dependent>
          </dep>
          <dep type="amod">
            <governor idx="1">something</governor>
            <dependent idx="5">arrogant</dependent>
          </dep>
          <dep type="conj_or">
            <governor idx="3">conceited</governor>
            <dependent idx="5">arrogant</dependent>
          </dep>
        </dependencies>
      </s>
    </parse>
    <ilf version="0.2">[rel(1,3,2,‘advmod’,G1_3,G1_2),
    rel(1,1,3,‘amod’,G1_1,G1_3), rel(1,1,5,‘amod’,G1_1,G1_5),
    rel(1,3,5,‘conj_or’,G1_3,G1_5), e(1,2,G1_2),
    w(1,2,‘overly’,‘r’,‘rb’), e(1,3,G1_3),
    w(1,3,‘conceited’,‘a’,‘jj’), syn(1,3,301891773), e(1,1,G1_1),
    w(1,1,’something’,‘n’,’nn’), e(1,5,G1_5),
    w(1,5,‘arrogant’,‘a’,‘jj’), syn(1,5,301889819)]</ilf>
    <pretty-ilf>something(x1) amod(x1,x3) amod(x1,x5) overly(x2)
    conceited(x3) advmod(x3,x2) conj_or(x3,x5) arrogant(x5)
    </pretty-ilf>
  </gloss>
  <lemma id="0">bigheaded</lemma>
  <lemma id="1">persnickety</lemma>
  <lemma id="2">snooty</lemma>
  <lemma id="3">snot-nosed</lemma>
  <lemma id="4">snotty</lemma>
  <lemma id="5">stuck-up</lemma>
  <lemma id="6">too_big_for_one’s_breeches</lemma>
  <lemma id="7">uppish</lemma>
  <example id="0">a snotty little scion of a degenerate family-
  Laurent Le Sage</example>
  <example id="1">they’re snobs--stuck-up and uppity and
  persnickety</example>
</sense>
```

on specific examples, most of the points made here are in general applicable to most of the logic forms of WordNet glosses.

First, pre-processing of glosses is a important step to ensure the quality of the resource, specially to remove any redundant and superfluous information from the glosses definitions. Comparing Examples 1 and 2 with 3, it is possible to see that while in Examples 1 and 2 the most relevant concepts (*overly conceited or arrogant*) were somewhat buried among other no so relevant information, in Example 3 it is in a prominent position both in the <parse> and the <ilf> elements.

Second, we have tried to simplify the generation of logical forms with respect to XWN 2 and WN30-lfs, with the objective of avoiding free variables, predicates not related to any other predicates, heterogeneity of the predicates arity, not obvious decisions with respect to the treatment of disjunction, or including the *definiendum* in the *definiens*.

A delicate issue related to logic forms is to decide the argument structure of words, specially verbs with different meanings. In previous representations, this must be specified, requiring some kind of mapping with other resources such as FrameNet [26]. Our representation overcomes this problem by allowing predicates to have its particular argument structure in each particular sentence.

An important feature of XWN 2 and WN30-lfs is the inclusion of word senses in the logical form of glosses. However, in these representations is not possible to consider the complete sense probability distribution of one word, or the different senses coming from different source ontologies. Although we didn't apply any existing disambiguation method to the glosses, the ILF representation proposed here allows to include word sense disambiguation adding the corresponding predicates linked to the corresponding word indexes.

6 Conclusion and Future Work

This paper presents ILF-WN, a freely available XML-structured resource that provides an Intermediate Logic Form for WordNet 3.0 glosses. We have compared ILF-WN with Extended WordNet and WN30-lfs and, while being inspired by them, we aimed to sort out a number of shortcoming presented in those projects. We have also discuss the suitability of ILFs (and of ILF-WN) for the treatment of semantic problems at discourse level.

However, there are several aspects on which ILF-WN has to improve, most notably, on a procedure to include word sense disambiguation [27]. Furthermore, co-reference and anaphora resolution seem to be particularly relevant for noun synsets. For example, the ILF of the (pre-processed) gloss of *blot.n.02*, "An act that brings discredit to the person who does it.", would presumably benefit from resolving the definite description 'the person' to 'who' and 'it' to 'an act'.

ILF-WN could be quantitatively evaluated following the procedure of Task 16 SemEval-2007, for the Evaluation of wide coverage knowledge resources [28]. In this sense it would be similar to the evaluation provided for eXtended Wordnet in [29] where they evaluated XWN's capability of disambiguating words contained in the glosses as reported in section 2.1.

We believe that as we improve ILF-WN towards version 1.0, we will be able to offer both intrinsic (perhaps based on WSD) and extrinsic (based on a task such as RTE [1]) evaluations of the resource.

Acknowledgments

This work has been supported by Madrid R+D Regional Plan, MAVIR Project, S-0505/TIC/000267. (http://www.mavir.net) and by the Spanish Government through the "Programa Nacional de Movilidad de Recursos Humanos del Plan Nacional de I+D+i 2008-2011 (Grant PR2009-0020).

References

1. Dagan, I., Glickman, O., Magnini, B.: The PASCAL Recognising Textual Entailment challenge. In: Quiñonero-Candela, J., Dagan, I., Magnini, B., d'Alché-Buc, F. (eds.) MLCW 2005. LNCS (LNAI), vol. 3944, pp. 177–190. Springer, Heidelberg (2006)
2. Clark, P., Murray, W., Thompson, J., Harrison, P., Hobbs, J., Fellbaum, C.: On the role of lexical and world knowledge in RTE3. In: Proceedings of the Workshop on Textual Entailment and Paraphrasing, ACL 2007, Prague, pp. 54–59 (2007)
3. Bos, J., Markert, K.: Recognizing textual entailment with robust logical inference. In: Quiñonero-Candela, J., Dagan, I., Magnini, B., d'Alché-Buc, F. (eds.) MLCW 2005. LNCS (LNAI), vol. 3944, pp. 404–426. Springer, Heidelberg (2006)
4. Giampiccolo, D., Magnini, B., Dagan, I., Dollan, B.: The Third PASCAL Recognizing Textual Entailment Challenge. In: Proceedings of the Workshop on Textual Entailment and Paraphrasing, Association for Computational Linguistics (ACL 2007), Prague, pp. 1–9 (2007)
5. MacCartney, B., Manning, C.: Modeling semantic containment and exclusion in natural language inference. In: Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), Manchester, UK, pp. 521–528 (2008)
6. Harabagiu, S.M., Miller, G.A., Moldovan, D.I.: eXtended WordNet - A Morphologically and Semantically Enhanced Resource (2003), http://xwn.hlt.utdallas.edu
7. Clark, P., Fellbaum, C., Hobbs, J.R., Harrison, P., Murray, W.R., Thompson, J.: Augmenting WordNet for Deep Understanding of Text. In: Bos, J., Delmonte, R. (eds.) Semantics in Text Processing. STEP 2008 Conference Proceedings. Research in Computational Semantics, vol. 1, pp. 45–57. College Publications (2008)
8. Davidson, D.: Essays on Actions and Events. Oxford University Press, Oxford (1980)
9. Kamp, H., Reyle, U.: From Discourse to Logic: Introduction to Modeltheoretic semantics of natural language, formal language and Discourse Representation Theory. Kluwer Academic Publishers, Dordrecht (1993)
10. Hobbs, J.: Ontological promiscuity. In: Annual Meeting of the ACL, Chicago, pp. 61–69 (1985)
11. Bos, J.: Computational semantics in discourse: Underspecification, resolution, inference. Journal of Logic, Language and Information 13, 139–157 (2004)
12. Rus, V.: Logic Form for WordNet Glosses and Application to Question Answering. PhD thesis, Computer Science Department, School of Engineering, Southern Methodist University, Dallas, Texas (2002)

13. Moldovan, D., Rus, V.: Explaining Answers with Extended WordNet. In: Proceedings of the Association for Computational Linguistics, ACL 2001 (2001)
14. Charniak, E.: A Maximum-Entropy-Inspired Parser. In: Procedings of the North American Association for Computational Linguistics, NAACL (2000)
15. Information Science Institute, University of Southern California: Logical Forms for WordNet 3.0 glosses (2007), http://wordnetcode.princeton.edu/standoff-files/wn30-lfs.zip
16. WordNet Gloss Disambiguation Project, Princeton University: Semantically annotated gloss corpus (2008), http://wordnet.princeton.edu/glosstag.shtml
17. Alias-i: Lingpipe 3.8.2 (2008), http://alias-i.com/lingpipe
18. de Marneffe, M.C., MacCartney, B., Manning, C.: Generating typed dependency parses from phrase structure parses. In: Proceedings of Language Resources and Evaluation Conference, LREC (2006)
19. Clark, S., Curran, J.: C&C tools (v1.0), http://svn.ask.it.usyd.edu.au/trac/candc
20. Phan, X.H.: CRFTagger: CRF English POS Tagger (2006), http://sourceforge.net/projects/crftagger
21. Clark, S., Curran, J.: Wide-coverage efficient statistical parsing with CCG and Log-Linear Models. Computational Linguistics 33, 493–553 (2007)
22. Marcus, M.P., Santorini, B., Marcinkiewicz, M.A.: Building a Large Annotation Corpus of English: The Penn Treebank. Computational Linguistics 19, 313–330 (1993)
23. Toutanova, K., Klein, D., Manning, C., Singer, Y.: Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In: Proceedings of HLT-NAACL, pp. 252–259 (2003)
24. The Stanford Natural Language Processing Group: The Stanford Parser: A statistical parser, http://nlp.stanford.edu/software/lex-parser.shtml
25. Prolog Version of WordNet 3.0 (2008), http://wordnetcode.princeton.edu/3.0/wnprolog-3.0.tar.gz
26. Ruppenhofer, J., Ellsworth, M., Petruck, M., Johnson, C., Sheffczyk, J.: Framenet ii: Extended theory and practice (2006), http://framenet.icsi.berkeley.edu/book/book.html
27. Agirre, E., Soroa, A.: Personalizing pagerank for word sense disambiguation. In: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2009), Athens, Greece (2009)
28. Cuadros, M., Rigau, G.: Semeval-2007 task 16: Evaluation of wide coverage knowledge resources. In: Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval 2007), Prague, Czech Republic, pp. 81–86. Association for Computational Linguistics (2007)
29. Harabagiu, S., Miller, G., Moldovan, D.: Wordnet 2 - a morphologically and semantic enhanced resource. In: Proceedings of SIGLEX (1999)

Worth Its Weight in Gold or Yet Another Resource — A Comparative Study of Wiktionary, OpenThesaurus and GermaNet

Christian M. Meyer and Iryna Gurevych

Ubiquitous Knowledge Processing Lab
Technische Universität Darmstadt
Hochschulstraße 10, D-64289 Darmstadt, Germany
http://www.ukp.tu-darmstadt.de

Abstract. In this paper, we analyze the topology and the content of a range of lexical semantic resources for the German language constructed either in a controlled (GermaNet), semi-controlled (OpenThesaurus), or collaborative, i.e. community-based, manner (Wiktionary). For the first time, the comparison of the corresponding resources is performed at the word sense level. For this purpose, the word senses of terms are automatically disambiguated in Wiktionary and the content of all resources is converted to a uniform representation. We show that the resources' topology is well comparable as they share the small world property and contain a comparable number of entries, although differences in their connectivity exist. Our study of content related properties reveals that the German Wiktionary has a different distribution of word senses and contains more polysemous entries than both other resources. We identify that each resource contains the highest number of a particular type of semantic relation. We finally increase the number of relations in Wiktionary by considering symmetric and inverse relations that have been found to be usually absent in this resource.

1 Introduction

Large-scale acquisition of lexical semantic knowledge from unstructured corpora has become a hot research topic, since numerous natural language processing tasks like semantic search, automatic word sense disambiguation or calculating semantic relatedness require large lexical semantic resources as a source of background knowledge. *Expert-built lexical semantic resources* (ELSR) like WordNet [1] or GermaNet [2] are hand-crafted in a controlled manner by linguists and have been extensively used for such applications. Keeping ELSRs up-to-date is however a costly and time-consuming process, which leads to limited coverage and thus insufficiency for obtaining high quality results in above tasks. Especially for languages other than English, ELSRs suffer from their small size.

With the evolution of the socio-semantic web, a new type of resources has emerged: *collaboratively constructed lexical semantic resources* (CLSR) like Wikipedia or Wiktionary, which are created by a community of (mainly) non-experts

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 38–49, 2010.

on a voluntary basis. As CLSRs are constantly updated by their community, they benefit from the wisdom of crowds and avoid the costly maintenance process of ELSRs. Zesch et al. [3] found that Wiktionary outperforms ELSRs when used as a source of background knowledge for calculating semantic relatedness.

Our assumption is that a combination of ELSRs and CLSRs would lead to better results, since it profits from the high quality of ELSRs and the broad coverage of CLSRs. The structural and content related properties of the latter are however largely unknown. We therefore perform a comparative study of Wiktionary, GermaNet and OpenThesaurus, in order to learn about their content as well as the individual strengths and weaknesses.[1]

Previous studies regarded Wiktionary's lexical semantic relations at the term level, although they are generally marked with a certain word sense. For the first time, we analyze them at the word sense level, whereby an automatic word sense disambiguation algorithm is applied to relations without sense marker.

2 Description of Lexical Semantic Resources

We have chosen the three resources Wiktionary, GermaNet and OpenThesaurus for our study, because they cover well the range between ELSR and CLSR: GermaNet is fully expert-created, while Wiktionary is clearly a CLSR with a large community of volunteers. It is not controlled by an editorial board. OpenThesaurus is in between, as it is collaboratively constructed but has a much smaller community and is reviewed and maintained by an administrator [4]. In the following, we describe each individual resource and their representational units.

Our notation mainly follows [5]. A *term* is a word form that is characterized by a certain string, e.g., *bass* or *read*.[2] A *lexeme* is a term that is tagged with its part of speech, e.g., *bass (noun)* or *read (verb)*. Each lexeme can be used in one or more *word senses* that carry the meaning of a lexeme. For the lexeme *bass (noun)* there could e.g. be two word senses *bass*⟨*music*⟩ and *bass*⟨*fish*⟩. Note that in this definition, a word sense is bound to a certain lexeme rather than representing a concept. The latter will be called a *synset* (short for synonymy set) that combines word senses with the same meaning but represented by different lexemes. The set {*bass*⟨*fish*⟩*, perch, Percidae*} is e.g. a synset for the meaning '*any of various marine and freshwater fish resembling the perch, all within the order of Perciformes*' that consists of three synonymous word senses. We use the notation $s \in S$ to indicate that word sense s is included in the synset S.

A *relation* is a pair (*source, target*), where *source* and *target* denote word senses that the relation connects. Relations are directed from source to target and have a certain relation type [5]. The term *bass* has e.g. a synonymy relation

[1] Although we focus on German resources, our methods are not language dependent and can also be applied to similar resources in other languages. Particularly, we conducted a study of the English Wiktionary, WordNet and Roget's Thesaurus and report our results at: http://www.ukp.tu-darmstadt.de/data/lexical-resources

[2] We provide English examples where possible to improve the understandability of the paper and choose words with similar ambiguities rather than translating literally.

(*bass⟨fish⟩, perch*) and a hypernymy relation (*bass⟨fish⟩, Perciformes*). For relations of type *synonymy* and *antonymy*, there can be a *symmetric* relation of the same type that connects the target with the source. Relations of the types *hypernymy*, *hyponymy*, *holonymy* and *meronymy* have however no symmetric but inverse relations that connect the target with the source. Instances of inverse relations are hypernymy–hyponymy and holonymy–meronymy. For the synonymy relation (*bass⟨fish⟩, perch*), there is e.g. a symmetric relation (*perch, bass⟨fish⟩*), while the hypernymy relation (*bass⟨fish⟩, Perciformes*) can have the inverse hyponymy relation (*Perciformes, bass⟨fish⟩*). A relation whose symmetric or inverse counterpart does not exist in a given resource will be called a *one-way relation*, otherwise a *two-way relation*.

Wiktionary[3] is a large online dictionary that is collaboratively constructed by a community. The resource is organized in article pages that represent a certain term and can consist of multiple lexemes. Each lexeme is tagged with its language and part of speech and can distinguish different word senses, which are represented by glosses. Figure 1 shows the Wiktionary entry *bass* as an example of this structure. Semantic relations are encoded as links to other articles. Wiktionary is available in more than 300 languages. Each language edition contains word entries from multiple languages. An entry about the English term *railway* can e.g. be found in both the German and the English Wiktionary. For our study, we focus solely on the German word entries in the German language version, which are parsed and accessed using the freely available Java-based Wiktionary Library[4] [6] and a Wiktionary dump of June 18, 2009.

Fig. 1. Wiktionary article *bass* with highlighted term, lexeme and word sense sections

GermaNet[5] [2] is an ELSR for the German language that is similar to the well-known Princeton WordNet [1]. GermaNet consists of a set of synsets that contain one or more word senses. While lexical relations such as antonymy are defined between lexemes, taxonomic relations like hypernymy can only exist between synsets. We use GermaNet 5.0 that is available upon a license.

[3] http://www.wiktionary.org
[4] http://www.ukp.tu-darmstadt.de/software/jwktl
[5] http://www.sfs.uni-tuebingen.de/lsd

OpenThesaurus[6] [4] is a thesaurus for the German language. Its main focus is collecting synonyms, but also some taxonomic relations can be found in the resource. OpenThesaurus consists of a list of *meanings* (synsets) that can be represented by one or more *words* (terms). The resource is released as a full database dump from the project homepage. We use a dump of July 27, 2009.

3 Related Work

To our knowledge, there is no other comparative study of the three resources Wiktionary, GermaNet and OpenThesaurus that analyzes both topological and content related properties. The latter issue has been addressed for single resources, but without any comparison [6,2,4]. Garoufi et al. [7] compared the topological properties of Wiktionary with GermaNet and both the Wikipedia category and article graphs. They however do not convert the resources into a uniform representation. Topological properties are also analyzed by Navarro et al. [8], who built a graph of synonymy links from the French, English, German and Polish Wiktionaries. They found similar properties for the different language versions. Both the studies regard Wiktionary relations between terms rather than word senses. The two hypernymy relations (*smallmouth bass, bass*⟨*fish*⟩) and (*bass*⟨*music*⟩*, pitch*) then share the vertex *bass*, which leads to a path length of only 2 between *smallmouth bass* and *pitch*. This is different from ELSRs like WordNet or GermaNet that encode such relations between word senses or synsets and may result in a biased comparison of the resources. We solve this problem by applying automatic word sense disambiguation to the Wiktionary relations.

4 Representing Lexical Semantic Resources as Graphs

In order to allow a systematic and fair comparison, all resources need to be converted into a uniform representation. We therefore introduce a directed graph $G = (V, E)$ of all word senses V and the corresponding set of relations $E \subseteq V^2$. Each resource has however its unique representation and thus requires an individual approach to the graph construction described below.

Wiktionary. The source of a Wiktionary relation is usually associated with a certain word sense. The syntax *[2] fish* within the article *bass*, e.g., indicates that the second sense of *bass* (the fish within the order of Perciformes) is the source of a (hypernymy) relation to the target term *fish*. Unfortunately, the target of a relation is not sense disambiguated in general, as it is only given by a link to a certain article. For the term *fish* in the relation above, it is not clear whether the maritime animal, a part of a ship's mast or a card game is meant. Automatic word sense disambiguation is required to determine the correct sense of the target. To our knowledge, this issue has not been addressed in any of the works based on Wiktionary.

[6] http://www.openthesaurus.de

Let (u, v) be a Wiktionary relation with the source word sense u and a target term v. We first determine the set of candidate word senses, i.e. all word senses that are defined for term v. Then, the semantic relatedness between the source and each candidate is calculated, based on the sense gloss and usage examples that will be called *extended gloss* in the following. The candidate with the highest score is chosen as the relation target. Figure 2 outlines this approach formally.

function RELATIONTARGETWSD(u, v)
  $g1$:= gloss(u) + examples(u);
  Candidates := {};
  score : *Candidates* $\longrightarrow \mathbb{R}$;
  for each Wiktionary word sense c of term v **do**
    Candidates := *Candidates* $\cup$ {c};
    $g2$:= gloss(c) + examples(c);
    score(c) := calcESA($g1, g2$);
  end;
  return arg max$_{c \in \text{Candidates}}$ *score*(c);
end.

Fig. 2. Automatic word sense disambiguation method for Wiktionary's relation targets

The semantic relatedness is computed using Explicit Semantic Analysis based on Wikipedia, which has been introduced to be capable of solving word sense disambiguation tasks [9]. It forms a vector space from all Wikipedia articles and creates a concept vector c for two input terms consisting of the *tfidf* scores [10] between the term and each Wikipedia article. The cosine of the concept vectors is then calculated as their semantic relatedness. Since we need to compare extended glosses, i.e. short texts, rather than single words, we use an extension of this method [3]: The concept vectors $c(t)$ of all non-stopword tokens $t \in g$ of the extended gloss g are calculated with the above method and combined by computing the normalized sum of the vectors, leading to:

$$\text{calcESA}(g1, g2) = \frac{c(g1) \cdot c(g2)}{|c(g1)| \cdot |c(g2)|} \quad \text{with} \quad c(g) = \frac{1}{|g|} \sum_{t \in g} c(t)$$

Consider e.g. the hypernymy relation (*bass*⟨*fish*⟩, *fish*). There are three target candidates for the term v = *fish* with relatedness scores: *score*(*fish*⟨*maritime animal*⟩) = .35, *score*(*fish*⟨*part of a mast*⟩)= .13 and *score*(*fish*⟨*card game*⟩) = .16. The word sense with the maximum *score* is chosen, which is *fish*⟨*maritime animal*⟩ in this case.

To evaluate this approach, we annotated 250 randomly sampled Wiktionary relations by marking each of the 920 possible target candidates with either + if the specified relation (u, v) holds, or with − otherwise. The annotators were allowed to assign multiple target senses of a relation with + if more than one relation holds, whereas also no + was possible. There is e.g. a hyponymy relation (*Antwerp*⟨*Province*⟩, *Essen*) about a Belgian municipality whose target

has only the three word sense candidates *nutrition*, *meal* and *German city*, so none of them was selected.[7] The annotations were created independently by two annotators, who are both German native speakers. Table 1 shows the number of target candidates both annotators agreed on, namely $(+,+)$ and $(-,-)$, as well as the number of candidates that the annotators did not agree on: $(+,-)$ and $(-,+)$. We report these numbers separately for each level of ambiguity a, i.e. the number of possible targets for a given relation and note the relation count r_a for each level. There are e.g. $r_a = 2$ relations that both have $a = 15$ target candidates, of which 1 was considered correct and 27 incorrect by both annotators, while they disagreed on 2 of them. We observe a uniform disagreement D_a at each level of ambiguity, although it is slightly higher for $a = 4$ and $a = 10$.

Table 1. Agreement table for the word sense disambiguation of Wiktionary relations

a	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Sum
r_a	103	59	27	24	8	9	7	5	3	1	1	0	1	2	250
$(+,+)$	90	50	23	23	7	9	6	5	1	1	2	0	1	1	219
$(-,+)$	14	8	5	2	2	2	0	1	0	1	0	0	0	1	36
$(+,-)$	19	5	13	7	2	6	5	2	6	0	1	0	1	1	68
$(-,-)$	83	114	67	88	37	46	45	37	23	9	9	0	12	27	597
D_a	.16	.07	.17	.08	.08	.13	.09	.07	.20	.09	.08	.00	.07	.07	

We measured the inter-annotator agreement following the methods introduced in [11] to ensure the reliability of our data. Therefore, we first considered the data as 920 binary annotations that judge if a target candidate is valid for a given source word sense and measured an observed agreement $A_O = .88$ and a chance-corrected agreement of $\kappa = .72$, which allows tentative conclusions [11]. We then interpreted our data as 250 set-valued annotations that provide a set of valid target word senses for a given relation. For measuring the agreement of set-valued data, we used MASI [12] as a distance function for Krippendorff's α, which resulted in $\alpha = .86$ and indicates good agreement. We refrained from removing items where no agreement was reached, since these are the most difficult instances whose removal would lead to biased results. We rather measured the agreement between our algorithm M and both the human annotators A and B. Besides the inter-annotator agreement A–B, which serves as an upper bound, we tried the naïve baseline approach *0* that always chooses the first target word sense. Table 2 summarizes the evaluation results. Our approach exceeds the baseline in each case. There is however room for improvements with respect to the upper bound A–B. We plan to compare several approaches in our future work.

The algorithm exclusively relies on the semantic relatedness of the word senses' extended glosses. Thus, the disambiguation is likely to fail if only a short or very general gloss is given, which has been found to be the most common source of errors. Besides cases, where the community did not provide a meaningful gloss, there are also minor errors in the extraction API that lead to truncated

[7] On June 27, 2009 the missing word sense has been added to the article *Essen*.

Table 2. Evaluation results of our word sense disambiguation approach

	0–A	*0–B*	*M–A*	*M–B*	*A–B*
A_O	.791	.780	.820	.791	.886
κ	.498	.452	.567	.480	.728
α	.679	.620	.726	.649	.866

glosses. Other errors are caused by references to other word senses within a gloss; the second sense of *tomato*, e.g., refers to its first sense: *[2] the fruit of [1].*

GermaNet and OpenThesaurus. To obtain a uniform representation of the resources, the synsets in GermaNet and OpenThesaurus need to be decomposed into the individual word senses. We therefore add a node to V for each word sense $s \in S$ of any synset S. Accordingly, an edge (s_1, s_2) is added to E for each word sense $s_1 \in S_1$ and $s_2 \in S_2$ of a relation (S_1, S_2) between synsets. As synsets represent sets of synonyms, we also add a synonymy edge (s_1, s_2) for all $s_1, s_2 \in S$, which results in a fully connected subgraph for each synset. Consider e.g. the synset {*bass⟨fish⟩, perch, Perciformes*}. The three word senses are added to V and the synonymy edges (*bass⟨fish⟩, perch*), (*bass⟨fish⟩, Perciformes*) and (*perch, Perciformes*) as well as their symmetric counterparts are added to E.

5 Topological Analysis of Resources

We now use this uniform representation of the resources and study topological properties of their graphs. Table 3 shows the results of our study for Wiktionary (WKT), GermaNet (GN) and OpenThesaurus (OT).

For applications that aim to calculate semantic relatedness using a lexical semantic resource, it is often crucial that the resource graph is connected. As none of the resources is connected as a whole, we studied the number of connected components CC, the largest ($lcc1$) and the second largest ($lcc2$) connected components. GermaNet was found to contain the fewest connected components, only about 2% of the respective number in Wiktionary and OpenThesaurus. 98% of all vertices are within $lcc1$ in GermaNet, thus allowing to use almost the whole

Table 3. Comparison of topological properties

	$\lvert V\rvert$	$\lvert E\rvert$	CC	$\lvert V_{\text{lcc1}}\rvert$	$\lvert E_{\text{lcc1}}\rvert$	$\lvert V_{\text{lcc2}}\rvert$	$\lvert E_{\text{lcc2}}\rvert$
WKT	107,403	157,786	20,114	80,638	149,947	69	68
GN	76,864	394,856	471	75,848	393,072	49	149
OT	87,735	288,121	26,624	12,742	48,590	704	4,078

	γ_{lcc1}	R^2	ℓ_{lcc1}	ℓ_{rand}	c_{lcc1}	c_{rand}	o_{lcc1}	o_{rand}
WKT	-2.37	96.2%	1.3	8.5	0.13	<0.01	0.59	0.32
GN	-1.71	75.9%	10.8	9.1	0.24	<0.01	0.41	0.11
OT	-1.91	63.4%	<0.01	4.8	0.26	<0.01	0.48	0.15

resource for applications that require connected graphs. For Wiktionary, 75% of the vertices are in *lcc*1, which leads to a similar number of nodes compared to GermaNet — the difference in $|V_{\text{lcc1}}|$ is merely 4,950. In OpenThesaurus, only 14% of the vertices are contained in *lcc*1, which makes it less useful for such tasks. We also analyzed *lcc*2, as it reveals if the remaining graph forms a usable semantic network itself or only consists of mainly unconnected vertices. Each resource showed a very small *lcc*2, both in the number of vertices and edges. It is thus sufficient to focus on the *lcc*1 as it contains the bulk of semantic information.

Albert and Barabási [13] studied the topology of several real world graphs and found governing organizational principles that significantly differ from those in random graphs. We applied their experimental approaches to our resource based graphs. The *degree distribution* of graph G is a function $D\colon \mathbb{N} \longrightarrow \mathbb{N}$ that maps each possible degree to its number of occurrences: $d \mapsto |\{v \in V \mid \deg(v) = d\}|$. While the function follows a normal distribution for random graphs, it shows a *power law distribution* for many real world graphs, which results from the way a graph grows over time and its organizational structures [14]. Such graphs are called scale-free, since their topology remains stable, regardless of their size. For a power law, the probability of each node $v \in V$ to have degree k is proportional to the γ-th power of k:

$$P(\deg(v) = k) \propto k^{-\gamma}$$

Garoufi et al. [7] studied if the degree distribution of Wiktionary and GermaNet follows a power law but did not provide any goodness-of-fit analysis to evaluate the quality of the fitted parameter γ. We use the coefficient of determination R^2 [15] for this purpose. The nearer R^2 is to 100%, the stronger is the evidence for a power law. In our setting, the Wiktionary graph shows a clear power law and can be considered scale-free, which was previously reported in [7,8]. For both other resources, R^2 is considerably lower. This is a surprising observation, since [7] found a power law in the degree distribution of the GermaNet graph. One explanation could be that their observed power law is not significant, as no goodness-of-fit analysis is provided. Another possibility is that our uniform representation of resources leads to different results. Further analyses need to be applied to learn about GermaNet's degree distribution. While the scale-free Wiktionary graph allows to project our topological insights to future (larger) versions of Wiktionary, this does not necessarily hold for GermaNet and OpenThesaurus.

Real world graphs tend to show a *small world property* [13]. Such graphs usually have a small average path length ℓ over each node pair $(u, v) \in V^2$. Besides that, they have a high fraction of transitive triplets, which can be measured by the clustering coefficient c, i.e. the average probability that two neighbors of a node are connected by an edge [16]. Both measures are required to clearly differ from the corresponding values of a random graph with similar vertex and edge count [13]. Table 3 contains the two measures for the resource's largest connected component (ℓ_{lcc1} and c_{lcc1}) together with the corresponding results of a random graph (ℓ_{rand} and c_{rand}). The clustering coefficient differs about an order of magnitude from a corresponding random graph. In Wiktionary and OpenThesaurus,

ℓ_{lcc1} is clearly lower than in the corresponding random graph. The small world property is thus clearly visible for these two resources.

The average path length of the GermaNet graph is slightly higher than ℓ_{rand}, which can also be seen in [7]. Especially terms from different parts of speech contribute path lengths of up to 39, which is the diameter of the graph. An average path length of 10.8 is still low for a graph of this size, we however aimed at comprehensibly verifying the existence or absence of the small world property. We therefore calculated the *topological overlap* o_{lcc1} for each resource graph as a third topological measure and compared it to the o_{rand} of the corresponding random graph. The topological overlap is the average $o(u, v)$ for each pair $(u, v) \in V^2$, which measures the number of vertices to which both u and v are linked. A high topological overlap characterizes hierarchical and small world graphs [17]. Our results in Table 3 show a considerably higher o_{lcc1} for the resource graphs compared to o_{rand} — in particular for the GermaNet graph, which hence reveals also a small world property for this resource.

Comparing lexical semantic resources requires a similar topology of their induced graphs. The small world property is a good indicator for that. It not only allows a fair and unbiased comparison but also promises that a combination of the resources is governed by the same structures that they show individually.

6 Content Analysis of Resources

After studying the resource topology, we focused on their content and examined the number of lexemes, word senses and relations. Table 4 shows the determined results. Each of the three resources contains a comparable number of lexemes and word senses. Wiktionary is however the largest resource with 23,857 lexemes more than OpenThesaurus, which is the smallest. GermaNet on the contrary contains the highest number of relations, 2.5 times more than Wiktionary and 1.3 times more than OpenThesaurus. This makes GermaNet the most densely connected resource. Wiktionary encodes a distinction between polysemy and homonymy: The former is expressed in word senses, while the latter is represented by different lexemes that arise from different etymology. None of the other resources explicitly encodes this type of information.

The target of a Wiktionary relation is represented by a link to a certain article, which is sometimes yet missing due to the collaborative construction approach. Therefore, a large number of relations exist whose targets are fairly rare terms still not encoded in the resource in the form of a dedicated Wiktionary entry. The article *bass* e.g. links to an article *bass music*, which has not yet been created by the community. We will refer to such relation targets as *dangling lexemes*. 56% of the lexemes in Wiktionary are dangling, thus showing that the resource contains many gaps. As Wiktionary is constantly growing by 1–2% of its size each month,[8] these gaps are however likely to be filled in the future and yield a lexical semantic resource with high coverage.

[8] http://stats.wikimedia.org/wiktionary/EN/TablesWikipediaDE.htm

Table 4. Descriptive statistics about the resources' content

	WKT	**GN**	**OT**
Number of lexemes:	90,611	67,402	66,754
...Homonyms:	2,327	-	-
...Monosemous:	29,025	61,129	54,939
...Polysemous:	10,643	6,273	11,815
...Dangling lexemes:	50,943	0	0
Number of word senses:	107,403	76,864	87,735
Number of relations:	157,786	394,856	288,121
...One-way:	139,453	11,941	5,731
...Synonymy:	62,235	69,097	282,390
...Antonymy:	24,167	3,486	0
...Hypernymy:	37,569	155,385	5,731
...Hyponymy:	33,815	155,237	0
...Holonymy:	0	8,977	0
...Meronymy:	0	2,674	0
Number of two-way relations:	297,120	406,328	293,846
...Two-way synonymy:	117,318	69,134	282,384
...Two-way antonymy:	43,128	3,134	0
...Two-way hypernymy:	136,674	310,856	11,462
...Two-way holonymy:	0	23,204	0

9% of the lexemes in GermaNet and 17% of the lexemes in OpenThesaurus are polysemous, i.e. at least two word senses are encoded for a lexeme. Wiktionary however contains 26% polysemous lexemes, which is significantly higher than in both other resources. Different explanations are possible for this observation: Either Wiktionary contains mainly high-frequency words that are known to be more ambiguous, or the community more likely creates articles for polysemous terms, since they might be more interesting to create. Besides that, it is also possible that the coverage of senses for a lexeme is on average higher within Wiktionary, or that the Wiktionary word senses are more fine-grained than those of the other resources. This remains to be thoroughly studied in the future.

GermaNet is the only resource that contains holonymy and meronymy relations, while its number of hypernymy and hyponymy relations is also higher than in the other resources. OpenThesaurus contains the most synonyms as it was the major goal for its creation. It yet contains less hypernyms and neither antonymy nor hyponymy relations. Wiktionary shows the most antonyms and contains nearly as many synonyms as GermaNet. At first glance, Wiktionary seems to have less relations than GermaNet and OpenThesaurus. Especially the difference to GermaNet is very prominent. Further examination however shows that 88% of the Wiktionary relations are one-way relations. GermaNet and OpenThesaurus have only between 2–3% one-way relations, which can be explained by their creation guidelines. Since synonymy and antonymy relations are symmetric and taxonomic relations are invertible, the number of relations can be increased by generating the corresponding counterparts, thus converting each relation to a two-way relation. The results of this extension are included

in Table 4 (hyponymy and meronymy are equal to their inverse counterpart and therefore omitted). Wiktionary benefits most from the extension and finally contains slightly more relations than OpenThesaurus. It still is the resource with the most antonyms and the second most synonymy and hypernymy relations.

7 Conclusions and Future Work

We analyzed the topological and content related properties of Wiktionary and compared them with GermaNet and OpenThesaurus. We have chosen the three resources, since they represent well the range between expert-built and collaboratively constructed lexical semantic resources. For the first time, we provide an analysis of lexical semantic relations in Wiktionary based on word senses. We applied word sense disambiguation to the relation targets in order to find the correct word sense of the relation target. We also transformed the synsets within GermaNet and OpenThesaurus into a set of synonymous word senses for each contained term, which allows a uniform representation of the three resources and thus a fair comparison of their encoded information. This setting is unique and has not been reported before to our knowledge.

In the first part of our analysis, we created a word sense based graph for each resource and studied the graph topology. All graphs showed the small world property, which is important for being able to compare the analysis results. The Wiktionary graph is additionally scale-free and thus allows to project our observations to future (larger) Wiktionary versions. Studying content related properties revealed that although Wiktionary contains the lowest number of relations it has the highest number of word senses. It however contains lots of dangling word senses, i.e. word senses that are used as targets of semantic relations but are not yet described in an article. The number of Wiktionary's lexical semantic relations has been greatly increased by considering also the symmetric and inverse counterpart of each relation if not directly encoded in the resource. While GermaNet provides the highest number of taxonomic relations and OpenThesaurus the highest number of synonyms, Wiktionary contains the most antonyms and the second most synonymy and hypernymy relations.

Our future work will focus on an enhanced automatic word sense disambiguation of Wiktionary's relation targets in order to compare different approaches and give a comprehensive evaluation of our method. We also plan to study the information overlap of the resources in order to learn if the resources share a large common vocabulary or contain complementary information. Besides that, we aim at analyzing English resources in a similar manner.

Acknowledgments. This work has been supported by the Volkswagen Foundation as part of the Lichtenberg-Professorship Program under grant No. I/82806. We thank Elisabeth Wolf and Torsten Zesch from the UKP Lab for their contributions to this paper and Dr. Lothar Lemnitzer from BBAW for his helpful comments.

References

1. Fellbaum, C. (ed.): WordNet: An Electronic Lexical Database (Language, Speech, and Communication). MIT Press, Cambridge (1998)
2. Kunze, C., Lemnitzer, L.: GermaNet — representation, visualization, application. In: Proceedings of the Third International Conference on Language Resources and Evaluation, Las Palmas, Canary Islands, Spain, vol. 5, pp. 1485–1491 (2002)
3. Zesch, T., Müller, C., Gurevych, I.: Using Wiktionary for Computing Semantic Relatedness. In: Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, Chicago, IL, USA, pp. 861–867 (2008)
4. Naber, D.: OpenThesaurus: ein offenes deutsches Wortnetz. In: Sprachtechnologie, mobile Kommunikation und linguistische Ressourcen: Beiträge zur GLDV-Tagung, Bonn, Germany, pp. 422–433 (2005)
5. Jurafsky, D., Martin, J.H.: Speech and Language Processing. An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Prentice Hall Series in Artificial Intelligence. Prentice Hall, Upper Saddle River (2000)
6. Zesch, T., Müller, C., Gurevych, I.: Extracting Lexical Semantic Knowledge from Wikipedia and Wiktionary. In: Proceedings of the 6th International Conference on Language Resources and Evaluation, Marrakech, Morocco, pp. 1646–1652 (2008)
7. Garoufi, K., Zesch, T., Gurevych, I.: Graph-Theoretic Analysis of Collaborative Knowledge Bases in Natural Language Processing. In: Proceedings of the Poster Session of the 7th International Semantic Web Conference, Karlsruhe, Germany (2008)
8. Navarro, E., Sajous, F., Gaume, B., Prévot, L., Hsieh, S., Kuo, I., Magistry, P., Huang, C.R.: Wiktionary for natural language processing: methodology and limitations. In: Proceedings of the ACL 2009 Workshop, The People's Web Meets NLP: Collaboratively Constructed Semantic Resources, Suntec, Singapore, pp. 19–27 (2009)
9. Gabrilovich, E., Markovitch, S.: Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, pp. 1606–1611 (2007)
10. Salton, G., McGill, M.J.: Introduction to modern information retrieval. McGraw-Hill, New York (1983)
11. Artstein, R., Poesio, M.: Inter-Coder Agreement for Computational Linguistics. Computational Linguistics 34(4), 555–596 (2008)
12. Passonneau, R.J.: Measuring agreement on set-valued items (MASI) for semantic and pragmatic annotation. In: Proceedings of the Fifth International Conference on Language Resources and Evaluation, Genoa, Italy, pp. 831–836 (2006)
13. Albert, R., Barabási, A.L.: Statistical mechanics of complex networks. Reviews of Modern Physics 74(1), 47–97 (2002)
14. Barabási, A.L., Albert, R.: Emergence of Scaling in Random Networks. Science 286(5439), 509–512 (1999)
15. Nagelkerke, N.J.D.: A note on a general definition of the coefficient of determination. Biometrika 78(3), 691–692 (1991)
16. Watts, D.J., Strogatz, S.H.: Collective dynamics of 'small-world' networks. Nature 393(6684), 440–442 (1998)
17. Ravasz, E., Somera, A.L., Mongru, D., Oltvai, Z.N., Barabási, A.L.: Hierarchical Organization of Modularity in Metabolic Networks. Science 297(5586), 1551–1555 (2002)

Issues in Analyzing Telugu Sentences towards Building a Telugu Treebank

Chaitanya Vempaty, Viswanatha Naidu, Samar Husain, Ravi Kiran, Lakshmi Bai, Dipti M Sharma, and Rajeev Sangal

Language Technologies Research Centre, IIIT-Hyderabad, India
srpchaitanya@students.iiit.ac.in
{vnaidu,samar,ravikiranv}@research.iiit.ac.in
{lakshmi,dipti,sangal}@mail.iiit.ac.in

Abstract. This paper describes an effort towards building a Telugu Dependency Treebank. We discuss the basic framework and issues we encountered while annotating. 1487 sentences have been annotated in Paninian framework. We also discuss how some of the annotation decisions would effect the development of a parser for Telugu.

1 Introduction

Currently, an effort is underway to develop a large scale treebank for Indian Languages (ILs). Lack of such resources has been a major limiting factor in the development of good natural language processing tools. It is well known that the use of Phrase Structure (PS), is not well-suited for free word order languages [16]. Instead, the dependency framework appears to be better suited [11], [13], [6]. The effort described in this paper follows the Paninian grammatical framework [6] which is a dependency based approach. Recently, the Paninian framework has been successfully used for Hindi[1] dependency annotation [2]. This paper introduces how this framework can be used for analyzing Telugu. Telugu[2], an IL, is a language with relatively high free word-order. It is also morphologically very rich.

In the past, there has been significant amount of work in preparing such annotated linguistic resources,most notably the Penn treebank (PTB) [14] for English and Prague Dependency treebank (PDT) [10] for Czech. PTB uses the Phrase Structure annotation scheme whereas PDT implements a three layered annotation scheme, namely morphological, analytical (shallow dependency syntax) and tectogrammatical (deep dependency syntax). Other major efforts in the dependency framework are Alpino [17] for Dutch, [15] for English, TUT [9] for Italian, TIGER [8] for German, a multi-representational and multi-layered treebank for Hindi/Urdu [7]. Development of a Latin Dependency Treebank (LDT) for Latin is also an ongoing work [1].

In our treebank each sentence was manually pos-tagged and chunked[3]. They were then annotated for dependency relations. While chunking, we assumed that a chunk

[1] Hindi is South Asian Language and an official language of India spoken by 300 million people.

[2] Telugu is a Dravidian language and an official language of India spoken by 75 million people.

[3] Akshar Bharati, Rajeev Sangal, Dipti Misra Sharma and Lakshmi Bai. 2006. AnnCorra: Annotating Corpora Guidelines for POS and Chunk Annotation for Indian Languages. Technical Report (TR-LTRC-31), Language Technologies Research Centre IIIT-Hyderabad. http://ltrc.iiit.ac.in/MachineTrans/publications/technicalReports/tr031/posguidelines.pdf

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 50–59, 2010.

is a minimal, non-recursive structure consisting of correlated groups of words. Karaka relations (discussed in section 2) were marked between chunk heads, as the emphasis was on showing the right modifier-modified relationship and to ignore local details[4].

For the following sentence all possible word combinations are grammatical.

((rAmudu_NNP))__NP ((paMdu_NN))__NP ((wiMtAdu_VM))__VGF.

rAmudu paMdu wiMtAdu.
'Ram' 'fruit' 'eats'.
Ram eats a fruit.

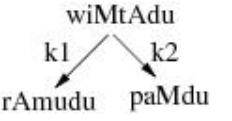


Fig. 1. Dependency Tree

In this paper we specifically discuss in detail, the following linguistic constructions::

1) Genitives:: In Telugu the genitive marker is often dropped.
2) Conjuncts:: Different constructions where a conjunct presence is explicit/implicit.
3) Copula :: Missing verbs i..e verbs are dropped.
4) "ani"[5] constructions:: Various ways of using the lexical item "ani" in language.

All the sentences are in 'wx' notation[6] and their annotation is done in SSF (Shakti Standard Format)[7].

2 Overview of Annotation Scheme

As mentioned earlier the annotation is done based on the Paninian grammatical framework that has been succesfully used for developing Hyderabad dependency treebank [2] (HyDT). The annotation scheme considers the verb as the central, binding element of the sentence. In other words, the verb's requirements for its arguments is the starting point of the analysis. The relationship between the participant and the activity/state denoted by the verb is marked using relations that are called *karaka*.

It has been shown that the notion of karaka incorporates the local semantics of a verb in a sentence and that it is syntactico-semantic [6], [18]. For example, *karta or k1 is a relation that describes an argument that is most central to the action described by the verb*. There are 6 basic karakas, namely; adhikarana 'location(k7)', apadaan 'source(k5)', sampradaan 'recipient(k4)', karana 'instrument(k3)', karma 'theme(k2)',

[4] Intra-chunk dependencies are easy to mark and a rule-based system can be developed with high performance in automatically marking the intra-chunk relations. Due to the lack of space we do not elaborate it here.

[5] Quotative marker in Telugu.

[6] In this notation, capitalization roughly means aspiration for consonants and longer length for vowels. In addition, 'w' represents 't' as in French entre and 'x' means something similar to 'd' in French de, hence the name of the notation. http://ltrc.iiit.net/ anusaaraka/SAN_MO/help.html#sec-b

[7] SSF: Shakti Standard Format Guide. Akshar Bharati, Rajeev Sangal and Dipti Misra Sharma. Technical Report no: IIIT/TR/2009/85. http://www.iiit.ac.in/techreports/2009_85.pdf

karta ‘agent(k1)’. Other than the basic karaka relations the scheme has other relations such as ‘nmod’, ‘vmod’, ‘r6’ etc. The scheme has around 28 tags[8]. The tags are hierarchical. Figure 2 shows the hierarchy of the tagset.

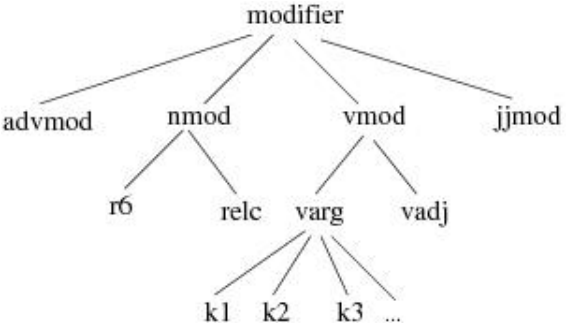


Fig. 2. Heirarchichy of tags

‘Advmod’, ‘nmod’, ‘vmod’and ‘jjmod’correspond to the adverb modifier, noun modifier, verb modifier and adjective modifier respectively. Below the noun modifier, we have the noun dependencies of r6 (possession) and relc (relative clause). Similarly, below the verb modifier we have the verb arguments, which are the karaka labels, k1, k2, k3 and so on.

3 Dependency Relations

The relations in the scheme are marked between chunk heads. The verb in simple sentence generally becomes the head of the sentence. The arguments of the verb are shown with appropriate labels. Figure 3(a),(b),(c) shows this for verbs ‘velwAdu’, ‘koVsw Adu’, ‘iccAdu’ respectively. Likewise, noun becomes the head in the case of genitives etc. This can be seen in Figure 3(d).

a. rAmudu hyderabad ki velwAdu.
   ‘Ram’ ‘hyderabad’ ‘to’ ‘go_will’.
   Ram will go to hyderabad.
b. rAmudu cAku wo paMdu koVswAdu.
   ‘Ram’ ‘knife’ ‘with’ ‘fruit’ ‘cuts’.
   Ram cuts the fruit with knife.
c. rAmudu sIwa ki apple iccAdu.
   ‘Ram’ ‘Sita’ ‘to’ ‘apple’ ‘gave’.
   Ram gave an apple to Sita.
d. rAmudi yoVkka puswakaM.
   ‘Ram’ ‘’s’ ‘book’.
   Ram’s book.

[8] See http://ltrc.iiit.ac.in/MachineTrans/research/tb/dep-tagset.pdf

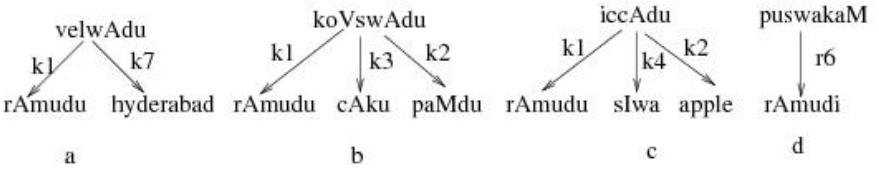


Fig. 3. Dependency Relations

The relation used for relative clauses is nmod_relc. Conjunct relations are treated as distinct from normal dependency relations. In this scheme [2] the conjuncts become the head. This is true for both coordinating & subordinating conjuncts.

4 Issues

In this section, we describe some issues we encountered while annotating the sentences and show how we analyzed them.

4.1 Genitives

Genitive is the case that marks a noun as modifying another noun. The relation between the Genitive noun and its head noun is denoted by "r6". In Telugu this can be exhibited broadly in two ways::

- Using explicit Genitive marker "yoVkka"

  rAmudi yoVkka puswakaM.
  'Ram' ''s' 'book'.
  Ram's book.

- Genitive marker is dropped::

  In Telugu generally the masculine nouns[9] have a possessive marker "i" indicating an implicit genitive marker. And in case of feminine nouns[10] however this relationship must be inferred.

rAmudi puswakaM	sIwa puswakaM
'Ram' '-'s' 'book'	'Sita' 'book'
Ram's book	Sita's book

  Yet, some masculine nouns (where the the lexical item and it's root are identical) also exhibit this property.

  raGu puswakaM.
  'raGu' 'book'.
  Raghu('s) book.

[9] Ram is a masucline noun.
[10] Sita is a feminine noun.

Decision: If the genitive marker is not present, the manual annotator will have to infer the relation based on the context. Initial inter-annotator agreement is high which suggests that native Telugu speakers can easily identify this relation based on the context.

4.2 Conjuncts

In Telugu, conjuncts can occur as suffixes, lexical items and as DheerGaas[11]

- Suffixes:: Conjuncts occur as TAM[12] of the verb.

  nenu iMtiki vel*we* nidrapowAnu.
  'I' 'house_to' 'go_if' 'sleep_will'.
  I will sleep if I go home.

  Decision: They are treated as *vmod of the type subordinating conjuncts* as shown in Figure 4.

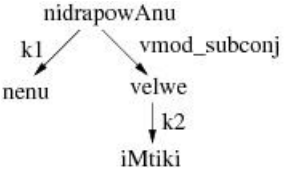


Fig. 4. Suffixes

- Lexical items:: They occur as *mariyu (and) , kAni (but)* etc...

  rAmudu iMtiki vellAdu mariyu mohana mArket ki vellAdu.
  'Ram' 'house_to' 'went' 'and' 'mohan' 'market_to' 'went'.
  Ram went home and Mohan went to the market.

  Decision: Handling simple coordinating conjuncts is straight forward. Figure 5 shows that their analysis in Telugu is consistent with the Hindi annotation.

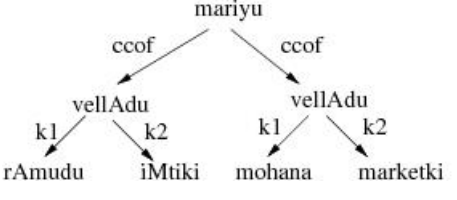


Fig. 5. Lexical Items

[11] DheerGaas are elongated forms of the vowels at the end of the lexical items [12].
[12] TAM - Tense Aspect Modality.

- DheerGaas:: By elongating the vowel at the end of the lexical items, the information of conjunction is implicit.

  rAmudU sIwa iMtiki vellAru.
  'Ram' '-*and*' 'Sita' 'home_to' 'went'.
  Ram and Sita went home.

  Decision: A null element with a special tag, NULL_CCP[13], is introduced.

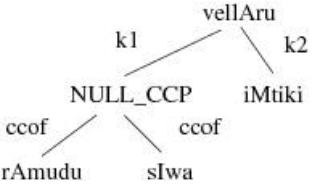


Fig. 6. dheergas

4.3 Copula

Copula is a linking verb. It is generally dropped in Telugu, unlike Hindi and English in which it takes the form "hE" and "be" respectively.

rAmudu maMci bAludu.
'Ram' 'good' 'boy'.
rAma accA laDakA hE[14].
Ram is a good boy.

Decision: An element with tag NULL_VG is introduced inorder to fit the criteria of the dependency schema which states that the root of a dependency tree[15] is a main verb.

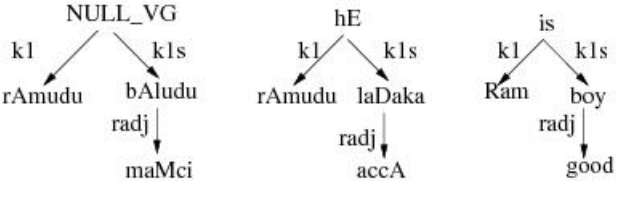


Fig. 7. Copula

The following are some more examples in which a verb is missing. NULL_VG inserted sentences are also given.

[13] When one or more element from a sentence is dropped, it is called ellipses. A null element marked with a special tag 'NULL' is introduced in such cases. Note that without inserting a NULL the tree cannot be drawn. NULL_NP, NULL_VG, NULL_CCP etc mark different kinds of ellipses.

[14] The sentence is a Hindi sentence.

[15] k1s stands for k1 samanadhikaran which means of equal status to k1.

1. rAmudu bAludu mariyu allari pillavAdu.
   ‘Ram’ ‘boy’ ‘and’ ‘mischievous’ ‘kid’.
   Ram is a boy and a mischievous kid.
   rAmudu bAludu mariyu allari pillavAdu NULL_VG.
2. rAmudu maMci bAludu mariyu pallu wiMtAdu.
   ‘Ram’ ‘good’ ‘boy’ ‘and’ ‘fruits’ ‘eats’.
   Ram is a good boy and eats fruits.
   rAmudu maMci bAludu NULL_VG mariyu paMdu wiMtAdu.

4.4 “ani" Constructions

There are broadly two different senses for the lexical item *‘ani’*.

a. As a complementizer (that):
   rAmudu pallu wiMtAdani mohana ceVppAdu.
   ‘Ram’ ‘fruits’ ‘eat_will’ ‘-that’ ‘mohan’ ‘told’.
   Mohan told that Ram eats fruits.
b. As a subordinating conjunct:
   rAmudu wanani vellamannAdani mohana vellipoyAdu.
   ‘Ram’ ‘him’ ‘to_go’ ‘-told’ ‘-because’ ‘mohana’ ‘went’.
   Mohan went because Ram told him to go.

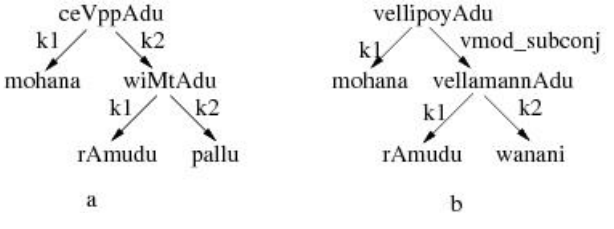


Fig. 8. “ani" Constructions

The following example and its dependency structure shown in Figure 9 covers almost all the above mentioned cases:

rAmudu maMci bAludani, raGu puwakAlu caxuvuwAdani sIwa ceVpwuMxi.
‘Ram’ ‘good’ ‘boy’ ‘-that’ ‘,(and)’ ‘Raghu’ ‘reading’ ‘books’ ‘-that’ ‘sIwa’ ‘says’.
Sita says that Ram is a good boy and reads Raghu’s books.

5 Parsing Issues

In this section, we shall look at how the above discussed cases will be problematic in parsing.

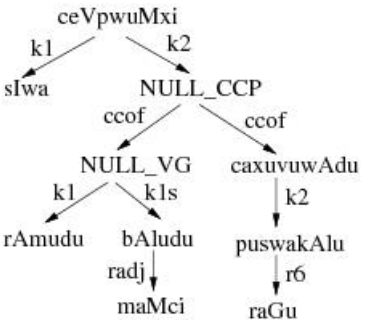


Fig. 9. All Cases

5.1 Genitives

As discussed earlier due to the potentiality of dropping the genitive marker, the sentences become ambiguous.

raGu puswakaM rAmudiki iccAdu.
'Raghu' 'book' 'Ram' '-to' 'gave'.

The above sentence is ambiguous. It's two different interpretations are shown below.

a. Raghu's book was given to Ram [by somebody].
b. Raghu gave a book to Ram.

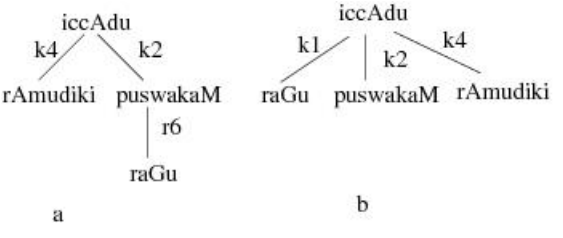


Fig. 10. Genitive

At the sentence level we cannot predict the sense among the two. It needs contextual reasoning. Hence, for a grammar-driven approach it will be wise to generate the possible two parses as shown in the Figure 10. Later a prioritizer will select the most appropriate parse based on relevant contextual features [5].

5.2 Conjuncts

For suffixes, a transformation frame [6] for the TAM, ***'we'***, corresponding to the verb frame [3] for the verb *vellu* repairs the dependency tree. For lexical items, the demands

will be met by the conjunct frame[16]. But there is a problem in identifying the conjunction in case of Dheergas. The main problem here is "how to insert a NULL_CCP" in the sentence. A heuristic to resolve the problems:

> If the vowel elongated lexical item and the word[17] succeeding to it are of the same POS category insert a NULL_CCP between those words and this proves to be true for 98.6% cases.

In the example given, rAmudu and sIwa, both are proper nouns and hence NULL_CCP is inserted. But in the sentence "rAmudu (Ram) kApI (coffee) wAgAdu (drank)" it won't be inserted because rAmudu is a proper noun whereas kApI is a common noun. A coordinating conjunct preserves the category of the words it conjoins.

5.3 Copula

The main problem here is "how to insert a NULL_VG" in the sentence. Below we state two possible heurisics to overcome this problem. Insert a NULL_VG if:

1. There is no main verb in the sentence.
2. If there is a clause end marking lexical item, like *ani*, and if that clause doesn't contain any verb. This heuristic fails if we consider the free word orderness of the language.
3. Once NULL_VG is inserted, we can check for proximity to identify k1 and k1s which are the potential children for NULL_VG, though there are cases where this fails too.

We need to address this issue in detail (more importantly, verb missing in a clause) as the above heuristics does not work all the time.

6 Conclusion and Future Work

In this paper we have introduced an ongoing effort to annotate Telugu sentences with dependency relations. We stated the motivation behind following the Paninian framework in the Indian language scenario. We discussed different cases where we came up with some generalizations for annotations. We also showed and discussed why and how these cases are problematic in parsing in perspective of a grammar-driven approach. In the future our major goal is to increase the number of annotated sentences in the treebank.

Along with that we wish to start exploring the treebank in terms of understanding which features of the language play a vital role in parsing from the perspective of Machine Learning. We are trying to adopt a two-stage constraint based parsing architecture for Telugu.

[16] See [4], [5] for details on the conjunct frames and how they are handled in a two stage parsing architecture.

[17] Words and lexical items are used interchangeably.

Acknowledgement

We would like to thank Ganga Bhavani, D V Sriram, Phani Gadde, Bharat Ambati for developing parts of the treebank.

References

1. Bamman, D., Crane, G.: The design and use of a Latin dependency treebank. In: Proc. of TLT 2006, pp. 67–78. FAL MFF UK, Prague (2006)
2. Begum, R., Husain, S., Dhwaj, A., Sharma, D., Bai, L., Sangal, R.: Dependency annotation scheme for Indian languages. In: Proc. of IJCNLP 2008 (2008)
3. Begum, R., Husain, S., Sharma, D.M., Bai, L.: Developing Verb Frames in Hindi. In: Proc. of LREC 2008, Marrakech, Morocco (2008)
4. Bharati, A., Husain, S., Sharma, D.M., Sangal, R.: A Two-Stage Constraint Based Dependency Parser for Free Word Order Languages. In: Proc. of the COLIPS IALP 2008, Chiang Mai, Thailand (2008)
5. Bharati, A., Husain, S., Sharma, D.M., Sangal, R.: In: Proc. of IWPT 2009, Paris (2009)
6. Bharati, A., Chaitanya, V., Sangal, R.: Natural Language Processing: A Paninian Perspective, pp. 65–106. Prentice-Hall of India, New Delhi (1995)
7. Bhatt, R., Narasimhan, B., Palmer, M., Rambow, O., Sharma, D.M., Xia, F.: A Multi-Representational and Multi-Layered Treebank for Hindi/Urdu. In: Proc. of TLT 2009 (2009)
8. Brants, S., Dipper, S., Hansen, S., Lezius, W., Smith, G.: The TIGER Treebank. In: Proc. of TLT 2002 (2002)
9. Bosco, C., Lombardo, V.: Dependency and relational structure in treebank annotation. In: Proc. of Workshop on Recent Advances in Dependency Grammar at COLING 2004 (2004)
10. Hajicova, E.: Prague Dependency Treebank: From Analytic to Tectogrammatical Annotation. In: Proc. TSD 1998 (1998)
11. Hudson, R.: Word Grammar. Basil Blackwell, 108, Cowley Rd, Oxford, OX4 1JF, England (1984)
12. Krishnamurti, B., Gwynn, J.P.L.: A grammar of modern Telugu. Oxford University Press, Delhi, New York (1985)
13. Mel'cuk, I.A.: Dependency Syntax: Theory and Practice. State University Press of New York (1988)
14. Marcus, M., Santorini, B., Marcinkiewicz, M.A.: Building a large annotated corpus of English: The Penn Treebank. In: Computational Linguistics (1993)
15. Rambow, O., Creswell, C., Szekely, R., Taber, H., Walker, M.: A dependency treebank for English. In: Proc. of LREC 2002 (2002)
16. Shieber, S.M.: Evidence against the contextfreeness of natural language. Linguistics and Philosophy, 8, 334–343 (1985)
17. van der Beek, L., Bouma, G., Malouf, R., van Noord, G.: The Alpino dependency treebank. In: Computational Linguistics in the Netherlands (2002)
18. Vaidya, A., Husain, S., Mannem, P., Sharma, D.M.: A karaka-based dependency annotation scheme for English. In: Gelbukh, A. (ed.) CICLing 2009. LNCS, vol. 5449, pp. 41–52. Springer, Heidelberg (2009)

EusPropBank: Integrating Semantic Information in the Basque Dependency Treebank

Izaskun Aldezabal[1], María Jesús Aranzabe[1], Arantza Díaz de Ilarraza[2], Ainara Estarrona[2], and Larraitz Uria[3]

IXA NLP Group
[1] Basque Philology Department,
[2] Languages and Information Systems
University of the Basque Country
[3] IKER UMR 5478, University of Pau and Pays de l'Adour (UPPA), CNRS
{izaskun.aldezabal,maxux.aranzabe,a.diazdeilarraza,
ainara.estarrona,larraitz.uria}@ehu.es

Abstract. This paper deals with theoretical problems found in the work that is being carried out for annotating semantic roles in the Basque Dependency Treebank (BDT). We will present the resources used and the way the annotation is being done. Following the model proposed in the PropBank project, we will show the problems found in the annotation process and decisions we have taken. The representation of the semantic tag has been established and detailed guidelines for the annotation process have been defined, although it is a task that needs continuous updating. Besides, we have adapted AbarHitz, a tool used in the construction of the BDT, to this task.

Keywords: Theoretical Problems in Semantic Annotation, Representation of Semantic Roles, Lexical Resources.

1 Introduction

The construction of a corpus with annotation of semantic roles is an important resource for the development of advanced tools and applications such as machine translation, language learning and text summarization. We present here the work that is been carried out for annotating semantic roles in the BDT. Our previous work on semantics has mainly focused on word senses (including the development of the Basque WordNet and Basque Semcor (Agirre et al., 2006a), building verbal models from corpora, including selectional preferences (Agirre et al., 2003) and subcategorization frames (Aldezabal et al., 2003), as well as manually developing a database with syntactic/semantic subcategorization frames for a number of Basque verbs (Aldezabal, 2004).

Our interest follows the current trend, as shown by corpus tagging projects such us the Penn Treebank (Marcus, 1994), PropBank (Palmer et al., 2005) and PDT (Hajic et al., 2003), and the semantic lexicons that have been developed alongside them, suche as VerbNet (Kingsbury et al., 2002) and Vallex (Hajic et al., 2003). FrameNet (Baker et al., 1998) is also an example of the joint development of a semantic lexicon and a hand-tagged corpus.

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 60–73, 2010.

After a preliminary study, we chose to follow the PropBank/VerbNet model for a number of reasons:

- The PropBank project starts from a syntactically annotated corpus, just as we do.
- The organization of the lexicon is similar to our database of verbal models.
- Given the VerbNet lexicon and the annotations in PropBank, many implicit decisions on problematic issues, such as the distinctions between arguments and adjuncts have been settled and are therefore easy to replicate when we tag the Basque data.
- Having corpora in different languages annotated following the same model allows for cross-lingual studies and hopefully the enriching of Basque verbal models with the richer information currently available for English.

In fact, the PropBank model is being deployed in other languages, such as Chinese, Spanish, Catalan and Russian. Palmer and Xue (2003) and Xue (2008) describe the Chinese PropBank. Civit et al. (2005) describe a joint project to annotate comparable corpora in Spanish, Catalan and Basque.

The paper will be organised as follows: after a brief introduction, we will present the resources used in the semantic tagging. Section 3 explains the steps followed in the annotation, the automatic procedures defined to facilitate the task of manual annotation. In section 4, we describe the tool used for tagging (AbarHitz) while section 5 discusses theoretical problems and decisions we are facing. Finally, section 6 presents the conclusions and future work.

2 The Resources Used

In this section we will present the PropBank/VerbNet model, the model followed, and the resources we have for the annotation of semantic roles. We will explain them briefly, more details can be found in Aldezabal (2007) and Agirre et al. (2006b).

2.1 PropBank/VerbNet

PropBank is a corpus that is annotated with verbal propositions and their arguments. In the PropBank model two independent levels are distinguished: the level of arguments and adjuncts, and the level of semantic roles. The elements that are regarded as arguments are numbered from *Arg0* to *Arg5*, expressing semantic proximity with respect to the verb. The lowest numbers represent the main functions (subject, object, indirect object, etc.). The adjuncts are tagged as *ArgM*.

With regard to roles, PropBank uses two kinds: roles specific to each specific verb (e.g. buyer, thing bought, etc.), and general roles (e.g. agent, theme, etc.) linked to the VerbNet lexicon (Kipper et al., 2002).

VerbNet is an extensive lexicon where verbs are organized in classes following Levin's classification (1993). The lexicon provides an association between the syntactic and semantic properties of each of the described verbs.

Table 1 shows the PropBank roleset for the verb 'go.01' and the corresponding VerbNet roleset with Levin's class number (go-47.7 51.1-2).

Table 1. PropBank and VerbNet rolesets of the verb 'go'

PropBank go.01	VerbNet go-47.7 51.1-2
Arg1: entity in motion/goer	Theme
Arg2: extent	
Arg3: start point	Source
Arg4: end point	Destination
ArgM: medium	
ArgM: direction (usually up or down)	

A verb equivalent to the English *go* should have a similar roleset. Table 2 shows a preliminary version for the roleset of the Basque verb *joan.01* (= 'go') based on the roleset in table 1. VerbNet roles are more general and sometimes, as the examples show, more simple. As a first approach, we decided to use the VerbNet1.0 roles (and when the tagging task required we would add the missing ones) because it is more similar to our in-house database. We will only mention the VerbNet roles in the rest of the paper, together with the argument number.

Table 2. Preliminary version of the lexical entry for *joan.01* (='go').

joan.01
Arg1: Theme
Arg3: Source
Arg4: Destination

Table 3 shows the argument numbers, the VerbNet roles and the syntactic functions which are usually associated with the numbered arguments and adjuncts in PropBank:

Table 3. The argument numbers, the roles and the syntactic functions usually associated with the numbered arguments and adjuncts in PropBank.

Arguments	**VerbNet roles**	**Syntactic function**
Arg0	agent, experiencer	subject
Arg1	patient, theme, attribute, extension	direct object, attribute, predicative, passive subject
Arg2	attribute, beneficiary, instrument, extension, final state	attribute, predicative, indirect object, adverbial complement
Arg3	beneficiary**,** instrument, attribute, cause	predicative, circumstantial complement
Arg4	destination	adverbial complement
Adjuncts		
ArgM	location, extension, destination, cause, time, manner, direction	adverbial complement

We have gathered the information contained in PropBank and VerbNet (VerbNet 1.0) in a single data base. The information contained in this data base is used when applying the automatic procedure.

2.2 The BDT Corpus

For our task we will use the Basque Dependency Treebank (BDT). The Basque Dependency Treebank was built on EPEC, a corpus that contains 300,000 words of standard written texts which is intended to be a training corpus for the development and improvement of several NLP tools (Bengoetxea and Gojenola, 2007). Around one third of this collection was obtained from the *Statistical Corpus of 20th Century Basque* (http://www.euskaracorpusa.net). The rest was sampled from *Euskaldunon Egunkaria* (http://www.egunero.info) a daily newspaper. EPEC has been manually tagged at different levels: morphosyntax, syntactic phrases, syntactic dependencies (BDT) and WordNet word senses.

2.3 The EADB Resource (Data Base for Basque Verbs)

The work done in Aldezabal (2004), which includes an in-depth study of 100 verbs for Basque from EPEC, is our starting point. Aldezabal defined a number of syntactic-semantic frames (SSF) for each verb. Each SSF is formed by semantic roles and the declension case that syntactically performs this role. The SSFs that have the same semantic roles define a coarse-grained verbal sense and are considered syntactic variants of an alternation. Different sets of semantic roles reflect different senses. This is similar to the PropBank model, where each of the syntactic variants (similar to a frame) pertains to a verbal sense (similar to a roleset).

Aldezabal defined a specific inventory of semantic roles; the set of semantic roles associated with a verb identifies the different meanings of that verb. The semantic roles specified are: Theme, Affected Theme, Created Theme, State, Location, Time, End Location, End State, Start Location, Path, Startpoint, Endpoint, Experiencer, Cause, Source, Container, Content, Feature, Activity, Measure, Manner. In addition, Aldezabal identified a detailed set of types of general predicates to facilitate the classification of verbs from a broad perspective in such a way that the meaning of the verbs is expressed from a cognitive point of view. The predicates are the following: Change of State of an Entity, Change of Location of an Entity, Change of an Entity, Creation of an Entity, Activity of an Entity, Interchange of an Entity, To contain an Entity, Assignment of a Feature to an Entity, Existence of an Entity, Location of an Entity, State of an Entity, Description of an Entity, Expression of a Suppisition.

We show an example of an EADB verb entry:

joan.1 ('go'): entity in motion
affected theme_ABS[1]; startpoint / path_ABL; endpoint_ALA
joan.2 ('go'): entity in motion
affected theme_ABS; startpoint [+animate]_DAT; endpoint_ALA
joan.3 ('go'): feature that disappears from an entity
container_DAT; content [-animate, -concrate]_ABS

2.4 Mapping between Basque and English Verbs Based on Levin's Classification

In Aldezabal (1998), English and Basque verbs are compared based on Levin's alternations and classification. For this purpose, all of the verbs in Levin (1993) were translated first considering the semantic class and then paying attention to the similarity of the syntactic structure of verbs in English and Basque. The main advantage of having linked the Basque verbs to Levin classes comes from the fact that other resources like PropBank and VerbNet lexicon are linked to Levin classes and contain information about semantic roles. Verbs in a Levin class have a regular behaviour (according to diathesis alternation criteria), different from verbs belonging to other classes. Also de classes are semantically coherent and verbs belonging to one class share the same semantic roles. In Table 4, we present some examples of these links.

Table 4. The link between verbs in Levin (1993) and Basque

glower	40.2	bekozko/kopetilun begiratu
glue	22.4	erantsi, kolatu
gnash	40.3.2	hortzak karraskatu
go	47.7	joan
go	51.1	joan
gobble	38	glu-glu egin
gobble	39.3	irentsi
goggle	30.3	liluratu moduan begiratu
gondola	51.4.1	gondolaz ibili/joan/eraman

3 The Annotation Process

When constructing BDT, we followed a Dependency Parsing Syntactic Formalism which provides a straight forward way for expressing semantic relation. The process of manual annotation of semantic roles associated to verbs will begin with the tagging of the most frequent verbs contained in the corpus (approximately 30% of all verb occurrences correspond to 10 verbs) and studied in (Aldezabal, 2004). The sentences of the corpus are grouped according to the verbs they have.

[1] ABS, ABL, ALA and DAT are the absolutive, ablative, adlative and dative cases respectively.

We don't annotate light and modal verbs that will be treated deeply later. That is the case of *egin* (='do') and *izan* (='be'), which are the two most frequent verbs in the corpus.

Once we finish the 100 verbs, we will continue with the rest of verbs, in the way we will explain in the methodology.

We carry out this work by means of the following phases:

1. The preprocessing phase: comparison of the Levin classes in our mapping and the PropBank data-base. As explained before, we have the English equivalent of a Basque verb in terms of Levin class so we were able to obtain automatically the PropBank/VerbNet information for each treated verb from the paid data-base, basing on Levin class.

However, we have to update our mappings since our mapping was done, some time ago, PropBank has changed and, consequently, new classes and subclasses have been added, erased and modified. We performed an automatic revision of our previous mappings and distinguished the four different situations, explained below:

- **equal**: represents the case in which the identification of the class for a verb has not changed since the mapping was done. For instance, *say* and *go* continue being in the 37.7 and 47.7 classes respectively. This option represents 51% of the cases.
- **subclass**: a new subclass has been defined in PropBank. For example, the verb *go* in the 51.1 class in our mapping has been redefined as 51.1-2 in PropBank. In these cases, we directly equalized the subclass with the general class, and maintain the mapping. (6%)
- **changed**: a Levin class in PropBank has changed and there is not a direct coincidence between our mapping and the one in PropBank. For instance, the class 45.6 for the verb *increase* has been changed in PropBank (2%)
- **missing:** the verb is not included in PropBank or it has not assigned any Levin class. For instance, the verb goggle is not in PropBank (41%)

In Table 5 we present the result of this automatic comparison for some of the verbs contained in Table 4. The first column in Table 5 shows the English verb, the second column corresponds to Levin's class, the third column presents the definition of the verb in Basque and the fourth one specifies to which group the mapping belongs.

Table 5. A sample of the results of the comparison between our mapping and PropBank, regarding Levin classes

glower	40.2	bekozko/kopetilun begiratu	MISSING
glue	22.4	erantsi, kolatu	EQUAL
glutenize	45.4		MISSING
gnash	40.3.2	hortzak karraskatu	MISSING
gnaw	39.2		MISSING
go	47.7	joan	EQUAL
go	51.1	joan	SUBCLASS
gobble	38	glu-glu egin	EQUAL
gobble	39.3	irentsi	EQUAL
goggle	30.3	liluratu moduan begiratu	MISSING
gondola	51.4.1	gondolaz ibili/joan/eraman	MISSING

We decided to deal with the first and second cases (those verbs detected as "equal" and "subclass") that cover the 46% of the EPEC corpus, leaving the rest to future study. We are refining our algorithm to see if it is possible to detect automatically more equivalences.

2. Establishing the tagging criteria. Three linguists tag 50 occurrences of the same verb for each of the verbs fixed in the first step. This step has the objective of obtaining the guidelines for the annotation.

3. Semiautomatic tagging. Again, three linguists tag 20 different occurrences of the same verb (60 occurrences in all). Once (at least) 60 occurrences of these verbs are tagged we begin with the rest of occurrences by means of automatic procedures. Throughout the process the guidelines are updated.

For the rest of the verbs, we will prepare an automatic pre-tagging process based on lexical models obtained from the tagged corpus. Features such as Verb, VNrol, Valence and Selectional Restriction will be taken into account. In Aldezabal (2001) and Zapirain et al. (2008), we have carried out some experiments in which different methods for role inference are proposed for English verbs.

3.1 Representation of the Semantic Information (Definition of the Tag)

From the set of dependency relations associated to a clause, we will take those relations that are candidates to be arguments or adjuncts of the verb[2] We denominate the semantic tag defined "arg_info" and it is composed by the following fields (explained in the order of appearence):

- **VN** (VerbNet/PropBank verb): the English verb and its PropBank number in "VerbNet-PropBank". As it is usual to find more than one verb in the same category, we put the necessary ones separated by the slash. Example: tell_01 / say_01.
- **V** (Verb): the main verb which acts as the head of the relation.
- **Treated Element** (TE): the element depending from the head that will be the adjunct or the argument.
- **VAL** (valence): value that identifies arguments or adjuncts: arg0, arg1, arg2, arg3, arg4, argmod.
- **VNrol** (role in VerbNet): those represented in Table3.
- **EADBrol** (semantic role according to EAD roleset). We can see an enumeration of them in Table 4.
- **HM** (Selectional Restriction). Up to now we only consider [+animate], [-animate], [+count], [- count], [+hum], [-hum]

Figure 1 shows a compound sentence syntactically annotated, where a semantic annotation has been added to the phrase in adlative (ALA) linked to the verb *joan*. We can see that the sentence is divided into phrases and that each phrase has a dependency relation (e.g. ncmod for prepositional phrase) with respect to the verb

[2] The relations considered are: ncsubj, ncobj, nczobj, ncmod, ncpred (non-clausal subject, object, indirect object, …), ccomp_obj, ccomp_subj, cmod (clausal finite object, subject, modifier), xcomp_obj, xcomp_subj, xcomp_zobj, xmod, xpred (clausal non-finite object, subject, indirect object, …).

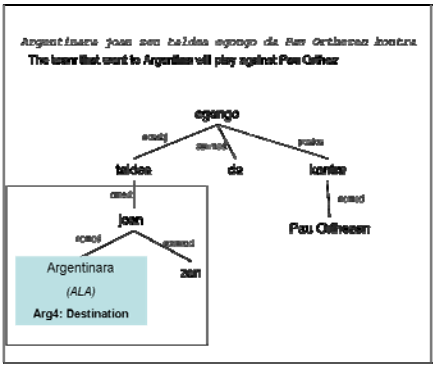


Fig. 1. A syntactically and semantically annotated clause in Basque

(joan). Syntactic dependencies[3] are marked on the links, and the semantic information in the nodes. Declension case has been included in the nodes as additional information.

The example (1) illustrates the arg_info tag that corresponds to the relation highlighted in Figure 1.

(1) arg_info: (go_01, joan, *Argentinara*[4], Arg4, Destination, end_location, -[5].)

4 AbarHitz, the Tool for Tagging

AbarHitz (Díaz de Ilarraza et al., 2004) is a tool designed to help the linguists in the manual annotation process of the BDT. AbarHitz has been implemented to assist during the definition of dependencies among the words of the sentence.

Similar tools have been implemented with the same aim as the AbarHitz; Annotation Graph Toolkit (AGTK) (Bird et al., 2002), TREPIL Treebanking Interface (Rosén et al., 2005) are some examples. It is important to emphasize that the design of Abar-Hitz follows the general annotation schema we established for representing linguistic information and it is part of a general environment we have developed so far in which general processors and resources have been integrated.

Let us first of all describe the tool in general terms and then we will explain how it is appropriate for the semantic annotation presented here.

Abar-Hitz communicates with the user by means of a friendly interface providing the following facilities:

(1) It visualizes the morphosyntactic information obtained so far and which, for our specific corpus, have previously been manually disambiguated. The tool is able to simultaneously use outputs from several tools (a morphological parser, a POS tagger and a syntactic parser) to guide the annotator's decisions.

---

[3] *cmod* is the relative clause; *auxmod* is the auxiliary verb; *ncsubj* is the noun-clause subject; and *postos* is an auxiliary tag to express a complex postposition.

[4] to Argentina (PP)

[5] When we are not sure of a value or we think it is not necessary to define it, we put the null mark ("-").

(2) It graphically visualizes the dependency-tree for each sentence. In addition, the tree drawn can be graphically manipulated in such a way that the user can change the tags and their fields, roll up sub-trees, remove/add nodes, remove/add connectors (dependencies) and so on.
(3) It provides an environment for syntactic checking while tagging. We have to take into account that mistakes can be made while tagging in the number and type of slots, and the name of the tag itself. Abar-Hitz keeps away from these mistakes by showing specific pop-up menus where the only thing the linguist can do is to select the appropriate tag.

Figure 2 shows the main window of Abar-Hitz in which we can identify:

- **sentence selection area** (in the right side of the figure). In the top part the linguist specifies the verb; in the example the verb joan (to go) has been selected. Below the specification area, a list of the files containing the selected verb is given. The annotator can select one of the files to proceed with the annotation. At the side, the system also maintains a record of the status of the annotation process indicating for each sentence whether: i) the annotation has been completed or not; ii) the annotation sentence is not clear enough and some aspects must be discussed, and so on.
- **text area** (upper left). When the annotator clicks on one the files listed, the sentence is shown in the upper part of the window highlighted.
- **tagging area (left side)**. The tree visualizer is activated by clicking on the corresponding icon.

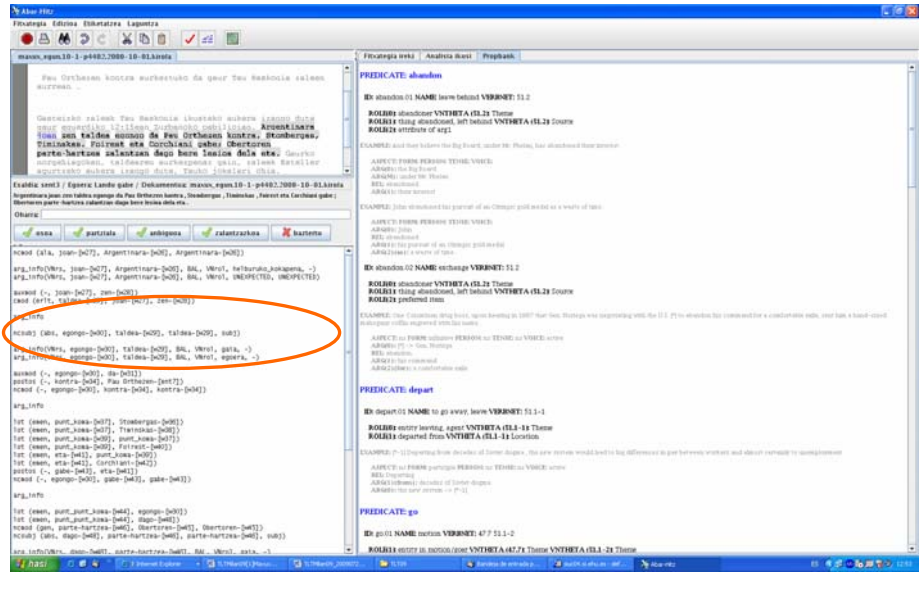


Fig. 2. Visualizing the information of PropBank/VerbNet (right side) to the human annotator. On left side arg_info tag proposed to be fullfilled by the annotator.

4.1 Adapting AbarHitz to the Tagging of Semantic Roles

A recent enhancement of AbarHitz facilitates the semantic annotation by offering the linguist new options:

(1) It provides the information associated with the verb being tagged, contained in PropBank and VerbNet. Figure 2 shows an example of this functionality, which is made explicit in two ways: i) by displaying in the right part of the window information from PropBank/VerbNet; and ii) by giving the corresponding information in the arg_info relation as seen in section 3.4.
(2) It provides new "incomplete" "arg_info" relations to be fulfilled by the annotator. We say "incomplete" because some of the arguments of the relation have been automatically obtained while others remain unspecified. Although the system doesn't provide all the "arg_info" relation complete, the approach has been proved to be very helpful to the linguists. Figure 3 shows, on the left side, the syntactic annotation of the sentence and the semantic tag "arg_info" associated to the verb under study (*joan*) fulfilled by the annotator.

Abar-Hitz has been developed in Java; it follows a modular design in order to be a portable and easily maintainable tool. It runs under the Microsoft Windows, Linux and Unix environments.

5 Theoretical Problems and Decisions

We tagged about 37,000 words of the corpus and analyzed 32 verbs (27% of the overall corpus). We consider for tagging only some of the most frequent verbs (those which appeared in the EADB). We confirmed that the most ambiguous a verb, the more problems and criteria have to be defined.

Then, we have defined general criteria for the tagging process. Structured and detailed set of guidelines for taggers and lexicon editors have been defined (Aldezabal et al., 2010). However, it is a task that needs continuous updating, as new verbs are analyzed.

Let us mention some of the problems defined and decisions taken during this process:

- When the correspondence to the PropBank model(s) can be established automatically, it happens that this association is not always complete and consistent. A (Basque) verb can be linked to more than one PropBank verb. In such cases, we have to check, first of all, whether the rolset-number, the role and the arguments in both languages are the same or not.

In case they are equivalent, there is no doubt for tagging: we assign the corresponding verb. For example, the verb *esan* can be linked unquestionably with tell_01 and say_01. We establish the correspondence and we indicate this double equivalence by the expression tell_01/say_01 as first value of arg_info tag. If, on the contrary, the roles and arguments are not the same, we specify the two verbs in the first field (for example: take_04/bring_01) and select the most suitable argument structure one after examining syntactic behavior of both English and Basque verbs.

- When the correspondence to the PropBank model(s) can not be established automatically, we try to find the information in other sources (Verb-Index http://verbs.colorado.edu/verb-index/index.php), make the corresponding inference about its argument structure and roleset and update our databases.

The following example illustrates this problem: the verb *jokatu* ("to bet") is not linked because our algorithm has not established *jokatu* as an equivalent of "to bet". In this case, the steps followed will be:

1. To get the argument-structure of "to bet" in PropBank
   Roleset id: bet.01 , *wager*, vncls: 54.5 94
   Roles:
   Arg0: *better*
   Arg1: *amount of bet*
   Arg2: *basis, proposition, bet on*
   Arg3: *co-better*
2. To look at Verb-Index we can see "to bill", "to rely" and "to risk" have similar behavior
3. To look at the roles of the appropriate one, in this case, "to bill"
   Agent: [+animate / + organization]
   Asset: [+currency]
   Recipient: [+animate / +organization]
   Cause:
4. To make the corresponding inference linking argument and role
   Arg0: Agent
   Arg1: Asset
   Arg2: theme
   Arg3: recipient

Another example to illustrate the difficulty in finding the adequate correspondence can be seen when studying the Basque verb *eskatu* (= "to ask"), we find that none of the equivalents given by the system correspond to the sense we are looking for. In this case, the argument structure of the English verb doesn't agree with the one included in EADB, so, we have to specify a new sense in the EADB data-base. In the case of the verb *eskatu* (= "to ask"), *ask_02* could the appropriate equivalent but its argument structure does not match with the one specified in EADB. The verb *ask_02* in PropBank and VerbNet, contains 3 arguments: Arg0: Agent, Arg1: Theme (proposition) and Arg2: Patient.

However, the verb "eskatu" contains only 2 arguments in EADB: Arg0: *esperimentatzailea* (experiencer) and Arg1: *gaia* (theme). Besides, it is said that the DAT (dative) argument is optional although it is not included within the subcategorized cases (this argument fits with Arg2: Patient in PropBank).

We decide to follow the PropBank model and change our data base. Example (2) shows a sentence that illustrates the final annotation linked to the argument structure of *eskatu.*

Example (2):
Nemesiok, joan baino lehen, Alejandro adiskideari eskatzen dio, zaindu dezala bere "x" zakurra
(Before leaving, Nemesio asks his friend Alejandro to look after his "x" dog)

arg_info (ask_02, eskatzen, Nemesiok, arg0, Agent, ...)
arg_info (ask_02, eskatzen, lehen, argM, TMP, -, -)
arg_info (ask_02, eskatzen, adiskideari, arg2, patient, ...)
arg_info (ask_02, eskatzen, zaindu, arg1, Theme, gaia, -biz.)

We do not follow the same procedure in all cases. For example, in the case of the verb lortu ("to obtain"), the Arg2 definition of PropBank for DAT cases , will be tagged as ArgM.

- Where the value of an item of the relation is not clear or when it has not any corresponding value, we use the symbol "-".
- We do not tag verbs as part of locutions. For example we will leave the tagging process of the roles linked to the verb *joan*[6] in the expressions, *usotara doa*[7], *desarmea aurrera badoa*[8] to a subsequent step.
- When VerbNet assings two different roles to the same argument, we have decided to base on EADB and to assign the corresponding roles of VerbNet roles. For example, we have found it in the case of the verb *ikusi* ("to see"). In EADB the verb *ikusi* contains two arguments and a role is assigned to each of the arguments:

  Arg0: *esperimentatzailea* (experiencer)
  Arg1: *gaia* (theme)

In PropBank/VerbNetThat assigns two roles to those arguments: Arg0 has associated "agent" and "experiencier" roles and Arg1, "theme" and "stimulus". In this ambiguous case, we use EADB information. The result would be:

Arg0: Agent, *esperimentatzailea*
Arg1: theme, *gaia*

6 Conclusions

We have presented the work being carried out on the annotation of semantic roles in the BDT, a dependency-based annotated Treebank. Some automatic and manual procedures have been developed in order to facilitate the annotation process. The idea is to present the human taggers with a pre-tagged version of the corpus.

From what we have analyzed up to now, we conclude that the PropBank model is suitable for treating Basque verbs, but, of course, cross-linguistic studies always have to cope with to difficult tasks when performing semantic mapping between verbs in different languages.

Structured and detailed set of guidelines for taggers and lexicon editors have been defined. However, it is a task that needs continuous updating.

Our database of verbal models was a good starting point for the tagging task. We detected some differences with English verbs regarding the status of arguments and adjuncts, due to different basic criteria, but those can be easily adjusted. Our database is stricter on arguments, while PropBank has a wider perspective.

[6] In general "to go".
[7] To go to hunt pigeons.
[8] If disarmament goes on.

Our study confirms that building a lexicon and tagging a Basque corpus with verbal sense and semantic role information following the VerbNet/PropBank model of PropBank is feasible but not lacking in problems. We have also shown the method for integrating our pre-existing resources into this new framework.

In the future we want to focus on the application of automatic methods for role tagging. We have seen that once a verb is tagged with a certain number of appearances, the resulting lexicon can be used to automatically tag the rest of the appearances. Previous experimentation (Aldezabal et al., 2003) shows us that, in some cases, we can automatically tag up to 82% of the occurrences of a verb and leave a small proportion of occurrences for manual tagging.

However, we want to stress that the automatic tagging is not a substitute for manual tagging. We plan to review all occurrences, regardless of whether they remain ambiguous or no.

Acknowledgments

This work has been partially funded by the Education Department of the Spanish Government (EPEC-RS project, HUM2004-21127-E) and (IMLT, TIN2007-63173).

References

Agirre, E., Aldezabal, I., Pociello, E.: A pilot study of English Selectional Preferences and their Cross-Lingual Compatibility with Basque. In: International Conference on Text Speech and Dialogue, Czech Republic, pp. 12–19 (2003)

Agirre, E., Aldezabal, I., Etxeberria, J., Izagirre, I., Mendizabal, K., Pociello, E., Quintian, M.: A methodology for the joint development of the Basque WordNet and Semcor. In: Proceedings of the 5th International Conference on Language Resources and Evaluations (LREC), Genoa, Italy (2006a)

Agirre, E., Aldezabal, I., Etxeberria, J., Pociello, E.: A Preliminary Study for Building the Basque PropBank. In: Proceedings of the 5th International Conference on Language Resources and Evaluations (LREC), Genoa, Italy (2006b)

Aldezabal, I.: Levin's verb classes and Basque. A comparative approach, UMIACS Departmental Colloquia. University of Maryland (1998)

Aldezabal, I., Aranzabe, M., Atutxa, A., Gojenola, K., Sarasola, K., Goenaga, P.: Extracción masiva de información sobre subcategorización verbal vasca a partir de corpus. In: Actas del XVII Congreso de la SEPLN, vol. 27, pp. 29–36. Universidad de Jaen, Spain (2001)

Aldezabal, I., Aranzabe, M.J., Atutxa, A., Gojenola, K., Oronoz, M., Sarasola, K.: Application of finite-state transducers to the acquisition of verb subcategorization information. Natural Language Engineering 9, 39–48 (2003)

Aldezabal, I.: Aditz-azpikategorizazioaren azterketa. 100 aditzen azterketa zehatza, Levin, oinarri harturik eta metodo automatikoak baliatuz. Leioa (Bilbao): University of Basque Country thesis (2004)

Aldezabal, I.: Estudio preliminar para la creación de Euskal PropBank. In: Castellón, I., Fernández, A. (eds.) Perspectivas de análisis de la unidad verbal, SERES. Universitat de Barcelona, Spain (2007)

Aldezabal, I., Aranzabe, M.J., Díaz de Ilarraza, A., Estarrona, A., Fernández, K., Uria, L.: EPEC-RS: EPEC (Euskararen Prozesamendurako Erreferentzia Corpusa) rol semantikoekin etiketatzeko eskuliburua [Guidelines to tag semantic roles in the EPEC corpus (the Reference Corpus for the Processing of Basque)]. Internal Report, UPV / EHU / LSI / TR 02-2010 (2010)

Baker, C.F., Fillmore, C.J., Lowe, J.B.: The Berkeley FrameNet project. In: Proceedings of the COLING-ACL, Montreal, Canada (1998)

Bengoetxea, K., Gojenola, K.: Desarrollo de un analizador sintáctico-estadístico basado en dependencias para el euskera [Development of a statistical parser for Basque]. Procesamiento del Lenguaje Natural 39, 5–12 (2007)

Bird, S., Maeda, K., Ma, X., Lee, H., Randall, B., Zayat, S.: TreeTrans: Diverse Tools Built on The Annotation Graph Toolkit. In: Third International Conference on Language Resources and Evaluation, Las Palmas, Canary Islands, Spain, pp. 29–31 (2002)

Civit, M., Aldezabal, I., Pociello, E., Taulé, M., Aparicio, J., Màrquez, L.: 3LB-LEX: léxico verbal con frames sintáctico-semánticos. In: XXI Congreso de la SEPLN, Granada, Spain (2005)

Díaz de Ilarraza, A., Garmendia, A., Oronoz, M.: Abar-Hitz: An annotation tool for the Basque Dependency Treebank. In: Paper presented at the International Conference on Language Resources and Evaluation, Lisbon, Portugal (2004)

Hajic, J., Panevová, J., Urešová, Z., Bémová, A., Kolárová, V., Pajas, P.: PDT-VALLEX: Creating a Largecoverage Valency Lexicon for Treebank Annotation. In: Proceedings of the Second Workshop on Treebanks and Linguistic Theories, Sweden, pp. 57–68 (2003)

Kingsbury, P., Palmer, M.: From Treebank to PropBank. In: Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC 2002), Las Palmas, Spain (2002)

Kipper, K., Palmer, M., Rambow, O.: Extending PropBank with VerbNet Semantic Predicates. In: Workshop on Applied Interlinguas, held in conjunction with AMTA 2002, Tiburon, CA (2002)

Levin, B.: English Verb Classes and Alternations. A preliminary Investigation. The University of Chicago Press, Chicago (1993)

Marcus, M.: The Penn TreeBank: A revised corpus design for extracting predicate argument structure. In: Proceedings of the ARPA Human Language Technology Workshop, Princeton, NJ (1994)

Xue, N.: Labeling Chinese predicates with semantic roles. Computational Linguistics 34(2), 225–255 (2008)

Palmer, M., Xue, N.: Annotating the Propositions in the Penn Chinese Treebank. In: Proceedings of the Second Sighan Workshop, Sapporo, Japan (2003)

Palmer, M., Gildea, D., Kingsbury, P.: The Proposition Bank: A Corpus Annotated with Semantic Roles. Computational Linguistics Journal 31(1) (2005)

Rosén, V., Smedt, K.D., Dyvik, H., Meurer, P.: TREPIL: Developing Methods and Tools for Multilevel Treebank Construction. In: Civit, M., Küber, S., Martí, M. (eds.) Proceeding of the Fourth Workshop on Trebank and Linguistics Theories, pp. 161–172. Universitat de Barcelona, Spain (2005)

Zapirain, B., Agirre, E., Màrquez, L.: Robustness and Generalization of Role Sets: PropBank vs. VerbNet. In: Proceedings of the 46th Annual Meeting of the Association of Computational Linguistics, ACL 2008: HLT, Columbus, Ohio, pp. 550–558 (2008)

Morphological Annotation of a Corpus with a Collaborative Multiplayer Game

Onur Güngör and Tunga Güngör

Department of Computer Engineering, Boğaziçi University,
Bebek, 34342 Istanbul,Turkey
{onurgu,gungort}@boun.edu.tr

Abstract. In most of the natural language processing tasks, state-of-the-art systems usually rely on machine learning methods for building their mathematical models. Given that the majority of these systems employ supervised learning strategies, a corpus that is annotated for the problem area is essential. The current method for annotating a corpus is to hire several experts and make them annotate the corpus manually or by using a helper software. However, this method is costly and time-consuming. In this paper, we propose a novel method that aims to solve these problems. By employing a multiplayer collaborative game that is playable by ordinary people on the Internet, it seems possible to direct the covert labour force so that people can contribute by just playing a fun game. Through a game site which incorporates some functionality inherited from social networking sites, people are motivated to contribute to the annotation process by answering questions about the underlying morphological features of a target word. The experiments show that the 63.5% of the actual question types are successful based on a two-phase evaluation.

1 Introduction

In most of the natural language processing tasks, state-of-the-art systems usually rely on machine learning methods for building their mathematical models [1]. Given that the majority of these systems employ supervised learning strategies, a corpus that is annotated for the problem area is essential.

But having a relevantly annotated corpus is not enough on its own. The corpus must have a number of crucial features. First, it must include a set of carefully selected examples so that the method can train the model without bias. For the training to be successful, the corpus must include a set of examples. The size of the set is mainly determined by the characteristics of the training method itself. In addition to be sufficient for training, the corpus must not introduce bias to the trained model. Second, the corpus must be free of errors. While some methods may be resistant to several kinds of errors in the corpus, in most cases the errors prevent the method from training the model to its maximum extent.

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 74–85, 2010.

When we recognize the crucial value of an error-free corpus with a vast number of examples in solving natural language processing tasks, the task of building a corpus with these properties gains importance. The most prominent method of building corpora today is to divide the work among experts and wait for them to finish their work [2]. However, it can be argued that this method is flawed in a number of points. First of all, this method dictates that the people who work on the work units must be experts in their field. Furthermore, they must be trained for this task. However, finding and training an expert is costly and time consuming. Even if we were successful in finding and hiring experts to work on building the corpus, there are other things that hinder the process. For example, the annotation patterns of two experts -even if they are highly experienced in the area- may be very different resulting in inconsistent annotation. We can expect to observe this situation especially in small and spontaneous annotation projects, where experts do not work in pairs and do not correct inconsistencies introduced by other experts later.

As a result of these problems, the process of building a corpus with the current methods is slow and expensive, if not low quality. This in turn impacts the rate of natural language processing research as well as its scope. This paper recognizes this problem as an important hindrance to the further development of natural language processing research and proposes a new method for building corpora.

We chose the morphological disambiguation of Turkish as the target domain. Morphological disambiguation problem is to select the correct morphological parse of a word in a given context among all of the possible parses of a word. We had two reasons for selecting this domain. First, this problem is at the core of other Turkish natural language processing tasks, i.e. parsing, speech recognition and sense tagging to name a few. Second, we have access to a corpus already tagged, which enabled us to test our results. In fact, the annotated corpus [3] is one of the very few annotated corpora in Turkish.

In this paper, we propose a novel system which incorporates a collaborative game for the morphological disambiguation of Turkish language. The game addresses the issues stated above and has two modes, one with a single player, where quiz-like questions are answered; the second is a two person game where one tries to explain a concealed word to the other, meanwhile answer some questions that are valuable for our annotation needs. The game is open to anyone and hosted on a publicly accessible web server.

We continue with the related literature on the subject in Section 2. Section 3 describes the game and the overall system that encapsulates the game in detail. Section 4 describes the experiment's setup and the obtained results. In Section 5, we draw conclusions and discuss some further research topics to be pursued.

2 Related Work

In [4], a game in which players are matched up with each other randomly and expected to win points by matching their inputs when viewing the same image simultaneously is described. Given that no other means of communication is

possible, the most obvious thing to input is the most distinctive figure in the image. It posed a nice challenge, this caused people to have a lot of fun and some of them eventually grew an addiction which lead to a very effective and fast way of labeling images on the web. This is the seminal work which introduced the idea of turning particular problems into games that people enjoy by harvesting the "wasted human-cycles"[1].

Later games by Luis von Ahn further extended the idea to various areas. Peekaboom [5] utilizes the idea to mark the portions of the images that depict target labels. Phetch [6] collects text descriptions of images by making one player describe the image and a group of players to simultaneously guess from the set of images they are confronted by a search engine result. Verbosity [7] collects facts about objects again by exploiting the collaborative game play method explained before. In Verbosity, one player tries to get the other player to guess the secret word that is exposed to her. Clues to the other user are given with predefined sentence templates like "it contains __". When the blanks are filled with appropriate content, this input conveys a fairly significant description about the secret word. Last game that Ahn designed is Tagatune [8]. It aims to transform the work of tagging music clips into a game. It works much like ESP Game. But it seems like it could not be that successful mainly because it is difficult to agree on a common word to describe the clip and listening to a sound could take a bit and become boring.

In [9], a method for collecting alternative forms of phrases, namely paraphrases is discussed. For achieving their goal, they develop a web site where people cooperate. The most important component of the system is their partial hinting system. By default, they already have 2-3 paraphrases. But they want to increase this number. This is achieved with partial hints. At the start of the game, no hint is given and users are expected to enter paraphrases of their own. If they are able to guess the already known paraphrases, this contributes to the confidence of that paraphrase. Otherwise, the contribution is stored as a new paraphrase to be guessed by other contributors. This much like resembles social bookmarking sites in which each contribution is accumulated and more submissions of the same contribution reinforces the importance of it. After guessing a paraphrase, if it is unsuccessful, the partial hinting mechanism reveals 33 per cent of the already obtained paraphrases like "this ... help". In [10], five design decisions are introduced. First, it is important to fine tune templates which will collect semantic information (abstract morphological data in our case). Besides fine tuning, it is necessary to provide guidance to users. It is also advisable to break the annotation process into several steps to be able to distribute the work among users. This way multiple users can validate the annotations. Also it would be good to have a way to automatically repair the contributions at least to some extent.

In [11], it is suggested to have a reward mechanism, which is not only instant rewards after successful annotation but also awards points when another player makes the same annotation at some future time.

[1] A term coined by Luis von Ahn to refer to the term "CPU cycles".

A semi-collaborative approach to corpus annotation is described in [12]. But the system simply acts as a data repository that can be accessed simultaneously both online or offline ([13] is also similar in this way). This makes the system miss the collaboration possibility. However, a well thought mechanism is implemented: the contributors are presented with a readily annotated text which is output by a program which accomplishes the task that the collected corpora will help developing programs for. We think this can be further extended to incorporate active learning in the system.

A work by Gülşen Eryiğit [14] describes a standalone (non-web) program which can be used as a tool for dedicated contributors. Relying on specially trained people to annotate the corpus is destined to be slow and costly, despite the increase in speed by using this tool.

In [15], several users can annotate the corpus individually, and later one "consensus user" selects the best annotation. Thus, we think the cooperation aspect of the project is not incorporated by design. Additionally, contribution requires specialized knowledge in the area and no ordinary user can help readily.

As our focus in the paper is to build an unambigously annotated corpus for morphological disambiguation of Turkish, we would like to list some of the current approaches to the problem. A trigram-based statistical model is presented in [16]. In [17], a decision list induction algorithm is introduced for performing morphological disambiguation. There are also several constraint-based methods for disambiguation [18,19]. Another method employs a perceptron algorithm for morphological disambiguation [20]. We use the tool produced by this study as a morphological parser ranging from preparing the corpus to the online question generation.

3 The Game

We continue with elaborating on the crucial properties which the game must possess. First of all, the game must be playable by ordinary people who are not necessarily educated in the field. This means that we have to find a way to break up the disambiguation process into pieces to be able to tailor the process for non-experts.

At this point, we assume that humans are equipped with a covert ability to sense the correct parse of the word. This ability is learned in the childhood but there is no known way of consistently describing this ability so that it can be programmed to be executed on computers. Thus it seems reasonable to generate all possibilities with a morphological parser and then somehow make the user select the correct parse. One problem here is that these parses cannot be directly understood by a person without knowledge on the subject. Given the facts that humans covertly "know" to separate the good parses form the bad parses and that the raw parses are not sufficiently clear, we find it useful to form questions acting as an abstraction layer between the user and the raw parses. Thus, we propose to discard bad parses from the set of parses by asking questions of two types; yes/no questions and multi-option questions. These questions must be

prepared so that they are automatically generated for any word in the corpus and be clearly understood by the users. By asking this question to a statistically sufficient number of users, we became assured whether the parses that are to be discarded will be discarded or not.

Possibly there will be other questions, because one question will discard only a portion of the set of all possible parses. However, after aggregating the users' answers for these questions, we will have discarded all the bad parses. This means that we have finished disambiguation and left with the correct parse.

In conclusion, our game is capable of generating questions for the words in the corpus automatically. These questions are asked in several stages of both the single and two player game. After aggregating sufficient number of answers, the correct parse of the corpus word is detected.

An additional aspect of the game is that it must be publicly accessible by our target population. To provide this, we chose to host the game on a web site which is accessible at any time of the day and without device restriction. One can access the site by just having the standard equipment which is used to browse the web, namely web browsers. Moreover, we allow people to access our game without formal introduction or qualification tests. This is unlike the previous corpus annotation efforts in which nearly all of them require their contributors to be known and recognized by the people responsible with the process. If we recall that they also usually require the contributors to come to a special office where the work is done, the advantage of our approach is recognized better. In summary, we host the game on a publicly accessible site and allow anyone to join and start the annotation. This in turn makes the potential level of participation (thus work accomplished) much higher than the previous annotation methods. If we take into account that the Internet is maybe the most frequently utilized time killing activity, we can assume this potential to grow even more.

Motivation of the users is another issue which is very closely related with the game design and the site that it is contained. We have two basic notions for building and nourishing motivation.

The first is fun. If the game is fun enough, people will begin to grow an addiction to the game instead of other time spending activities which sometimes can be boring in themselves. To provide the fun element to the game, we introduce a special stage in the game. This stage contains similar elements from Taboo and a famous game in which you try to explain some film title to the audience without speaking. As you might recall, in Taboo, similar to the game about explaining film titles, you are trying to convey a specific concept to the audience without using some words which are prohibited from using -even parts of it. This stage of the game, we call it as the taboo stage for simplicity, is activated only when playing the two player game. One of the users are chosen as the teller and the other as the guesser. The objective of the teller is to give clues about some specific word to the guesser to accomplish her own objective which is to guess the word as fast as possible. The word that is to be conveyed is actually a word in its sentence context. The sentence is shown to both players. But, obviously, the word in question is concealed from the guesser. The two players enjoy a sense

of cooperation while the teller gives clues and the guesser tries word after word. At the same time, they are challenged with a time limit that keeps them alive and attached to the game.

The other aspect of the game which is thought to increase motivation is competition. Naturally, people tend to compete with other people when challenged with a fairly hard problem. The key point here is to design the game so that it is neither too hard nor too easy. We employed several methods for building motivation. The run against the time limit in Stage 2 is itself a competitive factor. In that stage, players compete against the time cooperating with the other player. This forms the basic motivation for the game. Another method is to build motivation by introducing competition based on group membership. This idea is based on the fact that it is known that people form around groups to enjoy group membership advantages. These advantages can vary from just declaring that someone is a member of a prestigious group to gaining benefits for themselves by using the connections among the group. The site which the game is embedded provides users a way to create and join groups as they wish. People can create groups to represent their school, their football team or a way of thinking. People can also do this for completely arbitrary groups. When a group is created, anyone who wants to join is allowed, and as a result the points that are earned by that user are added to the total points of the group. Competition among the groups are thus constituted. We expect to see the total motivation to build up as a result of this competition.

Another dimenson of the competition factor in the game is to focus on individual representation. As it can be guessed, besides group membership, people pay attention to keep their online presences in a state which is desirable by other people. And to do that, people may want to devote a lot of time to earn high points in a game if the result is to be presented to a lot of audience as a highly skilled person. Thus, in order to exploit this behaviour, we present the highest scoring ten users on the home page of the game site. We assume that people will be motivated to get into that list.

3.1 Single Player Game

In single player game mode, the player is first shown a sentence from the corpus. One of the words in the sentence is marked with a distinctive color, namely red. The player is asked a question that is designed to detect a morphological feature of the indicated word. The answer of the player is stored, the player is awarded 50 points, and the game advances. The next stage is actually the same as the previous stage but this time another word from another sentence is selected and displayed along with its context. The game continues until it is ended by the player herself.

The target words are selected so that it is made sure that every type of question gets a statistically significant number of answers. To make the player answer in a reasonable time, there is a time limit on this stage which was set to two minutes during the experiment.

3.2 Two Player Game

Before starting a two player game, the system matches two users who indicate that they are willing to join a two player game session. After a pair is matched up, they are registered for the same game session. The game session consists of games that are played consequently. The rules of winning a game session is that you have to win all the ten games in a row. If you are not able to win a game in the process, you are not allowed to go to the next game and as a result the game session ends.

We call one of the players as "the teller", the other as "the guesser" throughout a game.

A game of two player mode consists of three stages:

1. the question is asked to the teller
2. the taboo stage
3. the question is asked to the guesser

Stage 1 is basically the same with the single game mode which is explained in Section 3.1. The answer submitted by the player is stored and the player is awarded 50 points. Then, the game advances to the next stage. Meanwhile, the guesser waits for the teller to answer the question while the game displays the same sentence but the target word is concealed. This is to warm up the guesser to Stage 2 and help her to build up some excitement instead of waiting tediously.

In Stage 2 which we call the taboo stage, the same sentence and the indicated word is shown to the teller. But the guesser still does not see the concealed word. The objective of this stage is to operate collaboratively to guess the word as quick as possible. Through an interface which they can communicate simultaneously, the teller tries to give as many clues as possible while the guesser acts upon these clues to guess the target word.

The interface for the teller is different from the interface of the guesser. While the guesser can only utilize a single text box to submit her guesses, the teller's interface contains much more text boxes (see Figure 1). There are a total of nine boxes which the teller can fill with clues. However, each of these boxes differ in the meaning they convey when used. The first box is for clues that are input in free form. While it would be sufficient for the communication between the users, we design the remaining boxes so that each of them reflects another semantic relation between the clue input and the target word itself. We call them clue templates.

The motivation behind these additional text boxes is to gather more fine-grained information about the target word. In fact, we see this is a side effect of the proposed game. A game feature which we add to make the game fun turns out to be helpful for another purpose in the end. This extra information about the word itself possibly can be used for sense tagging. We include the actual decriptive text on two of these clue templates and the meanings associated in Table 1. The points you get is higher if you use the text boxes which correspond to semantic relations. The actual numbers are 5 to 50 points which indicates a factor of ten between the two numbers.

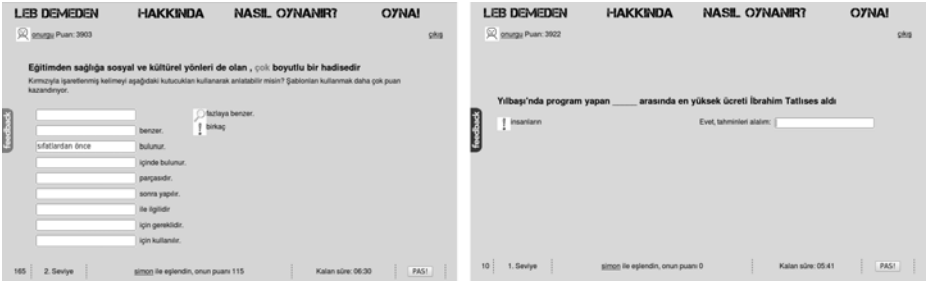


Fig. 1. Teller and Guesser Interfaces (respectively)

Table 1. Clue Templates

Clue Template	Semantic Relation	Description
____ benzer.	Similarity	Defines a similarity between two objects.
____ bulunur.	LocationOf	Location information.

We had to implement a filter to prevent cheating using these boxes. If we recall the experience obtained from previous work, the participants in these kind of games that offer you fame and some kind of identity representation medium often try to cheat to get those awards more easily (see [4]). The filtering mechanism works like this: First it is checked whether the clue text as a whole can be found in the text of target word, if it is found, the clue is discarded. If it is not, it is checked whether the text of target word can be found in the clue text, if it is found, the clue is discarded, otherwise the clue is accepted. When the clue is discarded, it is not shown to the other user not even partly.

While the interfaces for the teller and the guesser differ generally, there is indeed a widget which is common to both of them. This widget displays the conversation between the teller and the guesser in a sequential manner. As a new guess or clue is submitted, the widget is updated.

We chose a time limit of ten minutes for this stage. This limit is intended to encourage participation in fear of not being able to complete the stage. As you might expect, this stage continues until either the time limit expires or the pair succeeds in guessing the word correctly. Regardless of the situation, we advance to the next stage. However, if they could not guess the target word, the whole game session finishes after the next stage. Each guess from the guesser receives 10 points. Each free text clue is awarded by giving out 5 points. However, if the clue is submitted using the clue templates, the teller earns 50 points. When the pair successfully guess the target word, they receive 500 points.

In the third and the last stage of this game, the guesser is exposed the same question as the teller in Stage 1. None of the settings differ from Stage 1. Basically, the stage is designed to guarantee obtaining answers from different people for each question. After Stage 3 is finished, the game session goes on with another game if the target word is guessed successfully in Stage 2. If the number of consequent games that were successful reaches ten, we say that the game session

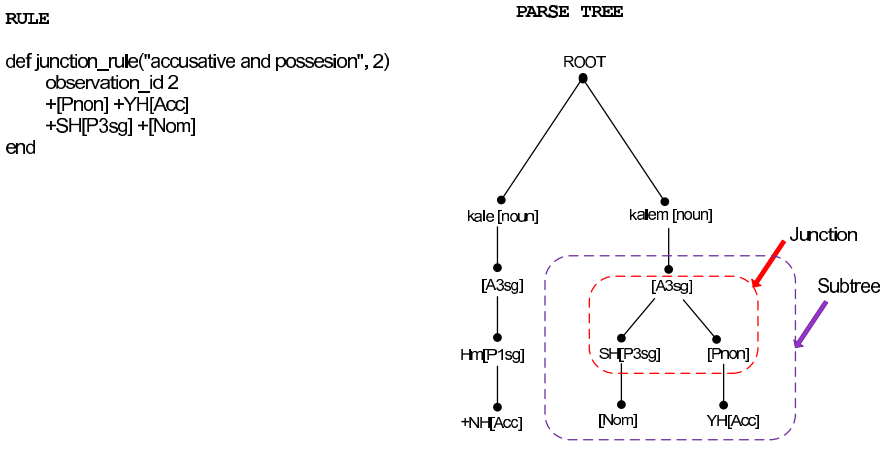


Fig. 2. A Junction Rule and the Corresponding Subtree

finishes successfully and the pair is taken back to the game lounge with a greeting note. As a result of this row of winning games, they are both awarded 5000 points. On the other hand, in case Stage 2 was unsuccessful, the game session is finished and they receive no points.

4 Results

The experiment had been done through a game site which is accessible publicly on the web[2]. While it is continuing its operation, we only use the data collected between 29 June 2009 and 9 July 2009, approximately 6000 answers.

According to our experiment plan, we prepared two lists of each having an instance of 74 possible question types and collected about 30 answers for each of them. By doing this, we were able to assess the quality of the questions over two instances, calling the first as Phase 1, and the second as Phase 2. We had to resort to this plan because after speculating on the expected number of visitors, we calculated that it could be infeasible to evaluate our method on the basis of complete disambiguation.

To understand the success criterion of a question, we must first explain the question generation methodology. To generate a question, we first start with enumerating the set of all morphological parses of the word by using a morphological analyzer. We then transform it into a tree. After this transformation, the detection of junction points by observation rules results in abstract objects called observations (see Figure 2). These observations are then matched with question rules. Each matched question rule is applied to the word to generate the unique questions which are tailored solely for determining the correct way to choose in the junction that is represented by the observation. After 30 people answer the question, we agree on the option with most submissions. We verify

[2] http://lebdemedenleblebi.com

this agreement answer by checking whether the correct parse reported in the corpus contains the resolution parse tag that is attached to each option.

We calculate the rate of successful questions in Phase 1 as 79.7 per cent. This figure is realized as 71.6 per cent in Phase 2. However, we want to report that a little modification to the definition of a successful question would increase these values to 87.8 per cent and 79.7 per cent. This modification would be to discard the answers of type 'None' or 'I did not understand the question' if they are the highest ones. We observed that this modification increases the rates but in any way we did not change the evaluation method so that to allow an elaboration. When we look at the combined results of these two phases, we see that the percentage of question types that are successful in both of these phases is 63.5 per cent.

There were 400 users registered on the site at the end of the experiment period. A total of 5284 games were played of which 4784 of them was in single player mode. Although the total number of clue templates were utilized only to a certain extent, the users who employed them used 3 templates on average. Experiments show that average time required to answer a question in Stage 1 or 3 took around 36 seconds and the most of the pairs completed Stage 2 well below 60 seconds as can be seen in Figure 3.

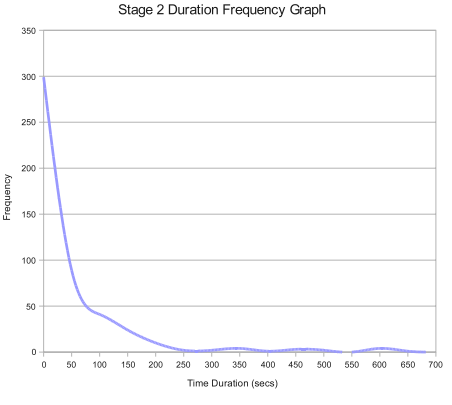


Fig. 3. Stage 2 Duration Frequency Graph

5 Conclusion and Future Work

In this work, a game for morphological annotation of a Turkish corpus is developed. This is the first work that incorporates human computation methods in corpus annotation. The game is meant to be played by two players simultaneously over the Internet. Basically, the annotation is done by collecting answers to questions that are automatically created based on a number of templates prepared manually. In one of the three stages of the two player game mode, one of the players has to describe the target word to the other player trying to collaboratively guess the word as fast as possible. The answers to the questions

posed in the other stages are then analyzed statistically and an aggregation of aggreement answers is built which in turn results in a complete morphological disambiguation.

The game is hosted on a publicly accessible web site. The results reported in the paper are compiled from the data obtained between 29 June 2009 and 9 July 2009. The evaluation was done by assesing the performance of all question types over two instances. The reported success rate over the two phases is 63.5%.

As a future work, we see that incorporating an awarding system that can measure the performance of the players and award accordingly can be more facilitating. Also, a method for measuring the difficulty of a question or at least categorizing them by hand would enable us to modify the game so that the levels become harder and harder, thus making the game more challenging. Another important future work is to host the game in a site with a high number of daily visitors to test our method in a real setting and succeed in a complete disambiguation of arbitrary text.

References

1. Marquez, L., Salgado, J.G.: Machine learning and natural language processing (2000)
2. 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)
3. Yüret, D.: Morphologically tagged corpus (2009)
4. von Ahn, L., Dabbish, L.: Labeling images with a computer game. In: CHI 2004: Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 319–326. ACM, New York (2004)
5. von Ahn, L., Liu, R., Blum, M.: Peekaboom: a game for locating objects in images. In: CHI 2006: Proceedings of the SIGCHI conference on Human Factors in computing systems, pp. 55–64. ACM, New York (2006)
6. von Ahn, L., Ginosar, S., Kedia, M., Liu, R., Blum, M.: Improving accessibility of the web with a computer game. In: CHI 2006: Proceedings of the SIGCHI conference on Human Factors in computing systems, pp. 79–82. ACM, New York (2006)
7. von Ahn, L., Kedia, M., Blum, M.: Verbosity: a game for collecting common-sense facts. In: CHI 2006: Proceedings of the SIGCHI conference on Human Factors in computing systems, pp. 75–78. ACM, New York (2006)
8. Law, E.L.M., von Ahn, L., Dannenberg, R.B., Crawford, M.: Tagatune: A game for music and sound annotation. In: ISMIR 2007: 8th International Conference on Music Information Retrieval (2007)
9. Chklovski, T.: Collecting paraphrase corpora from volunteer contributors. In: K-CAP 2005: Proceedings of the 3rd international conference on Knowledge capture, pp. 115–120. ACM, New York (2005)
10. Chklovski, T., Gil, Y.: Improving the design of intelligent acquisition interfaces for collecting world knowledge from web contributors. In: K-CAP 2005: Proceedings of the 3rd international conference on Knowledge capture, pp. 35–42. ACM, New York (2005)

11. Richardson, M., Domingos, P.: Building large knowledge bases by mass collaboration. In: K-CAP 2003: Proceedings of the 2nd international conference on Knowledge capture, pp. 129–137. ACM, New York (2003)
12. Bontcheva, C.T., Cunningham, H., Tablan, V., Bontcheva, K., Dimitrov, M., Lab, O.: Language engineering tools for collaborative corpus annotation. In: Proceedings of Corpus Linguistics 2003, pp. 80–87. Wiley, Chichester (2003)
13. Ma, X., Lee, H., Bird, S., Maeda, K.: Models and tools for collaborative annotation. In: Proceedings of the Third International Conference on Language Resources and Evaluation. European Language Resources Association, Paris (2002)
14. Eryiğit, G.: ITU treebank annotation tool. In: Proceedings of the Linguistic Annotation Workshop, Prague, Czech Republic, June 2007, pp. 117–120. Association for Computational Linguistics (2007)
15. Stührenberg, M., Goecke, D., Diewald, N., Mehler, A., Cramer, I.: Web-based annotation of anaphoric relations and lexical chains. In: Proceedings of the Linguistic Annotation Workshop, Prague, Czech Republic, June 2007, pp. 140–147. Association for Computational Linguistics (2007)
16. Hakkani-Tür, D.Z., Oflazer, K., Tür, G.: Statistical morphological disambiguation for agglutinative languages. In: Proceedings of the 18th conference on Computational linguistics, Morristown, NJ, USA, pp. 285–291. Association for Computational Linguistics (2000)
17. Yuret, D., Türe, F.: Learning morphological disambiguation rules for turkish. In: HLT-NAACL 2006 (June 2006)
18. Oflazer, K., Tur, G.: Combining hand-crafted rules and unsupervised learning in constraint-based morphological disambiguation. In: Brill, E., Church, K. (eds.) Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 69–81. Association for Computational Linguistics, Somerset (1996)
19. Oflazer, K., Tür, G., Tfir, G.: Morphological disambiguation by voting constraints. In: Proceedings of the Thirty-Fifth Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics, pp. 222–229 (1997)
20. Sak, H., Güngör, T., Saraçlar, M.: Morphological disambiguation of Turkish text with perceptron algorithm. In: Gelbukh, A. (ed.) CICLing 2007. LNCS, vol. 4394, pp. 107–118. Springer, Heidelberg (2007)

Computational Models of Language Acquisition

Shuly Wintner

Department of Computer Science, University of Haifa
Mount Carmel, 31905 Haifa, Israel
shuly@cs.haifa.ac.il

Abstract. Child language acquisition, one of Nature's most fascinating phenomena, is to a large extent still a puzzle. Experimental evidence seems to support the view that early language is highly formulaic, consisting for the most part of frozen items with limited productivity. Fairly quickly, however, children find patterns in the ambient language and generalize them to larger structures, in a process that is not yet well understood. Computational models of language acquisition can shed interesting light on this process. This paper surveys various works that address language learning from data; such works are conducted in different fields, including psycholinguistics, cognitive science and computer science, and we maintain that knowledge from all these domains must be consolidated in order for a well-informed model to emerge. We identify the commonalities and differences between the various existing approaches to language learning, and specify desiderata for future research that must be considered by any plausible solution to this puzzle.

1 Introduction

Language acquisition is one of Nature's most fascinating puzzles. Human languages are extremely complex systems, yet (most) children acquire them naturally, fairly quickly and seemingly effortlessly [24, p 144]. Research in language acquisition attempts to study the mechanisms of this puzzle in order to explain the very nature of language itself: the primary cognitive capacity which makes us human.

Two competing theories of language acquisition dominate the linguistic and psycholinguistic communities [60, pp. 257-258]. One, the *nativist* approach, originating in Chomsky [21, 22, 23] and popularized by Pinker [50], claims that the linguistic capacity is innate, expressed as dedicated "language organs" in our brains; therefore, certain linguistic universals are given to language learners for free, requiring only the tuning of a set parameters in order for language to be fully acquired. The other, *emergentist* explanation [8, 61, 42, 47, 43, 44, 60], claims that language emerges as a result of various competing constraints which are all consistent with general cognitive abilities, and hence no dedicated provisions for universal grammar are required. Consequently, "[linguistic universals] do not consist of specific linguistic categories or constructions; they consist of general ... cognitive abilities" [59, p. 101]. Furthermore, language is first acquired in an *item-based* pattern: "[young children] do not operate on the basis of any linguistic abstractions, innate or otherwise. Fairly quickly, however, they find some patterns in the way concrete nouns are used and form something like a category of a

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 86–99, 2010.

noun, but schematization across larger constructions goes more slowly. The process of how children find patterns in the ambient language and then construct categories and schemas from them is not well understood at this point." [59, pp. 106-107].

Computational models can shed new light on language acquisition processes and provide new insights into the nativist vs. emergentist debate. Three related fields of research address grammar induction from data [4]: "applied" grammar induction in linguistics; empirical (computational) grammar induction; and formal (mathematical or logical) grammar induction. Adriaans and van Zaanen [4] conclude that "it is time to remove the (artificial) boundaries and combine the research performed within each subfield." In this paper we survey several approaches to language learning from data and assess their contribution to a clearer picture of child language acquisition. We review existing work with an eye to consolidating the "applied" and "empirical" approaches to grammar induction. We then propose future research directions that aim at a better explanation of early stages of human language acquisition.

This paper is structured as follows. Section 2 defines the language learning task in a general way, indicating in what ways different approaches may vary. Section 3 discusses the difficult issue of evaluating the performance of models and algorithms that address such tasks. We then survey existing approaches in three classes: works whose motivation is to explain child language acquisition from a cognitive perspective (Section 4.1); works that aim to devise an efficient mechanism for inducing (formal) grammars from raw data (Section 4.2); and a few recent works that try to consolidate some aspects of the two approaches (Section 4.3). Finally, we propose in Section 5 some directions for future research that consolidate the benefits of rigorous computational models backed up by solid psycholinguistic findings.

2 The Language Learning Task

The task that we focus on is language learning: a *learner*, be it a child or a computer program, is presented with *data*, in the form of raw utterances. To approximate the fact that human learning is grounded in real-world situations, the raw data are sometimes annotated with part-of-speech (POS) categories, syntactic information or even semantic information. The learner's task is to generalize the data and induce a model of the grammatical utterances (in other words, a *grammar*). In the formal sense, a grammar is a generative device that defines a set of expressions, its *language*, and induces some structure (e.g., trees) on expressions in the language. A grammar can be expressed explicitly as a set of rules, perhaps with probabilities attached to them; or implicitly as a set of "operations" on strings, with or without "slots" (which are the equivalent of non-terminal symbols in a formal setting). The success of the learner can be evaluated by testing its grammar on new utterances. Two aspects of the grammar can be tested: its ability to generate new utterances; and its ability to assign a valid structure to the grammatical utterances.

Language learning tasks rely on the existence of large text corpora that document language use, both for training and for evaluation [25, 49, 31]. Computational linguistic tasks standardly use manually-annotated sentences from the Penn Tree Bank (PTB,

Marcus et al. [48]), whose data are taken from the Wall Street Journal. Grammar induction tasks are sometimes limited to a subset of the PTB, where sentences are limited to length of 10 or less (WSJ10). Clearly, this is a genre that is not well-defined linguistically, and quite likely irrelevant for language acquisition investigations.

In order to investigate the development of child language, corpora which document linguistic interactions involving children are needed. The CHILDES database [41], a 300MB corpus in over 25 languages, contains transcripts of spoken interactions between children at various stages of language development and their caretakers. CHILDES provides vast amounts of useful data for linguistic, psychological, and sociological studies of child language development. To date, this database has served as the basis for over 1500 published articles and as a secondary resource in hundreds of other studies. Many of the CHILDES corpora are morphologically analyzed and annotated in a compatible manner, which makes it possible to compare language development across different languages. Recently, the English CHILDES database has been annotated with syntactic structures in the form of *grammatical relations* [52, 53, 54]; similar efforts are currently underway for several other languages. The CHILDES database thus provides a perfect environment for investigating language development and for evaluating psycholinguistic hypotheses.

Computational approaches to language learning from data, and in particular the works surveyed in Section 4, can be distinguished along the following axes:

The data. What data are presented to the learner? Cognitively-motivated approaches (Section 4.1) assume that the data are raw texts, usually of child-directed speech (but sometimes including also child speech, i.e., utterances produced earlier by the same child). Sometimes, these data are accompanied by some form of annotation, aiming to reflect the grounding of language in real-life situations. Grammar induction algorithms (Section 4.2) often do not assume that the data are linguistic; and when they are, they are often annotated with POS tags, and sometimes only the sequences of POS tags are considered, ignoring the actual words. Training material in this paradigm consists mainly of the WSJ, and in particular WSJ10.

The task. What is the learner required to learn? Is it a language as a set of strings, or are these strings augmented by (usually, tree-) structures?

The grammar. What formalism is the grammar expressed in? Grammar induction algorithms are usually formal and explicit in their definition of the class of models that they attempt to learn. These can be deterministic finite-state automata (FSA) or Hidden Markov Models (HMMs) or Probabilistic context-free grammars (PCFGs) or Tree substitution grammars, with a variety of probabilistic models. Cognitive models are more vague on this point, and may represent the grammar implicitly and informally, sometimes in a rather ad-hoc manner.

Evaluation. Grammar induction algorithms are evaluated on annotated data; fundamentally, they are expected to learn the bracketing (and, sometimes, also the labels) of manually annotated corpora. In contrast, works in the cognitive tradition evaluate on child language data, which is crucially not annotated. We elaborate on evaluation in the next section.

3 Evaluation

Several factors make the evaluation of language learning systems difficult. First, especially when child-data are concerned, the training data provided to the system are extremely limited: even with high-density corpora, it is assumed that the corpus reflects less than 10% of the utterances the child was exposed to during a very short period (see Rowland et al. [51]). Second, it is unclear whether the task should be evaluated by testing the strings generated by the grammar, or also the structures that the model induces on them. The latter task is more demanding, and it is usually unclear what the "correct" structures are that the grammar should produce. Clearly, the PTB is inappropriate to investigate child language; and WSJ10 in particular is a very artificial genre. Finally, while it is relatively easy to measure the portion of the target utterances that the system properly generated, it is much harder to asses the proportion of the utterances generated by the model that are indeed grammatical.

In the computational linguistics community, similar tasks are standardly evaluated using two measures adopted from Information Retrieval: *precision* and *recall*, and their harmonic mean, f*-score*. Informally, recall measures the ability of the grammar to account for new utterances: it is roughly the proportion of the strings in the test data that can be correctly generated by the grammar. Precision, on the other hand, measures the extent to which grammar-generated strings are observed in the test data. In the extreme case, one can always maximize either of the two measures: if a grammar generates nothing its precision is 100% (but its recall is 0); if it generates everything its recall is 100% (but its precision is very low). The f-score therefore balances between the two. For language learning tasks, however, while recall is relatively easy to asses, precision is much harder: for it is possible to present sentences from a test corpus to a learner and verify that the learner accepts them as grammatical; but it is more difficult to ask the learner to generate utterances, and then verify that they are correct.

van Zaanen and Geertzen [66] identify four types of approaches to the evaluation of learning algorithms, each with is own problems. The *looks good to me* approach (namely, informal evaluation by the author of a system) is obviously subjective and unreliable. An alternative is *rebuilding a-priory known grammars*; here; the authors of a system construct a small grammar whose sentences are used as input for the language learning system. This is again subjective, and in addition only toy grammars can be build in this fashion. The *language membership* method amounts to measuring the precision and recall of a learner with respect to a test corpus, which is problematic as explained above; finally, *comparison against a treebank* is particularly problematic for child language, and is additionally very brittle [66].

To address the problem of evaluation, Chang et al. [20] propose a measure, called *sentence prediction accuracy (SPA)*, which basically quantifies the extent to which a learner (be it a child or a computer program) can correctly order the words in a target utterance, when these words are unordered. While SPA overcomes some problems (e.g., it is independent of the language and of any underlying linguistic theory), it is a very inaccurate measure of grammaticality. It is extremely strict, in the sense that a mistake in the placement of a single word renders the entire utterance unaccounted for; and it also implies that if two grammatical utterances differ in their word order (e.g., because an adverbial is shifted in a sentence), only one of them will be counted as correct by

the SPA measure. Finally, the measure was only tested on extremely short utterances.[1] We conjecture that it will not scale up to longer sentences (indeed, the results of Chang et al. [20] indicate that the SPA measure correlates better with the specific corpus used for learning than with the learning algorithm).

An alternative method for estimating both precision and recall is proposed by Brodsky et al. [18]. Based on the observation that two constrained models that are trained on disjoint corpora are unlikely to agree on the grammaticality of any given sentence, it uses large corpora to train language models that are then used to assess the probability of the test sentences. A full evaluation of this method is still underway.

Finally, Kol et al. [36] recently proposed a first approximation for assessing the level of over-generation in a learner. They measure the precision of a learner by training it on a large corpus. Specifically, working with CHILDES data, in which files are ordered chronologically, they train the model on early files, plus 90% of the (child-directed *and* child) utterances in the current file; they then test on the remaining 10% of the child utterances in the current file. To assess over-generation, they repeat the same procedure, training on the same data but evaluating on child utterances (longer than one word) *in reverse word order*. Ideally, a learner should perform well on the first task and very poorly on the second.

4 Existing Approaches

4.1 Cognitively-Motivated Computational Approaches to Child Language Acquisition

Within the cognitive linguistics paradigm, computational approaches to language learning investigate the degree to which the utterances a child is exposed to can be used to determine the multi-word expressions the same child will produce during early language development. Lieven et al. [40] suggest that "a lexically-based positional analysis can account for the structure of a considerable proportion of children's early multi-word utterances." This is tested on eleven children aged between 1;0 and 3;0. On average, 60% of all the children's multi-word utterances are defined as frozen by the analysis. These results are replicated by Lieven et al. [38], this time focusing on one child, but using a high-density corpus consisting of 5 hours of recordings per week (together with a maternal diary for the previous 6 weeks.) The findings are that only one third of the multi-word utterances of the child are novel, and three quarters of those can be accounted for by one operation only on the basis of previous utterances. Five types of "operations" are defined which the child can use to construct a new utterance from fragments of previously-heard
utterances.

Dąbrowska and Lieven [26] identify two problems with the above method: first, "the method does not provide an explicit description of the child's linguistic knowledge." In

[1] While this piece of data is missing in Chang et al. [20], it can be deduced that the average length of an utterance in their corpus was less than 3. These figures are specified in Dąbrowska and Lieven [26], p. 446, presumably referring to the same corpus. There, the number of words per utterance ranges between 1.56 (Brian, age 2;0) and 3.15 (Annie, age 3;0). Even in the adult speech the average utterance length is 4.44).

other words, no explicit model of linguistic knowledge, or *grammar*, is defined. Second, "the method is too unconstrained since the five operations defined by the authors made it possible, in principle, to derive any utterance from any string." In other words, the above models are *over-generating*. To overcome the problems, they propose *two* operations: juxtaposition and superimposition. Working with a dense corpus of two children at ages 2;0 and 3;0, they divide the corpora into two parts: approximately the first 80% of the utterances in each corpus are defined as the *main* corpus, and the remainder are called *test*. The focus is syntactic questions; for each such utterance in the test corpus, called the *target*, they extract relevant *component units* from the main corpus. These are utterances that share lexical material with the target. They then determine whether the target can be produced from the extracted utterances by means of juxtaposition (i.e., concatenation) or superimposition. The latter operation is loosely defined; it amounts to identifying similarities among patterns in the main corpus, and generalizing such patterns to *schemas with slots*. Superimposition allows slots to be filled by lexical material. Crucially, the corpora used in this research were manually annotated with semantic information. Superimposition is then constrained by the semantic type of the slot, such that only fillers of the same type can be used. The results show that as much as 75% of the questions at age 2;0 are immediate imitations of previous questions (this figure goes down to as low as 21% at age 3;0); and all the rest can be generated with few (at most 4) operations from previously-heard material.

Lieven et al. [39] adapt the previous research, extending it to four children. The method, which is referred to as the *traceback procedure* here, is basically the same. Again, the data are assumed to be semantically annotated and the semantic tags are used in the definition of 'slots'. The actual algorithm is not defined in sufficient detail (for example, it is unclear how schemas with more than one slot are generated, or whether there is an upper bound to the number of slots in a schema).

Bannard et al. [7] augment the traceback procedure by a *trace forward* procedure. Here, the task of the learner is better defined: given a main corpus to learn from, the learner has to extract a formal grammar by generalizing the utterances in the main corpus. The grammar induction procedure is not described with sufficient rigor, but it is clear that the emerging grammar is a context-free grammar with a single non-terminal symbol. Rules are generated based on utterances that share lexical material, as above, but the details are not specified. Out of the infinitely many possible grammars that fit the data, Bannard et al. [7] select those that are most probable given the data, using a Bayesian model with simple independence assumptions for optimizing the likelihood, and minimum description length (MDL) assumptions of the prior. The results show that the extracted grammars perform well both in terms of their recall and, in lieu of a precision evaluation, in terms of their *perplexity* (defined informally as "how surprised the model is by the data").

In a series of works, Freudenthal et al. [28, 29, 30] develop the MOSAIC (Model of Syntax Acquisition in Children) paradigm. This model takes as input corpora of transcribed child-directed speech and learns to produce as output utterances that become progressively longer as learning proceeds. The model is based on a hierarchical network in which more deeply embedded nodes represent longer utterances, and where links connect nodes to form certain generalizations. Crucially, the same corpus is given to

the learner several times. While MOSAIC has been shown to properly simulate several phenomena associated with early language acquisition in several languages, in part due to its inherent bias towards learning from the edges of the utterance, it is not viewed as a realistic model of the language acquisition process itself, but rather as one possible implementation of inherent biases in learning.

4.2 Computational Grammar Induction

A different line of research falls under the category of *grammar induction* (or, more specifically, *computational grammar inference*). Here the goal is to devise algorithms that can learn accurate, compact models for identification of language (i.e., grammars) from finite sets of examples [4]. Such approaches are usually not cognitively motivated; Klein and Manning [32], p. 35, for example, explicitly mention that "the presented system makes no claims to modeling human language acquisition," and Borensztajn et al. [17] add that their approach "has no pretense of being a model for language acquisition"; but the relation to the works discussed above is obvious. Formally, a finite set of examples is consistent with infinitely many different grammars, and thus different approaches must somehow constrain or bias the set of hypotheses from which grammars can be drawn [27].

The EMILE model [1, 2, 3] attempts to learn the grammatical structure of a language from positive examples, without prior knowledge of the grammar. It is based on the idea that expressions of the same (syntactic) type can be substituted in the same context, and hence it searches for clusters of expressions and contexts in the input, interpreting them as grammatical types. The model then generalizes the sample and learns rules of a context-free grammar. A related approach is Alignment-based Learning (ABL) [63, 64, 62]. Given a corpus of sentences, an alignment learning phase first finds possible constituents by aligning pairs of sentences and identifying parallel strings. Strings that are unequal in a pair of sentences are considered *hypotheses*. Then, non-terminal types are assigned to hypotheses, merging different non-terminals that occur in the same context. The result is an induced context-free grammar. While ABL and EMILE are implemented differently, and EMILE only extracts a rule when sufficient support is available in the corpus whereas ABL stores all possible rule candidates and selects the best ones [65], the two systems are similar in spirit. In particular, both can learn recursive structures. The two systems were evaluated on Dutch corpora; the metric was unlabeled bracketing f-score. Whereas EMILE reached an f-score of 0.25-0.41 (depending on the corpus), ABL's performance was much higher, at 0.39-0.62 [65].

Stolcke and Omohundro [58] propose a technique called *Bayesian Model Merging (BMM)*: first, strings that are observed in the data are incorporated by adding ad-hoc rules to form an initial grammar; then, the grammar is made more concise by merging some of the rules. Stolcke and Omohundro [58] discuss two incarnations of their technique, one in which the models are probabilistic context-free grammars (PCFGs), and another in which they are hidden Markov models (HMMs). In the former, rules are merged by identifying non-terminal symbols A and B if the rule $A \rightarrow B$ is in the grammar; this leads to (over-) generalizations, and renders the grammar more compact. In the latter, two HMM states are merged to a state that inherits the union of their transitions (and emission probabilities). In both cases, the prior probabilities are optimized

by minimizing their description length. Stolcke and Omohundro [58] discuss the application of BMM to natural language learning, but do not provide quantitative evaluation results.

In a series of works, Klein and Manning [32, 33, 35] present the *constituent-context model (CCM)*. Here the task is to determine the correct bracketing of sentences in the input: the assumption is that the input is tagged with parts of speech (POS); in fact, the algorithm ignores the actual words and works on POS tag sequences. The output is a tree structure without the labels of non-terminal symbols. The model is a generative one; first, an initial bracketing is chosen from some distribution and a sentence is generated given the bracketing, assuming that the context and yield of each span are independent of each other. Then, an EM algorithm is run on the model to induce structure, assuming that the sentence is observed but the bracketing is not. This model was evaluated on the PTB WSJ10 subset, resulting in an f-score of 0.71, reducing about a quarter of the errors of a trivial (right-bracnhing trees) baseline (which yields 0.60). This result improves to 0.776 when the model is combined with a dependency parsing model in subsequent work [34].

Data-oriented parsing (DOP, Bod et al. [15]) is a paradigm for supervised parsing that differs from other approaches in that it considers *all* the possible structures given in a training corpus, and estimates their likelihood from the data. It can then be used to assign a structure to a new utterance by combining sub-trees from the training corpus. In its unsupervised version [11, 12, 13], called U-DOP, the algorithm initially assigns all possible unlabeled binary trees to an un-annotated training set, and then employs a probabilistic model to determine the most likely tree for a new utterance (various probabilistic models were investigated). The best results outperform the previous model, with an f-score of up to 0.80 on WSJ10.

Various works address the issue of inducing labels for the unlabeled trees. Notably, Borensztajn and Zuidema [16] extend the BMM model of Stolcke and Omohundro [58], but they assume that the input is already bracketed. Their algorithm then proceeds by merging nonterminal labels to maximize a Bayesian objective function. The algorithm is evaluated on the PTB WSJ10 subset, and shows best performance on the labeling task (although when used only for bracketing, it is much inferior to competing algorithms).

While many grammar induction algorithms start with strings of POS tags, this is not the case with Seginer [55], who uses lexical information (and does not assume known POS tags). While other algorithms resort to unsupervised learning of POS tags, which amounts to clustering, here the algorithm collects lists of labels for each word, based on its neighbors, and uses these labels to parse. The parser is incremental, local and greedy, and hence quite efficient. Evaluated on WSJ10, the results are an f-score of almost 0.76 when parsing begins from plain text.

Note that algorithms for inducing part of speech categories from raw data (i.e., unsupervised POS tagging) abound, both in the cognitive linguistic literature (e.g., Li et al. [37] and references therein) and in the computational linguistic literature (e.g., Banko and Moore [5], Smith and Eisner [56]).

4.3 Consolidating the Two Approaches

Most of the works described above fall into one of two classes: either the motivation is to explain child language acquisition from a cognitive perspective (Section 4.1); or it is to devise an efficient mechanism for inducing (formal) grammars from raw data (Section 4.2). Very few works, and only recently, try to consolidate some aspects of the two approaches.

The ADIOS system [57] implements a novel algorithm that learns a complex context-free grammar from raw data. Based on a graph representation, the algorithm performs segmentation and generalization of the input simultaneously. The system was applied to several types of data, both linguistic (including CHILDES data) and non-linguistic (protein sequences). Recall was evaluated automatically, while to assess precision human judgements were used. The results show that ADIOS is superior to other grammar induction algorithms that can learn from raw data. In a subsequent work, Berant et al. [9] observe that the algorithm does not deal well with complex texts and improve it by applying a two-stage learning technique: first, sentences in the input are split to sub-sentences on the basis of conjunctions in the text; then, the resulting simpler corpus is processed as above. Precision was evaluated by feeding the sentences generated by the learner to an alternative parser, and f-score varied between 0.24 and 0.39, depending on the precise task. Brodsky et al. [18] apply ADIOS to the full (English section of the) CHILDES corpus. Training on 300,000 utterances and testing on 500, the system reached a recall of 0.5 and precision of 0.63. Precision was again evaluated manually by humans judging the grammaticality of 100 generated utterances.

Borensztajn et al. [17] use the DOP paradigm as a vehicle for investigating psycholinguistic hypotheses. Specifically, they use the syntactically-anotated Brown corpus of CHILDES [54] to learn DOP-style structures. These tree fragments are then used to induce structure on utterances in the test corpus. This is an automatic approach to identifying the most probable multi-word units (constructions) in children's utterances. The main result is that *abstraction*, defined as the ratio between non-terminal and terminal leaves in the tree fragments that represent constructions, increases with age. One of the main drawbacks of this approach is that the grammar is induced from POS-tagged *and* syntactically-annotated corpora; cognitively, this amounts to assuming that children have access to the syntactic structure of the utterances they are exposed to, which cannot be the case in early language acquisition.

5 Directions for Future Research

According to Edelman and Waterfall [27], p. 265, "of the three goals of linguistic theory... the most promising one at present is, in fact, an algorithmic discovery procedure for grammar." Similarly, Adriaans and van Zaanen [4], p. 200 observe that "researchers within the several sub-fields [linguistic, empirical and formal grammatical inference] seem to have created certain boundaries between the fields" and conclude that "it is time to remove the (artificial) boundaries and combine the research performed within each sub-field." We propose that future research should indeed consolidate well-established findings of psycholinguistics with developments in computational linguistics to yield a

research program that is on one hand informed by our understanding of early language acquisition, and on the other hand is rigorously defined and robustly evaluated.

We list below some desiderata for the kind of computational models that we envision.

Data. Unlike much work in the area of grammar induction algorithms (Section 4.2), research concerned with child language acquisition must be trained (and evaluated) on dedicated corpora, of the kind exhibited by CHILDES. Ideally, they should be tested on more than one language. Models can be trained on both child and child-directed speech.

Task. We suggest that computational models of language acquisition focus on the easier task of learning language as a set of strings, leaving the induction of syntactic structures to future research.

Grammar. Cognitive works (Section 4.1) tend to be more vague on the formal properties of the class of languages admitted by the models they suggest. A good model must be explicit on this point. We believe that a reasonable language class for early language (up to, say, three years of age) is a proper subset of the regular languages. Models that can learn unrestricted context-free languages, for example, miss an important point and are likely to over-generate.

Model. Works that are specifically designed to model early child language acquisition should incorporate into the framework biases that reflect psycholinguistic models of acquisition processes [46, 42]. These include item-based learning, rote learning [45], left-edge biases [30], etc.

Evaluation. Clearly, evaluation is still an unsolved problem (Section 3), and much work is still needed in this area. Still, and in contrast to some of the approaches described in section 4.1, any computational model of language acquisition must be rigorously evaluated on real data.

To further emphasize this last point, Kol et al. [36] conducted an alternative evaluation of the *traeback* model [38, 26, 39, 6, 67]. They show that the original evaluation scheme in these works is lacking, as it focuses on recall but completely ignores precision. As a measure of over-generation, Kol et al. [36] apply the traceback method not just to child utterances in the test corpus, but also to the same utterances in reverse order. While the model can generate 64-64% of the genuine child utterances (showing reasonable recall), it can also generate 42-50% of the reverse utterances, indicating a serious over-generation problem.

One of the reasons for this problem is that the traceback model is not defined in a sufficiently rigorous way. The sets of operations allowed for traceback is not fixed, and changes from one work to another. Much of the work involved in applying the model to data is done manually in a way that prohibits computational re-implementation.

In contrast, works such as those discussed in Section 4.2 are often evaluated on WSJ10, a clearly inadequate corpus for assessing child language development. Some of them assume that the data are already annotated with parts of speech or even syntactic information, an obviously unacceptable assumption when child language is concerned. Language learning from sequences of POS-tags is a particularly bad example of how to model child language acquisition.

Clearly, then, the benefits of these different approaches must be consolidated in order for a formal, computational, linguistically- and cognitively-informed model of language

acquisition to emerge. Such a model must be rigorously defined, in a way that lends itself to computational implementation; formally, it should exhibit highly-retricted computational expressivity; it should employ biases that correspond to established observations of child language research (such as item-based learning, rote learning [45], left-edge biases [30], adherence to stages of acquisition [19, 10], etc.) Only future works that will correspond to such considerations will properly address the criticism of Bod [14], whereby "almost any current linguistic theory ... has given up on the construction of a precise, testable model of language use and language acquisition."

Acknowledgements

I am grateful to Sheli Kol, Alon Lavie, Brian MacWhinney and Bracha Nir for their continuous support. This research was supported in part by Grant No. 2007241 from the United States-Israel Binational Science Foundation (BSF) and by the National Science Foundation (NSF) under grant IIS-0414630.

References

[1] Adriaans, P.: Language Learning from a Categorial Perspective. PhD thesis, Universiteit van Amsterdam (1992)
[2] Adriaans, P.: Learning shallow context-free languages under simple distributions. In: Copestake, A., Vermeulen, K. (eds.) Algebras, Diagrams and Decisions in Language, Logic and Computation. CSLI/CUP, Stanford (2001)
[3] Adriaans, P., Vervoort, M.: The EMILE 4.1 grammar induction toolbox. In: Adriaans, P.W., Fernau, H., van Zaanen, M. (eds.) ICGI 2002. LNCS (LNAI), vol. 2484, pp. 293–295. Springer, Heidelberg (2002)
[4] Adriaans, P.W., van Zaanen, M.M.: Computational grammatical inference. In: Holmes, D.E., Jain, L.C. (eds.) Innovations in Machine Learning. Studies in Fuzziness and Soft Computing, vol. 194, ch. 7. Springer, Heidelberg (2006)
[5] Banko, M., Moore, R.C.: Part of speech tagging in context. In: COLING 2004: Proceedings of the 20th international conference on Computational Linguistics, Morristown, NJ, USA, p. 556. Association for Computational Linguistics (2004)
[6] Bannard, C., Lieven, E.: Repetition and reuse in child language learning. In: Corrigan, R., Moravcsik, E., Ouali, H., Wheatley, K. (eds.) Formulaic Language. John Benjamins, Amsterdam (2009)
[7] Bannard, C., Lieven, E., Tomasello, M.: Early grammatical development is piecemeal and lexically specific. Proceedings of the National Academy of Science 106(41), 17284–17289 (2009)
[8] Bates, E., MacWhinney, B.: Competition, variation, and language learning. In: [46], ch. 6, pp. 157–193 (1987)
[9] Berant, J., Gross, Y., Mussel, M., Sandbank, B., Edelman, S.: Boosting unsupervised grammar induction by splitting complex sentences on function words. In: Proceedings of the 31st Boston University Conference on Language Development, pp. 93–104. Cascadilla Press (2007)
[10] Berman, R.A.: Between emergence and mastery: The long developmental route of language acquisition. In: Berman, R.A. (ed.) Language development across childhood and adolescence. Trends in Language Acquisition Research, vol. 3, pp. 9–34. John Benjamins, Amsterdam/Philadelphia (2004)

[11] Bod, R.: An all-subtrees approach to unsupervised parsing. In: ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, Morristown, NJ, USA, pp. 865–872. Association for Computational Linguistics (2006a)
[12] Bod, R.: Unsupervised parsing with U-DOP. In: Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X), New York City, pp. 85–92. Association for Computational Linguistics (2006b)
[13] Bod, R.: Is the end of supervised parsing in sight? In: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, Prague, Czech Republic, pp. 400–407. Association for Computational Linguistics (2007)
[14] Bod, R.: Constructions at work or at rest? Cognitive Linguistics 20(1) (2009)
[15] Bod, R., Sima'an, K., Scha, R. (eds.): Data-Oriented Parsing. CSLI Publications, Stanford (2003)
[16] Borensztajn, G., Zuidema, W.: Bayesian model merging for unsupervised constituent labeling and grammar induction. ILLC Prepublication PP-2007-40, ILLC, University of Amsterdam (2007)
[17] Borensztajn, G., Zuidema, J., Bod, R.: Children's grammars grow more abstract with age — evidence from an automatic procedure for identifying the productive units of language. In: Proceedings of CogSci 2008 (2008)
[18] Brodsky, P., Waterfall, H., Edelman, S.: Characterizing motherese: On the computational structure of child-directed language. In: Proceedings of the 29th Cognitive Science Society Conference. Cognitive Science Society (2007)
[19] Brown, R.: A first language: the Early stages. Harvard University Press, Cambridge (1973)
[20] Chang, F., Lieven, E., Tomasello, M.: Automatic evaluation of syntactic learners in typologically-different languages. Cognitive Systems Research 9(3), 198–213 (2008)
[21] Chomsky, N.: Aspects of the theory of syntax. MIT Press, Cambridge (1965)
[22] Chomsky, N.: Language and Mind. Harcourt Brace Juvanovich, New York (1968)
[23] Chomsky, N.: Rules and representations. Behavioral and Brain Sciences 3, 1–61 (1980)
[24] Chomsky, N.: Reflections on Language. Pantheon, New York (1975)
[25] Church, K.W., Mercer, R.L.: Introduction to the special issue on computational linguistics using large corpora. Computational Linguistics 19(1), 1–24 (1993)
[26] Dąbrowska, E., Lieven, E.: Towards a lexically specific grammar of children's question constructions. Cognitive Linguistics 16(3), 437–474 (2005)
[27] Edelman, S., Waterfall, H.: Behavioral and computational aspects of language and its acquisition. Physics of Life Reviews 4(4), 253–277 (2007)
[28] Freudenthal, D., Pine, J.M., Gobet, F.: Modelling the development of children's use of optional infinitives in Dutch and English using MOSAIC. Cognitive Science 30, 277–310 (2006)
[29] Freudenthal, D., Pine, J.M., Gobet, F.: Understanding the developmental dynamics of subject omission: the role of processing limitations in learning. Journal of Child Language 34(01), 83–110 (2007)
[30] Freudenthal, D., Pine, J.M., Gobet, F.: Simulating the referential properties of Dutch, German, and English root infinitives in MOSAIC. Language Learning and Development 5, 1–29 (2009)
[31] Kennedy, G.: An introduction to corpus linguistics. Addison Wesley, Reading (1998)
[32] Klein, D., Manning, C.D.: Natural language grammar induction using a constituent-context model. In: Dietterich, T.G., Becker, S., Ghahramani, Z., Dietterich, T.G., Becker, S., Ghahramani, Z. (eds.) NIPS, pp. 35–42. MIT Press, Cambridge (2001)
[33] Klein, D., Manning, C.D.: A generative constituent-context model for improved grammar induction. In: ACL, pp. 128–135 (2002)

[34] Klein, D., Manning, C.D.: Corpus-based induction of syntactic structure: Models of dependency and constituency. In: ACL, pp. 478–485 (2004)
[35] Klein, D., Manning, C.D.: Natural language grammar induction with a generative constituent-context model. Pattern Recognition 38(9), 1407–1419 (2005)
[36] Kol, S., Nir, B., Wintner, S.: Acquisition of abstract slot-filler schemas: Computational evaluation. Presented at the COGSCI 2009 Workshop on Psychocomputational Models of Human Language Acquisition (2009)
[37] Li, P., Farkas, I., MacWhinney, B.: Early lexical development in a self-organizing neural network. Neural Networks 17(8-9), 1345–1362 (2004)
[38] Lieven, E., Behrens, H., Speares, J., Tomasello, M.: Early syntactic creativity: a usage-based approach. Journal of Child Language 30(2), 333–370 (2003)
[39] Lieven, E., Salomo, D., Tomasello, M.: Two-year-old children's production of multiword utterances: a usage-based analysis. Cognitive Linguistics 20(3), 481–507 (2009)
[40] Lieven, E.V., Pine, J.M., Baldwin, G.: Lexically-based learning and early grammatical development. Journal of Child Language 24(1), 187–219 (1997)
[41] MacWhinney, B.: The CHILDES Project: Tools for Analyzing Talk, 3rd edn. Lawrence Erlbaum Associates, Mahwah (2000)
[42] MacWhinney, B.: Models of the emergence of language. Annual Review of Psychology 49, 199–227 (1998)
[43] MacWhinney, B.: A multiple process solution to the logical problem of language acquisition. Journal of Child Language 31, 883–914 (2004a)
[44] MacWhinney, B.: A unified model of language acquisition. In: Kroll, J., De Groot, A. (eds.) Handbook of bilingualism: Psycholinguistic approaches. Oxford University Press, Oxford (2004b)
[45] MacWhinney, B.: Rules, rote, and analogy in morphological formations by Hungarian children. Journal of Child Language 2, 65–77 (1975)
[46] MacWhinney, B. (ed.): Mechanisms of language acquisition. Lawrence Erlbaum Associates, Hillsdale (1987)
[47] The emergence of language. In: MacWhinney, B. (ed.) Carnegie Mellon Symposia on Cognition. Lawrence Erlbaum Associates, Mahwah (1999)
[48] Marcus, M.P., Santorini, B., Marcinkiewicz, M.A.: Building a large annotated corpus of English: The Penn treebank. Computational Linguistics 19(2), 313–330 (1993)
[49] McEnery, A., Wilson, A.: Corpus Linguistics. Edinburgh University Press, Edinburgh (1996)
[50] Pinker, S.: The Language Instinct. William Morrow and Company, New York (1994)
[51] Rowland, C.F., Fletcher, S.L., Freudenthal, D.: Repetition and reuse in child language learning. In: Behrens, H. (ed.) Corpora in Language Acquisition Research: History, methods, perspectives, pp. 1–24. John Benjamins, Amsterdam (2008)
[52] Sagae, K., MacWhinney, B., Lavie, A.: Automatic parsing of parent-child interactions. Behavior Research Methods, Instruments, and Computers 36, 113–126 (2004)
[53] Sagae, K., Davis, E., Lavie, A., MacWhinney, B., Wintner, S.: High-accuracy annotation and parsing of CHILDES transcripts. I. In: Proceedings of the ACL-2007 Workshop on Cognitive Aspects of Computational Language Acquisition, Prague, Czech Republic, pp. 25–32. Association for Computational Linguistics (2007)
[54] Sagae, K., Davis, E., Lavie, A., MacWhinney, B., Wintner, S.: Morphosyntactic annotation of CHILDES transcripts. Journal of Child Language (to appear)
[55] Seginer, Y.: Fast unsupervised incremental parsing. In: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, Prague, Czech Republic, pp. 384–391. Association for Computational Linguistics (2007)

[56] Smith, N.A., Eisner, J.: Annealing techniques for unsupervised statistical language learning. In: ACL 2004: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, Morristown, NJ, USA, p. 486. Association for Computational Linguistics (2004)
[57] Solan, Z., Horn, D., Ruppin, E., Edelman, S.: Unsupervised learning of natural languages. Proceedings of the National Academy of Sciences of the United States of America 102(33), 11629–11634 (2005)
[58] Stolcke, A., Omohundro, S.M.: Inducing probabilistic grammars by bayesian model merging. In: Carrasco, R.C., Oncina, J. (eds.) ICGI 1994. LNCS, vol. 862, pp. 106–118. Springer, Heidelberg (1994)
[59] Tomasello, M.: On the different origins of symbols and grammars. In: Christiansen, M.H., Kirby, S. (eds.) Language Evolution. Studies in the Evolution of Language, ch. 6, pp. 94–110. Oxford University Press, Oxford (2003)
[60] Tomasello, M.: Acquiring linguistic constructions. In: Kuhn, D., Siegler, R. (eds.) Handbook of Child Psychology, pp. 255–298. Wiley, New York (2006)
[61] Tomasello, M.: Language is not an instinct. Cognitive Development 10, 131–156 (1995)
[62] van Zaanen, M.: Implementing alignment-based learning. In: Adriaans, P.W., Fernau, H., van Zaanen, M. (eds.) ICGI 2002. LNCS (LNAI), vol. 2484, pp. 312–314. Springer, Heidelberg (2002)
[63] van Zaanen, M.: ABL: alignment-based learning. In: Proceedings of the 18th conference on Computational linguistics, Morristown, NJ, USA, pp. 961–967. Association for Computational Linguistics (2000)
[64] van Zaanen, M.: Bootstrapping Structure into Language: Alignment-Based Learning. PhD thesis, University of Leeds, Leeds, UK (2002a)
[65] van Zaanen, M., Adriaans, P.: Alignment-Based Learning versus EMILE: A comparison. In: Proceedings of the Belgian-Dutch Conference on Artificial Intelligence (BNAIC), Amsterdam, The Netherlands, pp. 315–322 (2001)
[66] van Zaanen, M., Geertzen, J.: Problems with evaluation of unsupervised empirical grammatical inference systems. In: Clark, A., Coste, F., Miclet, L. (eds.) ICGI 2008. LNCS (LNAI), vol. 5278, pp. 301–303. Springer, Heidelberg (2008)
[67] Vogt, P., Lieven, E.: Verifying theories of language acquisition using computer models of language evolution. In: Adaptive Behavior Special issue on Language Evolution: Computer models for Empirical Data (forthcoming)

ETL Ensembles for Chunking, NER and SRL

Cícero N. dos Santos[1], Ruy L. Milidiú[2],
Carlos E. M. Crestana[2], and Eraldo R. Fernandes[2,3]

[1] Mestrado em Informática Aplicada – MIA
Universidade de Fortaleza – UNIFOR
Fortaleza, Brazil
cnogueira@unifor.br
[2] Departamento de Informática
Pontifícia Universidade Católica do Rio de Janeiro – PUC-Rio
Rio de Janeiro, Brazil
milidiu@inf.puc-rio.br
[3] Laboratório de Automação
Instituto Federal de Educação, Ciência e Tecnologia de Goiás – IFG
Jataí, Brazil
{ccrestana,efernandes}@inf.puc-rio.br

Abstract. We present a new ensemble method that uses Entropy Guided Transformation Learning (ETL) as the base learner. The proposed approach, ETL Committee, combines the main ideas of Bagging and Random Subspaces. We also propose a strategy to include redundancy in transformation-based models. To evaluate the effectiveness of the ensemble method, we apply it to three Natural Language Processing tasks: Text Chunking, Named Entity Recognition and Semantic Role Labeling. Our experimental findings indicate that ETL Committee significantly outperforms single ETL models, achieving state-of-the-art competitive results. Some positive characteristics of the proposed ensemble strategy are worth to mention. First, it improves the ETL effectiveness without any additional human effort. Second, it is particularly useful when dealing with very complex tasks that use large feature sets. And finally, the resulting training and classification processes are very easy to parallelize.

Keywords: entropy guided transformation learning, ensemble methods, text chunking, named entity recognition, semantic role labeling.

1 Introduction

Ensemble methods are learning algorithms that generate multiple individual classifiers and combine them to classify new samples. Usually, the final classification is done by taking a weighted or majority vote of the individual predictions. Such model combinations are known as *ensemble models* or *committees*. The main purpose of model combination is to reduce the generalization error of a classifier. Ensemble algorithms have received considerable attention in the last years [1,2].

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 100–112, 2010.

Transformation Based Learning (TBL) is a machine learning algorithm introduced by Brill [3]. TBL is a corpus-based error-driven approach that learns a set of ordered transformation rules which correct mistakes of a baseline classifier. It has been successfully used for several important NLP tasks. Nevertheless, it suffers from a serious drawback: the need of costly human expertise to build the required TBL rule templates. This is a bottleneck for wide spreading its application. Entropy Guided Transformation Learning (ETL) [4] eliminates the TBL bottleneck by providing an automatic mechanism to construct good rule templates. Hence, ETL allows the construction of ensemble models that use Transformation Learning.

In this work, we present an ensemble method that uses ETL as the base learner. The proposed approach, ETL Committee, combines the main ideas of Bagging [5] and Random Subspaces [6]. In order to evaluate the effectiveness of the ensemble method, we apply it to three Natural Language Processing tasks: Text Chunking (TCK), Named Entity Recognition (NER) and Semantic Role Labeling (SRL). Our experimental findings indicate that ETL Committee significantly outperforms single ETL models, achieving state-of-the-art competitive results for the three tasks. As far as we know, this is the first study that uses transformation rule learning as the base learner for an ensemble method.

The remainder of the paper is organized as follows. In section 2, we briefly describe the ETL strategy. In section 3, we detail the ETL Committee approach. In section 4, the experimental design and the corresponding results are reported. Finally, in section 5, we present our concluding remarks.

2 Entropy Guided Transformation Learning

Entropy Guided Transformation Learning [4] generalizes Transformation Based Learning by automatically generating rule templates. ETL employs an *entropy guided template generation* approach, which uses Information Gain (IG) in order to select the feature combinations that provide good template sets [7]. ETL has been successfully applied to part-of-speech (POS) tagging [8], phrase chunking [4], named entity recognition [7], clause identification [9] and dependency parsing [10], producing results at least as good as the ones of TBL with handcrafted templates. A detailed description of ETL can be found in [7]. In the next two subsections, we present two variations on the basic strategy. These variations are very useful when using ETL as a base learner for an ensemble method.

2.1 Template Sampling

There are cases where learning the largest rule set is necessary. For instance, when training an ensemble of classifiers using different training data sets, overfitting can be beneficial. This is because, in this specific case, overfitting can introduce diversity among the ensemble members. As an example, some DT ensemble learning methods do not use pruning [11,12,6].

However, the larger the rule set the longer it takes to be learned. Therefore, in our ETL implementation, we also include the *template sampling* functionality,

which consists in training the ETL model using only a randomly chosen fraction of the generated templates. Besides being simple, this strategy provides a speed up control that is very useful when multiple ETL models are to be learned.

2.2 Redundant Transformation Rules

As previously noticed by Florian [13], the TBL learning strategy shows a total lack of redundancy in modeling the training data. Only the rule that has the largest score is selected at each learning iteration. All alternative rules that may correct the same errors, or a subset of the errors, are ignored. This greedy behavior is not a problem when the feature values tested in the alternative rules and the ones tested in the selected rule always co-occur. Unfortunately, this is not always the case when dealing with sparse data.

Florian includes redundancy in his TBL implementation by adding to the list of rules, after the training phase has completed, all the rules that do not introduce error. Florian shows that these additional rules improve the TBL performance for tasks were a word classification is independent of the surrounding word classifications.

In our ETL implementation, we also include redundancy in the TBL step, but in a different way. At each iteration, when the best rule b is learned, the algorithm also learns all the rules that do not include errors and correct exactly the same examples corrected by b. These redundant rules do not alter the error-driven learning strategy, since they do not provide any change in the training data. This kind of redundancy is more effective for low scored rules, since they are more likely to use sparse feature values and their selection is supported by just a few examples.

Redundant rules increase the model overfitting since more information from the training set is included in the learned model. Therefore, redundant rules does not improve the performance of single ETL classifiers. However, the inclusion of redundancy improves the classification quality when several classifiers are combined, since overfitting can be beneficial to generate more diverse classifiers in an ensemble strategy.

3 ETL Committee

According to Dietterich [14], a necessary and sufficient condition for an ensemble of classifiers to have a lower generalization error than any of its individual members is that the classifiers are accurate and diverse. A classifier is considered to be accurate if its error rate on new data is lower than just guessing. Two classifiers are diverse if they make different errors on new data.

In this section, we present ETL Committee, an ensemble method that uses ETL as a base learner. The ETL Committee strategy relies on the use of training data manipulation to create an ensemble of ETL classifiers. ETL Committee combines the main ideas of Bagging [5] and Random Subspaces [6]. From Bagging, we borrow the bootstrap sampling method. From Random Subspaces, we

use the feature sampling idea. In the ETL Committee training, we use ETL with template sampling, which provides an additional randomization step.

3.1 ETL Committee Training Phase

Given a labeled training set $\mathcal{T}$, the ETL Committee algorithm generates L ETL classifiers using different versions of $\mathcal{T}$. In Figure 1, we detail the ETL Committee training phase. The creation of each classifier is independent from the others. Therefore, the committee training process can be easily parallelized. In the creation of a classifier c, the first step consists in using *bootstrap sampling* to produce a bootstrap replicate $\mathcal{T}'$ of the training set $\mathcal{T}$. Next, *feature sampling* is applied to $\mathcal{T}'$, generating the training set $\mathcal{T}''$. Finally, in the *ETL training* step, a rule set is learned using $\mathcal{T}''$ as a training set. In Section 4.5, we show some experimental results that highlight the contribution of each one of these steps to the committee behavior. These steps are detailed in the following subsections.

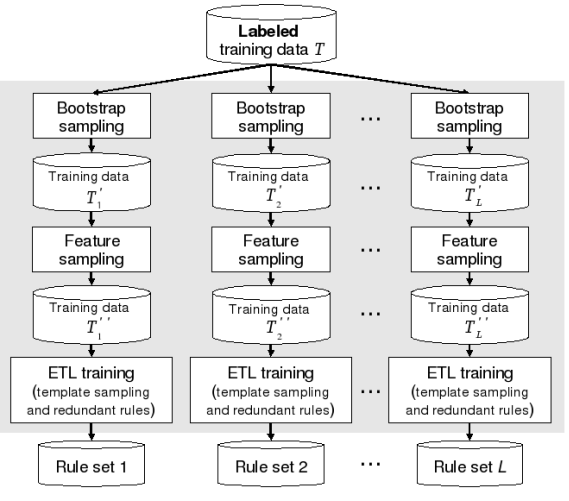


Fig. 1. ETL Committee training phase

Bootstrap sampling. In the *bootstrap sampling* step, a new version of the training set is generated using bootstrapping. *Bootstrapping* consists of sampling at random with replacement from the training set to generate an artificial training set of the same size as the original one. Hence, given a training set $\mathcal{T}$ consisting of n examples, a bootstrap replicate $\mathcal{T}'$ is constructed by sampling n examples at random, with replacement, from $\mathcal{T}$. Bootstrapping is the central idea of Bagging, where it is used to provide diversity among the ensemble members.

According to Breiman [5], an ensemble of classifiers trained on different bootstrap replicates can be effective if the base learner is *unstable*. An unstable classifier is the one where small changes in the training set result in large changes in its predictions. Due to the greedy nature of the TBL learning process, rule

selection is very sensitive to the occurrence of just a few examples. Usually, the rules in the tail of the learned rule set are selected based on just one or two error corrections. Therefore, we believe that small changes in the training set are able to significantly change the learned rule set. Moreover, since ETL uses DT to obtain templates and DT is an unstable learner [5], there are variability between the template sets generated from different bootstrap replicates. The use of different template sets has the potential to increase the ensemble diversity. The number of bootstrap replicates is called the *ensemble size*.

Feature sampling. In this step, a new version of the training set is generated by randomly selecting a subset of the available features. The manipulation of the input feature set is a general technique for generating multiple classifiers. As each classifier is generated using a randomly drawn feature subset, the diversity among the ensemble members tends to increase. Feature sampling is the main idea used in the Random Subspaces ensemble method. This strategy is particularly useful when a large set of features is available. The percentage of input features to be included in the subset is a parameter of ETL Committee.

ETL training. In the *ETL training* step, a set of transformation rules is learned using the training set resulted from the two previous steps. Here, *template sampling* and *redundant transformation rules* are used. We use template sampling for two reasons: (1) it provides more diversity among the ensemble members, since it increases the chance of each classifier to be trained with a very different template set; (2) it speeds up the training process, since less templates are used, enabling the learning of larger rule sets in a reasonable time. Note that by sampling templates we are sampling feature combinations. Hence, the template sampling can be seen as a kind of feature sampling at the base learner level. The number of templates to be sampled is also a parameter of ETL Committee.

We use redundant rules since it increases the overfitting, and more information from the training set is included in the learned model. Overfitting is another way to introduce diversity among the ensemble members [6,12,11].

3.2 ETL Committee Classification Phase

When classifying new data, each transformation rule set is independently applied to the input data. For each data point, each ETL model gives a classification, and we say the model "votes" for that class. The final data point classification is computed by majority voting. A drawback of ETL Committee, as well as the other ensemble methods, is that it increases the classification time. However, this process can be easily parallelized, since the application of each rule set is independent from the others.

3.3 Related Work

Breiman [12] presents an ensemble model called *Random Forest*, which uses bootstrapping and feature sampling. In the Random Forest learning process,

first, bootstrap sampling is employed to generate multiple replicates of the training set. Then, a decision tree is grown for each training set replicate. When growing a tree, a subset of the available features is randomly selected at each node, the best split available within those features is selected for that node. Each tree is grown to the largest extent possible, and there is no pruning. Random Forest is specific for decision trees, since the feature sampling step occurs at the base learner level. ETL Committee differs from Random Forest in three main aspects: the base learner, where ETL is used; the feature sampling, which is done outside of the base learner; and the template sampling, which is a feature combination sampling method employed at the base learner level.

Panov & Dzeroski [1] describe an ensemble method that also combines Bagging and Random Subspaces. Their intention is to achieve an algorithm whose behavior is similar to the one of Random Forests, but with the advantage of being applicable to any base learner. Their method uses bootstrap sampling followed by feature sampling to generate different training set. They show that, when using DT as a base learner, their approach has a comparable performance to that of random forests. The ETL Committee method is similar to the one of Panov & Dzeroski in terms of training set manipulation. On the other hand, ETL Committee differs from the Panov & Dzeroski approach because it includes template sampling, which is a randomization at the base learner level.

4 Experiments

This section presents the experimental setup and results of the application of ETL Committee to three tasks: Text Chunking (TCK), Named Entity Recognition (NER) and Semantic Role Labeling (SRL). ETL Committee results are compared with the results of ETL and the state-of-the-art system for each corpus.

4.1 Machine Learning Modeling

The three tasks are modeled as token classification problems. Which means that, given a text, the learned system must predict a class label for each token.

We use the following ETL and ETL Committee common parameter setting in our experiments with the three tasks. The parameters are empirically tuned using the training and development sets available for the NER and SRL tasks.

ETL: we use a context window of size seven. We use templates which combine at most six features. Therefore, when extracting templates from DTs, the extraction process examines only the six first DT levels. We let the ETL algorithm learn rules whose score is at least two.

ETL$_{CMT}$: for the ETL Committee, in the *bootstrap sampling* step, we use sentences as sampling units for bootstrapping. We set the ensemble size to 100. In the *feature sampling* step, we randomly sample 90% of the features for each classifier. In the *ETL training* step, we let the ETL algorithm to learn the largest rule set possible. We use 50 as the default number of templates to be sampled in the creation of each classifier. However, we use 100 templates for the SRL task. This

is because SRL involves a large number of features, which produces a larger number of templates.

BLS: For the TCK task, the initial classifier, or baseline system (BLS), assigns to each word the *chunk tag* that was most frequently associated with the part-of-speech of that word in the training set. For the NER task, the BLS assigns to each word the *named entity tag* that was most frequently associated with that word in the training set. If capitalized, an unknown word is tagged as a person, otherwise it is tagged as non entity. *Unknown words*, are the words that do not appear in the training set. For the SRL task, we use the same BLS proposed for the CoNLL-2004 shared task [15], which is based on six heuristic rules that make use of POS and phrase chunks.

4.2 Text Chunking

Text chunking consists in dividing a text into syntactically correlated parts of words [16]. It provides a key feature that helps on more elaborated NLP tasks such as NER and SRL.

The data used in the Text Chunking experiments is the CoNLL-2000 corpus, which is described in [16]. This corpus contains sections 15-18 and section 20 of the Penn Treebank, and is pre-divided into 8936-sentence training set and a 2012-sentence test set. This corpus is tagged with both POS and chunk tags. The *chunk tags* feature provides the phrase chunking annotation. We use the `IOB2` tagging style, where: `O`, means that the word is not a phrase; `B-X`, means that the word is the first one of a phrase type `X` and `I-X`, means that the word is inside of a phrase type `X`.

In [17], the authors present an SVM-based system with state-of-the-art performance for the CoNLL-2000 Corpus. Therefore, for this Corpus, we also list the SVM system performance reported by Wu et al.

In Table 1, we summarize the system performance results. The ETL system reduces the BLS $F_{\beta=1}$ error by 66%, from 22.93 to 7.72. The ETL_{CMT} system significantly reduces the $F_{\beta=1}$ error by 13% when compared to the single ETL. The ETL_{CMT} performance is competitive with the one of the SVM system.

4.3 Named Entity Recognition

Named Entity Recognition (NER) is the problem of finding all proper nouns in a text and to classify them among several given categories of interest. Usually, there are three given categories: Person, Organization and Location.

For the NER experiment, we use the Spanish CoNLL-2002 Corpus [18]. This corpus is annotated with four named entity categories: Person, Organization, Location and Miscellaneous. This corpus is pre-divided into training and test sets. It also includes a development set which have characteristics similar to the test corpora. This corpus is annotated with *POS* and *named entity (NE) tags*. We use the `IOB1` tagging style, where: `O`, means that the word is not a NE; `I-X`, means that the word is part of a NE type `X` and `B-X` is used for the leftmost word of a NE beginning immediately after another NE of the same type.

Table 1. System performances for the CoNLL-2000 Corpus

System	Precision (%)	Recall (%)	$F_{\beta=1}$
SVM	**94.12**	**94.13**	**94.12**
ETL_{CMT}	93.11	93.42	93.27
ETL	92.24	92.32	92.28
BLS	72.58	82.14	77.07

We generate three derived features: *Capitalization Information*, which classify the words according to their capitalization: First Uppercase, All Uppercase, Lowercase, Number or Punc.; *Dictionary Membership*, which assumes one of the following categorical values: Upper, Lower, Both or None; and *Word Length*, which classify the words according to their lengths: 1, 2, 3, 4, 5, 6-8 or $>$8.

Our named entity recognition approach follows the two stages strategy proposed in [3] for POS tagging. The first stage, the *morphological*, classifies the unknown words using morphological information. The second stage, the *contextual*, classifies the known and unknown words using contextual information. We use ETL and ETL Committee for the contextual stage only, since the morphological stage uses trivial templates.

In [19], the authors present an AdaBoost system with state-of-the-art performance for the Spanish CoNLL-2002 Corpus. Their AdaBoost system uses decision trees as a base learner. Therefore, for this Corpus, we also list the AdaBoost system performance reported by Carreras et al.

Table 2. System performances for the Spanish CoNLL-2002 Corpus

System	Precision (%)	Recall (%)	$F_{\beta=1}$
AdaBoost	**79.27**	**79.29**	**79.28**
ETL_{CMT}	76.99	77.94	77.46
ETL	75.50	77.07	76.28
BLS	49.59	63.02	55.51

In Table 2, we summarize the system performance results for the test set. The ETL system reduces the BLS $F_{\beta=1}$ error by 47%, from 44.49 to 23.72. The ETL_{CMT} system reduces the $F_{\beta=1}$ error by 5% when compared to the single ETL system. The ETL_{CMT} performance is very competitive with the one of the AdaBoost system. Moreover, for the Spanish CoNLL-2002, the ETL Committee system is in top three when compared with the 12 CoNLL-2002 contestant systems.

4.4 Semantic Role Labeling

Semantic Role Labeling (SRL) is the process of detecting basic event structures such as *who* did *what* to *whom*, *when* and *where* [20]. More specifically, for each

predicate of a clause, whose head is typically a verb, all the constituents in the sentence which fill a semantic role of the verb have to be recognized. A verb and its set of semantic roles (arguments) form a *proposition* in the sentence. SRL provides a key knowledge that helps to build more elaborated document management and information extraction applications.

Since our purpose is to examine the ETL Committee performance for a complex task, we do not use the full parsing information in our SRL experiments. Therefore, we evaluate the performance of ETL Committee over the CoNLL-2004 Corpus [15]. This corpus was used in the CoNLL-2004 shared task, which consisted in resolving SRL without full parsing. It is a subset of the Proposition Bank (PropBank), an approximately one-million-word corpus annotated with predicate-argument structures. The PropBank annotates the Wall Street Journal part of the Penn TreeBank with verb argument structure. The CoNLL-2004 Corpus uses Penn TreeBank sections 15-18 for training and section 21 for test. Section 20 is used as a development set.

The CoNLL-2004 Corpus is annotated with four basic input features: *POS tags*, *phrase chunks*, *clauses* and *named entities*. The Corpus also includes two other features: the *target verbs* feature, which indicates the verbs whose arguments must be labeled; and *srl tags*, which provides the semantic labeling. The *srl tags* used in the PropBank annotation numbers the arguments of each predicate from `A0` to `A5`. Adjunctive arguments are referred to as `AM-T`, where `T` is the type of the adjunct. Argument references share the same label with the actual argument prefixed with `R-`. References are typically pronominal.

Using the input features, we produce the following thirteen derived features. *Token Position*: indicates if the token comes before or after the target verb. *Temporal*: indicates if the word is or not a temporal keyword. *Path*: the sequence of chunk tags between the chunk and the target verb. *Pathlex*: the same as the *path* feature with the exception that here we use the preposition itself instead of the `PP` chunk tag. *Distance*: the number of chunks between the chunk and the target verb. *VP Distance*: distance, in number of `VP` chunks, between the token and the verb. *Clause Path*: the clause bracket chain between the token and the target verb. *Clause Position*: indicates if the token is inside or outside of the clause which contains the target verb. *Number of Predicates*: number of target verbs in the sentence. *Voice*: indicates the target verb voice. *Target Verb POS*: POS tag of the target verb. *Predicate POS Context*: the POS tags of the words that immediately precede and follow the predicate. *Predicate Argument Patterns*: for each predicate, we identify the most frequent left and right patterns of the core arguments (`A0` through `A5`) in the training set. All these features were previously used in other SRL systems [21].

SRL Preprocessing. Our system classifies chunks instead of words. Therefore, here, a token represents a complete text chunk. In the preprocessing step, the original word-based tokens are collapsed in order to generate the new representation. In the collapsing process, only the feature values of the phrase chunk headwords are retained. The chunk headword is defined as its rightmost word. This preprocessing speeds up the training step, since the number of tokens to

be annotated are reduced. Moreover, larger sentence segments are covered with smaller context window sizes.

We treat propositions independently. Therefore, for each target verb we generate a separate sequence of tokens to be annotated. In general, all the arguments of a proposition are inside the target verb clause. Hence, we do not include tokens that are outside of the target verb clause. The only exception is when we have a nested clause that begins with a target verb. Here, we must also include the external clause.

SRL Results. Hacioglu et al. [21] present a SVM system with state-of-the-art performance for the CoNLL-2004 Corpus. Therefore, we also list the SVM system performance reported by Hacioglu et al.

In Table 3, we summarize the system performance results for the test set. The ETL system reduces the BLS $F_{\beta=1}$ error by 40%, from 60.55 to 36.63. The ETL_{CMT} system reduces the $F_{\beta=1}$ error by 11% when compared to the single ETL system. The ETL_{CMT} performance is very competitive with the SVM system. Nevertheless, the ETL Committee system is in top two when compared with the 10 CoNLL-2004 contestant systems. Moreover, the precision of the ETL_{CMT} system is better than the one of the SVM system, and a reasonable recall is maintained. We obtain similar results in the development set.

Table 3. System performances for the CoNLL-2004 Corpus

System	Precision (%)	Recall (%)	$F_{\beta=1}$
SVM	72.43	**66.77**	**69.49**
ETL_{CMT}	**76.44**	60.25	67.39
ETL	70.60	57.48	63.37
BLS	55.57	30.58	39.45

4.5 ETL Committee Behavior

In this section, we present some results on the behavior of the ETL Committee learning strategy. Our intention is three-fold: to analyze the importance of redundant rules; to investigate how the ensemble performance behaves as the ensemble size increases and; to analyze the ETL Committee performance sensitivity to the percentage of sampled features. We use the SRL CoNLL-2004 development set to assess the system performances.

Figure 2 demonstrates the relationship of the $F_{\beta=1}$ for a given number of ETL classifiers in the ensemble. We can see that the ensemble performance increases rapidly until approximately 40 classifiers are included. Then, the $F_{\beta=1}$ increases slowly until it gets stable with around 100 classifiers. Note that using just 50 models we have a $F_{\beta=1}$ of 68.7. ETL Committee has a similar behavior in the other two tasks: TCK and NER.

In Table 4, we show the ETL Committee performance for different values of the feature sampling parameter. For this experiment, we create ensembles of 50

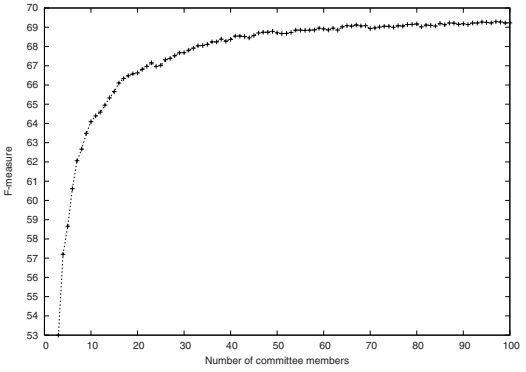


Fig. 2. $F_{\beta=1}$ x Number of committee members curve

classifiers. The best performance occurs when 70% of the features are randomly sampled for each classifier. In this case, the $F_{\beta=1}$ increases by about 0.7 when compared to the result in the first table line, where all features are used. In Table 4, we can see that even using only 50% of the features, the performance does not degrade. However, using less than 70% of the features can lead to poor results for tasks with a few number of features such as TCK.

Table 4. ETL Committee performance sensitivity to the percentage of sampled features

Percentage of sampled features	Precision (%)	Recall (%)	$F_{\beta=1}$
100%	75.43	61.95	68.03
90%	75.97	62.21	68.40
70%	76.44	**62.40**	**68.71**
50%	**76.64**	61.50	68.24

In Table 5, we show the ETL Committee performance when redundant rules are used or not used. For this experiment, we also create ensembles of 50 classifiers. The result in the first table line corresponds to the default ETL Committee method, which uses redundant rules. The second table line presents the ensemble performance when redundant rules are not used. In this case, the $F_{\beta=1}$ drops by about two points. This indicates that the overfitting provided by redundant rules is very important to the construction of more diverse ETL classifiers.

Table 5. Importance of redundant rules for the ETL Committee performance

Redundant rules	Precision (%)	Recall (%)	$F_{\beta=1}$
YES	76.44	62.40	68.71
NO	76.63	59.10	66.73

5 Conclusions

Entropy Guided Transformation Learning is a machine learning algorithm that generalizes TBL. In this work, we present ETL Committee, a new ensemble method that uses ETL as the base learner. It combines the main ideas of Bagging and Random Subspaces. We also propose a strategy to include redundancy in transformation-based models. To evaluate the effectiveness of the ensemble method, we apply it to three NLP tasks: TCK, NER and SRL.

Our experimental results indicate that ETL Committee significantly outperforms single ETL models. We also find out that redundant rules have a significant impact in the ensemble result. This finding indicates that the overfitting provided by redundant rules helps the construction of more diverse ETL classifiers. Some positive characteristics of the proposed ensemble strategy are worth to mention. First, it improves the ETL effectiveness without any additional human effort. Second, it is particularly useful when dealing with very complex tasks that use large feature sets. This is the case of the SRL task, where ETL Committee provides a significant $F_{\beta=1}$ improvement. And finally, the resulting training and classification processes are very easy to parallelize, since each classifier is independent from the others. The main drawback of ETL Committee is the increasing of the classification time. A possible way to overcome this issue is to convert transformation rules into deterministic finite-state transducers, as proposed by [22].

References

1. Panov, P., Džeroski, S.: Combining bagging and random subspaces to create better ensembles. In: Berthold, M.R., Shawe-Taylor, J., Lavrač, N. (eds.) IDA 2007. LNCS, vol. 4723, pp. 118–129. Springer, Heidelberg (2007)
2. García-Pedrajas, N., Ortiz-Boyer, D.: Boosting random subspace method. Neural Networks 21(9), 1344–1362 (2008)
3. Brill, E.: Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. Comp. Linguistics 21(4), 543–565 (1995)
4. Milidiú, R.L., dos Santos, C.N., Duarte, J.C.: Phrase chunking using entropy guided transformation learning. In: Proceedings of ACL 2008, Columbus, Ohio (2008)
5. Breiman, L.: Bagging predictors. Machine Learning 24(2), 123–140 (1996)
6. Ho, T.K.: The random subspace method for constructing decision forests. IEEE Trans. Pattern Anal. Mach. Intell. 20(8), 832–844 (1998)
7. dos Santos, C.N., Milidiú, R.L.: Entropy Guided Transformation Learning. In: Foundations of Computational Intelligence. Learning and Approximation, vol. 1. Studies in Computational Intelligence, vol. 201, pp. 159–184. Springer, Heidelberg (2009)
8. dos Santos, C.N., Milidiú, R.L., Rentería, R.P.: Portuguese part-of-speech tagging using entropy guided transformation learning. In: Teixeira, A., de Lima, V.L.S., de Oliveira, L.C., Quaresma, P. (eds.) PROPOR 2008. LNCS (LNAI), vol. 5190, pp. 143–152. Springer, Heidelberg (2008)
9. Fernandes, E.R., Pires, B.A., dos Santos, C.N., Milidiú, R.L.: Clause identification using entropy guided transformation learning. In: Proceedings of STIL 2009 (2009)
10. Milidiú, R.L., Crestana, C.E.M., dos Santos, C.N.: A token classification approach to dependency parsing. In: Proceedings of STIL 2009 (2009)

11. Banfield, R.E., Hall, L.O., Bowyer, K.W., Kegelmeyer, W.P.: Ensemble diversity measures and their application to thinning. Information Fusion 6(1), 49–62 (2005)
12. Breiman, L.: Random forests. Machine Learning 45(1), 5–32 (2001)
13. Florian, R.: Transformation Based Learning and Data-Driven Lexical Disambiguation: Syntactic and Semantic Ambiguity Resolution. PhD thesis, The Johns Hopkins University (2002)
14. Dietterich, T.G.: Ensemble methods in machine learning. In: Kittler, J., Roli, F. (eds.) MCS 2000. LNCS, vol. 1857, pp. 1–15. Springer, Heidelberg (2000)
15. Carreras, X., Màrquez, L.: Introduction to the conll-2004 shared task: Semantic role labeling. In: Ng, H.T., Riloff, E. (eds.) HLT-NAACL, Workshop CoNLL 2004, Boston, USA, May 2004, pp. 89–97. ACL (2004)
16. Sang, E.F.T.K., Buchholz, S.: Introduction to the conll-2000 shared task: chunking. In: Proceedings of the 2nd workshop on Learning language in logic and the 4th CONLL, Morristown, NJ, USA, pp. 127–132. ACL (2000)
17. Wu, Y.C., Chang, C.H., Lee, Y.S.: A general and multi-lingual phrase chunking model based on masking method. In: Gelbukh, A. (ed.) CICLing 2006. LNCS, vol. 3878, pp. 144–155. Springer, Heidelberg (2006)
18. Tjong Kim Sang, E.F.: Introduction to the conll-2002 shared task: Language-independent named entity recognition. In: Proceedings of CoNLL 2002, Taipei, Taiwan, pp. 155–158 (2002)
19. Carreras, X., Màrques, L., Padró, L.: Named entity extraction using adaboost. In: Proceedings of CoNLL 2002, Taipei, Taiwan, pp. 167–170 (2002)
20. Surdeanu, M., Màrquez, L., Carreras, X., Comas, P.: Combination strategies for semantic role labeling. JAIR 29, 105–151 (2007)
21. Hacioglu, K., Pradhan, S.S., Ward, W.H., Martin, J.H., Jurafsky, D.: Semantic role labeling by tagging syntactic chunks. In: Ng, H.T., Riloff, E. (eds.) HLT-NAACL, Workshop CoNLL 2004, Boston, USA. ACL (May 2004)
22. Roche, E., Schabes, Y.: Deterministic part-of-speech tagging with finite-state transducers. Comput. Linguist. 21(2), 227–253 (1995)

Unsupervised Part-of-Speech Disambiguation for High Frequency Words and Its Influence on Unsupervised Parsing

Christian Hänig

University of Leipzig
Natural Language Processing Group
Department of Computer Science
04103 Leipzig, Germany
christian.haenig@yahoo.de

Abstract. Current unsupervised part-of-speech tagging algorithms build context vectors containing high frequency words as features and cluster words – regarding to their context vectors – into classes. While part-of-speech disambiguation for mid and low frequency words is achieved by applying a Hidden Markov Model, no corresponding method is applied to high frequency terms. But those are exactly the words being essential for analyzing syntactic dependencies of natural language. Thus, we want to introduce an approach employing unsupervised clustering of contexts to detect and separate a word's different syntactic roles. Experiments on German and English corpora show how this methodology addresses and solves some of the major problems of unsupervised part-of-speech tagging.

1 Introduction

Part-of-speech tagging is a crucial prerequisite for natural language processing (NLP) and thus, it is often one of the first steps to bring structure into unstructured data. Hence, the accuracy of the applied POS tagger influences the performance of the complete workflow significantly. Until a few years ago, research on this area focused on creating algorithms to derive tagger models from hand annotated corpora. Approaches using decision trees as in [Schmid, 1994] or Markov Models as in [Brants, 2000] achieve very high accuracies of about 96% in comparison to the corresponding gold standard. Although several high quality POS taggers exist for the most important languages, two major problems persist: the dependency on hand annotated corpora for training and the restriction to a fixed tag set.

Taggers trained on relatively formal resources of high quality like newswire do not perform very well when being applied to other kinds of text. Today a lot of data, spread over a variety of text types like domain specific texts in industrial (e. g. repair order texts in the automotive domain) or colloquial resources (e. g. forum and blog entries, chat logs, emails), exists. In domain specific data taggers loose accuracy due to a high portion of unknown terminology and special syntactic constructions which do not appear in the tagger's training data. The creation and enrichment of language resources for every domain cannot be handled in the long run because this is very time and money consuming.

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 113–120, 2010.

While language in domain specific data is restricted, the opposite is the case for colloquial sources. The internet offers nearly unlimited amounts of data containing a lot of slang, jargon and neologisms. Additionally, new word classes like emoticons arise. Conventional taggers use predefined tag sets which do not provide the possibility to assign correct tags to new occurrences like emoticons and internet slang.

Unsupervised methods can handle these phenomena and adapt to the accelerated change of language without the need for huge annotated corpora for training purposes.

1.1 Related Work

Different approaches to unsupervised part-of-speech tagging (e. g. [Schütze, 1995], [Freitag, 2004] and [Biemann, 2006b]) exist. They do not need any annotations and induce word classes – defined by the textual data – themselves. Thus, they can deal with languages and textual data for which no resources are available yet. As they are not restricted to fixed tag sets, new classes like emoticons can be explored.

Approaches to unsupervised POS tagging cluster words regarding to their context similarity. Context vectors are created using the most frequent words as features. Afterwards, a cluster algorithm is applied to classify the words into different classes. Most algorithms use a fixed number of word classes to which the words are assigned (e. g. as in [Schütze, 1995]), but there are also approaches using graph clustering algorithms providing the possibility to determine the number of resulting clusters themselves (e. g. [Biemann, 2006b]). There is another difference between those two approaches: the applied context size. In [Schütze, 1995], only direct neighbours of a word (in addition to the right/left context vector of the preceding/following word) are considered to be relevant, [Biemann, 2006b] instead regards the two left and right neighbours of that word. An experiment in section 3.1 will evaluate the influence of context size.

Although the induced word classes show high similarity to the well known parts-of-speech, they differ in granularity. In general, induced word classes are more granular than parts-of-speech. This can be observed on Named Entities, which are often split into different groups (e. g. city names, female resp. male first names and last names) and for different grammatical cases of nouns. Even word clusters containing only one word occur.

1.2 Motivation

Along with the advantages of unsupervised tagging, several problems emerge. Different syntactic functions of low frequency words are disambiguated, because POS tags are assigned indirectly (e. g. using a HMM as in [Biemann, 2006b]).

High frequency words are categorized using a clustering algorithm and thus, they are not disambiguated by current approaches. Tables 1 and 2 show some examples of high frequency words of English (SUSANNE corpus[1]) and German (TIGER corpus, see [Brants et al., 2002]) holding several functions. Those high frequency words influence the structure of a sentence decisively, so it is crucial for further processing to disambiguate their different roles. Statistical approaches to unsupervised parsing (e. g. based

[1] http://www.grsampson.net/RSue.html

Table 1. Examples of different syntactic functions of high frequency words for English

English						
to	Infinitive marker	63%	Preposition	37%		
that	Conjunction	75%	Pronoun	15%	Determiner	10%
as	Preposition	82%	Adverb	18%		
this	Determiner	62%	Pronoun	38%		
about	Preposition	77%	Adverb	23%		

Table 2. Examples of different syntactic functions of high frequency words for German

German						
die	Determiner	89%	Subst. rel. pronoun	10%	Subst. dem. pronoun	1%
das	Determiner	76%	Subst. dem. pronoun	17%	Subst. rel. pronoun	8%
zu	Infinitive marker	64%	Preposition	32%	Particle	4%
auf	Preposition	94%	Particle of separable verb	6%		
als	Preposition	61%	Comp. conjunction	31%	Subord. conj. with sentence	8%

on co-occurrences as in [Hänig et al., 2008] or Data-Oriented Parsing as in [Bod, 2006]) rely on POS tags and are significantly influenced by the tagger's accuracy. Especially the very fine granulation of word classes entails difficulties in parsing algorithms. Examples for word classes containing only a single word are prepositions (e. g. *from*, *with*, *of* for English and *in*, *ab*, *mit* for German) and articles (e. g. *der*, *die*, *das*, *ein* for German). As those words are put into separate clusters instead of being clustered into classes for prepositions or articles, the parsing algorithm has to induce similar rules for each word instead of for each word class. The consequences for grammar induction are lower significances for co-occurrences respectively sub trees and thus, some syntactic dependencies cannot be induced.

To circumvent this effect, a proper part-of-speech tagging providing classification based on word form and contextual disambiguation is essential.

2 Clustering of Contexts

In this section, we want to introduce how clustering of contexts can be used to discover a word's syntactic roles. In current approaches, taggers use information about context to cluster words into classes. As we want to disambiguate different syntactic functions of words, we need to cluster the contexts of a word to identify its various usages. From our corpus linguistic point of view, the local context c_{w_i} of the ith word w_i of a sentence is a vector containing the n left and n right neighbours of w_i. Special tokens are inserted to deal with the beginning and the end of a sentence.

$$c_{w_i} = (w_{i-n}, \ldots, w_{i-1}, w_{i+1}, \ldots, w_{i+n}) \tag{1}$$

The global context C_w of a word w is defined as the sum of all contexts it appears in. As we want to cluster local contexts to separate the global context into clusters representing the word's different syntactic categories, we need to find a proper way to calculate

similarity between local contexts. To calculate the similarity between two words a and b we apply the cosine measure to their global contexts. Similarity sim_{c_a,c_b} between two context vectors c_a and c_b is calculated as weighted average of their component's similarity.

$$sim_{c_a,c_b} = \sum_i w(i) \cdot sim\left(C_{c_{a_i}}, C_{c_{b_i}}\right) \tag{2}$$

A weighting function $w(i)$ is introduced. Intuitively, direct neighbours of a word seem to be more important than distant ones. In this paper, we analyze and discuss four possible functions:

Uniform. Each component of the context vector has the same weight.

$$w(i) = \frac{1}{2n} \tag{3}$$

Linear descending. Direct neighbours of the word have more influence than distant ones. The weight decreases linearly in both directions of the context.

$$w(i) = \frac{n - \left|\frac{2n-1}{2} - i\right| + \frac{3}{2}}{n \cdot (n+1)} \tag{4}$$

Exponential. Direct neighbours of the word have more influence than distant ones. The weight decreases exponentially in both directions of the context.

$$w(i) = \frac{2^{n - \left|\frac{2n-1}{2} - i\right| + \frac{1}{2}}}{2^{n+1} - 2} \tag{5}$$

Linear ascending. Contrary to the other functions, the weight increases (linearly) with the distance of the context components from the word. This function does not respect the assumption that direct neighbours are more important than distant ones and is used to prove this in the experiments.

$$w(i) = \frac{\left|\frac{2n-1}{2} - i\right| + \frac{1}{2}}{n \cdot (n+1)} \tag{6}$$

Each of the weighting functions depends on the context width n, hence we did experiments for several values of n to investigate its influence.

As we do not know a priori, how many functions a word can hold, we use a graph clustering algorithm determining the number of clusters itself. We choose Chinese Whispers ([Biemann, 2006a]), which has been proven to be applicable in NLP.

3 Experiments

In preparation for evaluation we transformed both corpora changing each word's POS tag to its most frequent one. This simulates unsupervised POS tagging without disambiguation and we maintain the possibility of easy comparison with the gold standard. This also provides us with convenient visualization as word classes induced by unsupervised taggers do not have familiar labels due to missing knowledge about which cluster contains which part-of-speech. Thus, it is easier to estimate resulting cluster maps.

3.1 Context Size and Weighting Function

We want to evaluate the influence of context size and the applied weighting function. In our scenario the impact of the proposed weighting functions, which are very similar to each other for small n, strongly depends on context size. We calculated the *cluster purity* for each possible combination to find the most reasonable one. Cluster purity p_{c_i} for a cluster c_i is defined as

$$p_{c_i} = \frac{1}{|c_i|} max_k \left(|c_i|_{class=k} \right) \tag{7}$$

where $|c_i|$ is the cluster size of c_i and $|c_i|_{class=k}$ denotes the number of items of class k assigned to cluster c_i. The overall purity P of a clustering of dataset D is the weighted sum of all individual cluster purities.

$$P = \sum_i \frac{|c_i|}{|D|} p_{c_i} \tag{8}$$

To prove the success of our approach, we created the following experiment:

```
init empty tagger model

for each of the most frequent 100 words:
  collect contexts
  calculate similarities
  cluster contexts
  add rules to tagger model

tag corpus
```

The baseline for the experiment outlined above is for English 0.931 and for German 0.921. Results are listed in Table 3. Best results are achieved using the exponential weighting function, it performs best regardless of the size of n.

A context size of 2 seems to be appropriate as it yields the best combined purity for both languages. Due to similarity of *linear descending* and *exponential* weighting function for $n = 2$, we recommend to use the less complex *linear descending* function for faster computation. For English, a smaller value of n will perform worse and bigger

Table 3. Cluster purity depending on context size and applied weighting function

	uniform	lin. desc.	exp.	lin. asc.
English				
$n = 1$	**0.932**	**0.932**	**0.932**	**0.932**
$n = 2$	0.934	**0.936**	**0.936**	0.933
$n = 3$	0.933	0.935	**0.936**	0.933
German				
$n = 1$	**0.941**	**0.941**	**0.941**	**0.941**
$n = 2$	0.938	**0.940**	**0.940**	0.926
$n = 3$	0.925	0.928	**0.929**	0.923

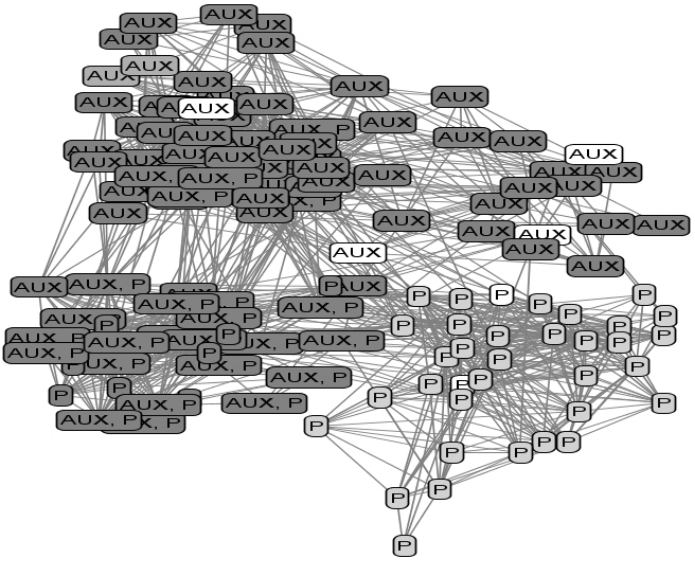


Fig. 1. Resulting clusters for *to*

values effect results negatively for German. To reduce complexity of context clustering, only contexts with a minimum frequency of 5 were taken into account. Thus, for larger n, less contexts will be available for clustering. Using large corpora will bypass this effect.

Although baselines are very high, clustering contexts reduces the gap to perfect POS tagging in comparison to the gold standard. Since cluster purity can be gamed by putting each word into a separate class, we additionally want to discuss the influence on parsing. The resulting clusters for *to* are given in Figure 1. Two huge clusters are found, one containing only prepositions (P; bottom right; light grey), the other one contains principally particles marking the following verb as infinitive (AUX; top and bottom left; dark grey). While the smaller cluster is absolutely pure, the other one also contains prepositions. Unsupervised POS tagging would tag all occurrences of *to* the same way – using context clustering, we find two different roles and thus, improve tagging accuracy. Furthermore, parsing algorithms can distinguish between *to* used as infinitive marker or as preposition which facilitates the detection of rules for *to* as preposition. Otherwise those rules would be hidden behind the most common usage of *to* as infinitive marker due to less significant co-occurrences.

3.2 Single Word Clusters

Unsupervised POS tagging without proper disambiguation of different syntactic functions often creates clusters containing only one word, because similarities to other words are too low due to multiple roles. In this section, we want to show that clustering contexts can fix this problem and increases similarity between clusters containing similar syntactic roles.

Table 4. Similarities of German prepositions without disambiguation

	ab	aus	mit	vor	zu
ab	-	0.763	0.662	0.651	0.428
aus		-	0.953	0.821	0.587
mit			-	0.810	0.603
vor				-	0.513
zu					-

Table 5. Similarities of German prepositions with disambiguation

	ab	aus	mit	vor	zu
ab	-	0.932	0.740	0.921	0.914
aus		-	0.957	0.993	0.977
mit			-	0.820	0.972
vor				-	0.784
zu					-

In our experiment, we disambiguate five prepositions of German which can also be used as particle part of separable verbs. Similarities between those prepositions were calculated and gathered in Table 4. Most values are not very high, hence they do not tend to be clustered. After disambiguation, the similarity values for prepositional use are significantly higher as it can be seen in Table 5.

Induced tag sets often have a larger size (300 - 500 word classes) than manually annotated ones. Being able to reduce the portion of single word clusters will improve the manageability and acceptance of the induced tag set. Furthermore, without having similar parts-of-speech distributed over many classes, manual labeling of those word classes is facilitated and can be used for semi-supervised part-of-speech tagging without the need for training data.

4 Conclusion and Further Work

We showed that it is possible to disambiguate different syntactic functions of words in a completely unsupervised manner using clustering of contexts. This improves unsupervised POS tagging in several ways. Firstly, disambiguation will be possible to all words and not only to low frequent ones. Secondly, the induced word classes will show higher purity and their number will be reduced as the different functions of words can be separated and assigned to corresponding classes. Thirdly, processes based on POS tags – especially parsing – will yield better results with proper POS tagging as they do not have to deal with different functions within one word class.

In the future, we want to extend existent approaches to unsupervised POS tagging with clustering of contexts and build a new unsupervised tagger. Additionally, we want to evaluate the best way, how to deal with contexts that do not appear in the training corpus. Classifying them using maximum likelihood estimation achieves good results, but we also want to analyze the performance of decision trees.

References

[Biemann, 2006a] Biemann, C.: Chinese whispers - an efficient graph clustering algorithm and its application to natural language processing problems. In: Proceedings of TextGraphs: the Second Workshop on Graph Based Methods for Natural Language Processing, New York City, USA, pp. 73–80 (2006a)

[Biemann, 2006b] Biemann, C.: Unsupervised part-of-speech tagging employing efficient graph clustering. In: Proceedings of the COLING/ACL 2006 Student Research Workshop, Sydney, Australia (2006b)

[Bod, 2006] Bod, R.: An all-subtrees approach to unsupervised parsing. In: ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, Morristown, NJ, USA. Association for Computational Linguistics (2006)

[Brants et al., 2002] Brants, S., Dipper, S., Hansen, S., Lezius, W., Smith, G.: The TIGER treebank. In: Proceedings of the Workshop on Treebanks and Linguistic Theories (2002)

[Brants, 2000] Brants, T.: Tnt - a statistical part-of-speech tagger. In: Proceedings of the Sixth Applied Natural Language Processing (ANLP 2000), Seattle, WA (2000)

[Freitag, 2004] Freitag, D.: Toward unsupervised whole-corpus tagging. In: COLING 2004: Proceedings of the 20th international conference on Computational Linguistics, Geneva, Switzerland (2004)

[Hänig et al., 2008] Hänig, C., Bordag, S., Quasthoff, U.: Unsuparse: Unsupervised parsing with unsupervised part of speech tagging. In: Proceedings of the Sixth International Language Resources and Evaluation (LREC 2008), Marrakech, Morocco (2008)

[Schmid, 1994] Schmid, H.: Probabilistic part-of-speech tagging using decision trees. In: Proceedings of the International Conference on New Methods in Language Processing, Manchester, UK (1994)

[Schütze, 1995] Schütze, H.: Distributional part-of-speech tagging. In: Proceedings of the 7th EACL conference (1995)

A Machine Learning Parser Using an Unlexicalized Distituent Model

Samuel W.K. Chan, Lawrence Y.L. Cheung, and Mickey W.C. Chong

Dept. of Decision Sciences
Chinese University of Hong Kong
Shatin, Hong Kong SAR
swkchan@cuhk.edu.hk

Abstract. Despite the popularity of lexicalized parsing models, practical concerns such as data sparseness and applicability to domains of different vocabularies make unlexicalized models that do not refer to word tokens themselves deserve more attention. A classifier-based parser using an unlexicalized parsing model has been developed. Most importantly, to enhance the accuracy of these tasks, we investigated the notion of *distituency* (the possibility that two parts of speech cannot remain in the same constituent or phrase) and incorporated it as attributes using various statistic measures. A machine learning method integrates linguistic attributes and information-theoretic attributes in two tasks, namely sentence chunking and phrase recognition. The parser was applied to parsing English and Chinese sentences in the Penn Treebank and the Tsinghua Chinese Treebank. It achieved a parsing performance of *F*-Score 80.3% in English and 82.4% in Chinese.

Keywords: parsing, distituency, unlexicalized model, machine learning.

1 Introduction

Most of the state-of-the-art statistical parsers work with a lexicalized grammatical model that utilizes the information and statistics of word tokens (Charniak 2000, Collins 2003). Lexicalized models are reported to offer performance enhancement over unlexicalized models by 2% to 10+% in parsing (Charniak 1997, Klein and Manning 2003). While word tokens undeniably provide instructive information to parsing, there are still good reasons to study unlexicalized parsing models that do not directly use word tokens in parsing. Unlexicalized grammar models not only make parsers more efficient but also make the estimation of probabilities more reliable with lesser worry about data sparseness even when the language resources are more limited. Language resource constraints are practical issues that should not be underestimated. For instance, most parsers are trained and evaluated using textual data with vocabulary of the same domain, typically, financial news vocabulary in the Wall Street Journal of the Penn Treebank (Marcus *et al*. 1993). However, the vocabulary of the input texts (say, novels) could be quite different from those from the training treebanks. It would be unrealistic to constantly build treebanks for various genres and

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 121–136, 2010.

registers to accommodate the ever-evolving vocabularies. Unlexicalized parsing models are arguably more adaptive even when the vocabulary changes significantly. Furthermore, the potential of unlexicalized grammar models has sometimes been underestimated. Klein and Manning (2003) demonstrated that using an unlexicalized probabilistic context-free grammar (PCFG), their parser outperformed early lexicalized PCFG parsers, though it was not as good as current state-of-the-art parsers.

Another crucial issue to be dealt with in this research is the investigation of *distituency* in the identification of phrases in bottom-up parsing. It indicates when two part-of-speech (POS) tags cannot co-occur in the same constituent or phrase. The investigation of distituency is crucially important in this study because parsing models without word token attributes must be enhanced with other types of attributes. Early rule-based parsers (e.g. shift-reduce parsers) rely entirely on matching grammar rules with the input POS tags and sentence words. More recently, statistical parsing models add rule/tree probability as another criterion for phrase detection and selection (Charniak 1997, Collins 2003). Chunk-based parsers exploit word sequence, POS tag pattern and *n*-gram to tag phrases (Magerman 1995, Sang 2001). Distituency provides another kind of important information that has been shown to be useful in phrase boundary identification (Church *et al.* 1989, Magerman and Marcus 1990). Yet, it has received no serious attention in the parsing literature. We will show that distituency can be integrated effectively into our parser to improve the unlexicalized parsing model.

To demonstrate the potential of distituency in unlexicalized parsing, we present a parser that is specially designed to exploit mutual information (MI) together with some other attributes in the identification of phrase boundaries (hereafter, we call "chunking points"). Unlike popular statistical parsers, our parser does not rely on any PCFGs. Our design is, in some ways, similar to chunk-based parsers (Ramshaw and Marcus 1995, Sang 2001, Tsuruoka and Tsujii 2005). However, it differs significantly from chunk-based parsers in many different ways. *First*, we propose a richer set of attributes, including distituency in terms of mutual information and likelihood ratio, to enhance the accuracy of parsing. Distituency estimates the potential of fragments not forming a phrase. *Second*, in most chunk-based parser design, IOB-tags are directly applied to each word, which is a $(2m+1)$-ary[1] classification. Our parser, however, considers the chunking point of two neighboring phrases as a binary classification problem. As a result, it significantly reduces the target classes and provides flexibility in attribute selection. The reduction eases the training errors produced in the classifier. *Third*, most chunk-based parsers use word-token probability in their tagging algorithms, but our unlexicalized parser does not.

The organization of the paper is as follows. The related work in parsing is first described in Section 2. In this research, we make use of a machine learning technique to devise a classifier which is based on a string of POS tags and their collocation information-theoretic measures. The classifier is then capable of predicting whether there is a boundary between two phrases. Section 3 shows the detailed architecture of the parser. The parser has already been implemented using the Java language. In order to demonstrate the capability of our system, the parser was applied to parsing English and Chinese sentences using the English Penn Treebank (Marcus *et al.* 1993) and the

[1] m stands for the total number of non-terminal phrase types (e.g. NP, VP, etc.).

Tsinghua Chinese Treebank (Zhou 2003, 2004) respectively. A detailed evaluation is given in Section 4. Further directions and possible enhancement of the parser are also hinted in Section 5, followed by a conclusion.

2 Related Work

This section reviews major approaches to parsing with particular attention to the role of word token information in data-driven parsing models. The use of mutual information (MI) in natural language parsing will also be reviewed towards the end of this section.

Statistical Parsers

Statistical parsers associate grammar rules/parses with probabilities estimated from treebanks. Probabilistic parsing enables the use of probabilities to reduce parse search space and select the most probable parse. Due to massive potential structural ambiguity, PCFGs are often not enough to resolve ambiguity. Parsing models usually take advantage of lexical information and phrase heads to disambiguate parses. As the inclusion of lexical information gives rise to data sparseness of PCFG rules, various strategies are necessary to combat the problem. Machine learning methods make it possible for parsers to simultaneously consider multiple types of information in parsing. Collins (1997, 1999, 2003) and Charniak (2000) approximated the conditional probability estimation by making the lexical independence assumption in the language model using a 0-th order Markov grammar. They used chart parsing algorithms for PCFGs. Charniak (2000) further applied a "maximum-entropy-inspired" model to combine many different conditioning features and obtained an *F*-score of 89.5%. On the whole, state-of-the-art English parsers generally achieve an *F*-score of 86%—90%. Bikel (2004) and Huang (2009) subsequently ported Collins' parser and Charniak's parser to Chinese respectively. Using the Penn Chinese Treebank (Xue *et al*. 2005), the former obtains an *F*-score 81.2%, and the latter 82.2%. The initial results indicate that with similar parsing strategies, Chinese is harder to parse than English.

Classifier-based Shift-Reduce Parsers

Apart from PCFGs, shift-reduce parsing recently revived, partly due to the help of classifiers. Ratnaparkhi (1999) adopted a design similar to a shift-reduce parser but used a maximal entropy learning model to control when the reduce step is applied to form phrases. The head word information is one of the contextual features of the learning algorithm. More recently, Sagae and Lavie (2006) developed a classifier-based shift-reduce parser including the lexical feature, producing an *F*-score of 87.9%.

Chunk-based Parsers

Chunk-based parsing originates from chunking which was proposed (Church 1988, Abney 1991) as a fast and robust means to identify major non-recursive phrases. It is considered to be partial parsing because it does not offer a complete syntactic analysis between chunks. Abney (1991, 1996) adapted the idea of chunking to *full* syntactic parsing using finite transducers. Chunk-based parsing operates by repeated application

of chunking by levels. Magerman (1994, 1995) laid the foundation of lexicalized statistical parsing by recasting parsing as a pattern recognition task. The decision tree algorithm integrated the attribute of word/head tokens in the model. Inspired by Ramshaw and Marcus (1995) and Magerman (1995), chunk-based parsing has often been cast as a tagging problem. Apart from POS tags, each word is assigned an additional tag to indicate the relative position of the word in a chunk of type X: Beginning of chunk (B-X), Inside chunk (I-X), or Outside chunk (O). Chunkers assign IOB tags based on the input words and their POS tags. Instead of searching with dynamic programming, the key is to develop strategies to assign IOB tags. Sang (2001) utilized memory-based learning algorithm to assign IOB tags and obtained 80.49% *F*-score. Tsuruoka and Tsujii (2005) adopted a sliding-window approach to collect potential chunks, and achieved a better *F*-score of 86.20%. Fung *et al.* (2004) utilized the Maximum Entropy model and Penn Chinese Treebank to perform word segmentation, POS-tagging and chunking, and achieved an *F*-score of 79.56%. Though the performance of chunk-based parsers lags behind the state-of-the-art parsers, an advantage of chunk-based parsing is its simplicity.

Dependency Parsers

The above parsers all work with phrase structure grammars. There is another set of parsers that works with a different grammar formalism called "dependency grammars." The syntactic structure is described purely in terms of binary relations between a head word and non-head word(s) without non-terminal nodes (or constituent relations). Some also assign grammatical relations to the edges linking words. Lexical information generally plays a very crucial role in dependency parsing model. Dependency parsers can be divided into two major paradigms, namely transition-based parsers and graph-based parsers (Nivre *et al.* 2007). The most common model for transition-based parsers is one inspired by shift-reduce stack-based parsing, first explored by Yamada and Matsumoto (2003). As the transitions between different states are usually non-deterministic (i.e. more than one valid transition to the next state), a classifier trained on a dependency treebank is used to determine the sequence of transitions (Kübler *et al.* 2009). Typical features include word token, POS tag and position in the transition sequence. Different types of classifiers were employed such as support vector machine (Yamada and Matsumoto 2003, Sagae and Lavie 2005) and memory-based learning (Nivre and Scholz 2004, Sagae and Lavie 2005). While transition-based parsers use training data to learn how to derive dependency graphs, the goal of graph-based parsers is to acquire a model that produces a good dependency graph corresponding to the input sentence. They define a scoring function over various possible parses. The work was pioneered by McDonald *et al.* (2005). Much work has then been done on improving the approach since then. In the 2007 CoNLL Shared Task of dependency parsing, the best English dependency parser (Carreras 2007) is based on graph-based dependency parsing.

Distituency, Mutual Information and Parsing

Though distituency has not been discussed in parsing often, it was actually studied early on by Magerman and Marcus (1990) and Brill *et al.* (1990). They measured the MI of two neighboring fragments, X (n_1-gram) and Y (n_2-gram). The MI value becomes minimum when X and Y are in two different constituent, i.e. X and Y form a

distituent. In this way, the parser can decide where the phrase boundaries fall. Unfortunately, in 1990, annotated gold standard Penn Treebank was still not available. The researchers were not able to carry out systematic evaluation on the effectiveness of their parsers. They reported that the parser worked quite well with short sentences with simple structures but became degraded in longer sentences. Given that all they had was the MI of fragments, we believe that it is worthy re-examining its effectiveness. Drábek and Zhou (2000) developed a classifier-based shift-reduce Chinese parser that utilizes both POS attributes and word-based MI attributes. It obtained 88.2% unlabeled precision and 87.5% unlabeled recall. It is noteworthy that MI has been used fairly widely in Chinese word segmentation (Sproat and Shih 1990, Chen *et al*. 1997 among others). It is in many ways similar to phrase boundary identification but it is about word, or character, boundary.

To sum up, almost all practical parsers reviewed directly use word tokens and phrase head tokens in their parsing models. However, as Klein and Manning (2003) demonstrated, the performance of unlexicalized parsing models should not be underestimated. In the following, we will show that distituency is a useful source of information that can help phrase identification.

3 Classifier-Based Parsing

Our parser is based on chunk-based parsing which is a bottom-up derivation strategy. Instead of having a single pass over the input string of words, the parser starts from a string of POS without any hints from words. As in other similar approaches (Ramshaw and Marcus 1995, Sang 2001, Tsuruoka and Tsujii 2005, Sagae and Lavie, 2005), there can be points of ambiguity in the derivation. The first and the foremost is that, with more realistic sentences, it is pretty unclear where is the right chunking point between two adjacent phrases. Second, what is the appropriate syntactic structure for the phrases? In this research, we resolve the ambiguities by identifying the chunking/merging points of the input string based on the word-free context, without any explicit grammar rules. In other words, given a set of word-free context information, namely, only the POS strings and their collocation information-theoretic measures, it is to be decided whether there is a boundary between two phrases. We adopt the ensemble technique in learning the classifier which is built recursively from a set of training data in which the classes are known. In the following, we first give a brief review on the technique, followed by the detailed discussion of the parser.

3.1 Ensemble Learning in Classification

Ensemble learning creates a finite set of classifiers from random sets of training instances and then uses them together for the classification. Empirically, ensembles tend to yield better results and enhance their predictive power when there is a significant diversity among the data. Boosting, a widely used ensemble technique, is an effective method that produces a very accurate prediction rule by combining rough and moderately inaccurate rules of thumb (Schapire and Singer 2000). In boosting, an initial base classifier using a set of training instances having equal weight is constructed. When the prediction of the base classifier differs from the expected outcome, the

Table 1. Adaboost boosting algorithm

Given: $(x_1, y_1), \ldots, (x_m, y_m)$ where $x_i \in X, y_i \in Y = \{-1, +1\}$
Initialize $D_1(i) = 1/m$
For $t = 1, \ldots, T$

- Train a weak learner using distribution D_t
- Get a weak hypothesis $h_t : X \rightarrow \{-1, +1\}$ with error

$$\varepsilon_t = Pr_{i \sim D_t}[h_t(x_i) \neq y_i]$$

- Choose $$\alpha_t = \frac{1}{2}\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$$
- Update:

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha t} & \text{if } h_t(x_i) = y_i \\ e^{\alpha t} & \text{if } h_t(x_i) \neq y_i \end{cases}$$

$$= \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

where Z_t is a normalization factor

Output: $$H(x) = \operatorname{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

weight of this poorly predicted instance increases. A new training data set is then selected randomly from the weighted instances. As a result, the learning of the next classifier pays more attention to the poorly predicted instances. This process continues until a specified number of iterations is reached or a predefined termination condition is met. In brief, the main idea of boosting is to combine many simple and moderately inaccurate categorization rules into a single, highly accurate categorization rule. The simple rules are trained sequentially; conceptually, each rule is trained on the examples that were the most difficult to classify by the preceding rules. The first practical boosting algorithm, AdaBoost, which was introduced by Freund and Schapire (1997), solved many of the practical difficulties of the earlier boosting algorithms. Table 1 illustrates the main idea of the algorithm. Interested readers can refer to the literature for more detailed discussion (Freund and Schapire 1997, Hastie *et al.* 2001).

3.2 Parser Architecture

Our parser is divided into three major modules, namely, (i) chunker, (ii) phrase recognizer, and (iii) learning module. The chunker locates the boundaries of the chunks. The phrase recognizer predicts the non-terminal syntactic class (SC) tag of identified chunks respectively. The learning module acquires the knowledge encoded in treebanks to support various classification tasks. Figure 1 shows the architecture of the parser. The input tag sequence is first fed into the chunker. The phrase recognizer then analyzes the chunker's output and assigns SC tags to identified chunks. The updated tag sequence is fed back to the chunker for processing at the next level. The iteration continues until a complete parse is formed.

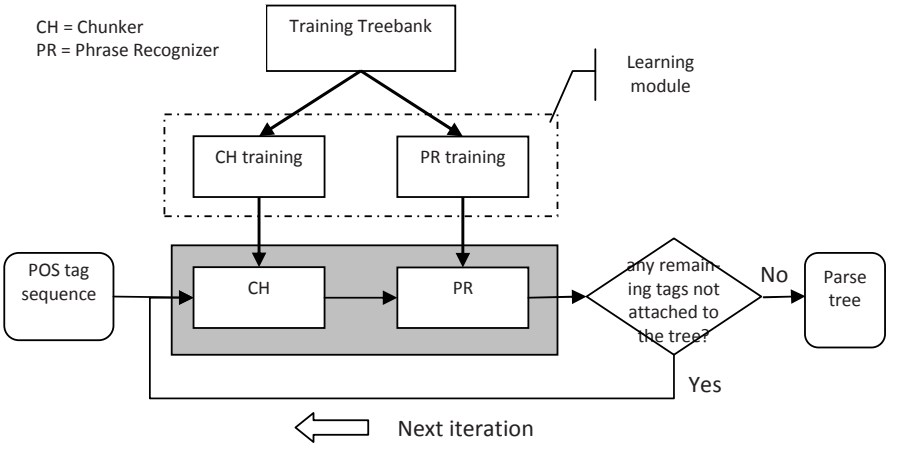


Fig. 1. Architecture of the parser

Chunk/Merge Approach

We explore a new approach that aims at identifying chunk boundaries. Assume that the input of the chunker is a POS tag sequence $<x_0 \ldots x_n \ldots x_m>$ where $0 \leq n \leq m$. Let us define the focus point y_n as the point between two consecutive tags x_n and x_{n+1}. The chunker classifies all focus points as either a *chunking point* or *merging point* at the relevant level. A focus point y_n is a merging point if x_n and x_{n+1} share the same parent node in the target parse tree. Otherwise, y_n is a chunking point. Consider the Penn Treebank POS sequence in (1) and the expected classification of points. Chunking points are marked with "%" and merging points with "+".

(Level 0) `PRP % VBZ % DT % RB + JJ % NN` (1)

Word: he is a very good scientist

The point between `RB` and `JJ` is a merging point because they are siblings of the parent node `ADJP` in the target parse tree. The point between `DT` and `RB` is a chunking point. `DT` and `RB` are not siblings and do not share the same parent node. Chunks are defined as the consecutive tag sequences in the chunker output that are not separated by %. Here is how the notion of distituency comes in. When a focus point y_n is classified as a chunking point, it effectively means that no fragment preceding y_n can combine with any fragment following y_n to form a phrase (i.e. a distituent). By the same procedure, the output strings of Level 1–3 chunking are represented as (2)–(4) respectively.

(Level 1) `NP % VBZ % DT + ADJP + NN` (2)

(Level 2) `NP % VBZ + NP` (3)

(Level 3) `NP + VP` (4)

Table 2. Training attributes for the chunker

Linguistic Attributes (discrete)
POS1P, …, POS3P, POS1F, …, POS3F
Information-theoretic Attributes (continuous)
MI_d_1, …, MI_d_{14}; $MI\_\delta_{1,2}$, $MI\_\delta_{2,3}$, $MI\_\delta_{4,6}$, $MI\_\delta_{5,7}$;
LR_d_1,…, LR_d_{14}; $LR\_\delta_{1,2}$, $LR\_\delta_{2,3}$, $LR\_\delta_{4,6}$, $LR\_\delta_{5,7}$
Target: PointStatus (i.e. `ChunkingPoint` vs `MergingPoint`)

In our training, an attribute vector is constructed for each training case and the corresponding target attribute with one of the binary values, i.e. chunking vs. merging point, is also provided. The set of attributes used in the chunker is listed in Table 2.

The attributes above can be classified into two broad categories. The first type of attributes is linguistic attributes, i.e. the POS/SC tags surrounding the focus point. A sliding window of 6 POS tags is defined as the context of the focus point. Suppose the focus point y_n between x_n and x_{n+1} is considered. The attributes include the 3 tags preceding the focus point (POS1P…POS3P) and 3 tags following the focus point (POS1F…POS3F). The second type is the information-theoretic attributes such as mutual information (MI) and likelihood ratio (LR) of the POS tags, which reflect the likelihood of the fragment collocation. Various adjacent POS/SC fragments in the neighborhood of x_n and x_{n+1} are defined in Table 3. We consider an n-gram as a 2-gram of an n_1-gram and an n_2-gram, where $n_1 + n_2 = n$ (Magerman and Marcus 1990). Table 3 summarizes the information measures computed for training the classifier.

Two kinds of information-theoretic functions ζ, namely, MI and LR, are applied in quantifying the co-occurrence of fragments. MI compares the probability of observing n_1-gram and n_2-gram together to the probability of observing them by chance. The MI of two POS/SC fragments is given in (5).

$$MI(n_1\text{-gram}, n_2\text{-gram}) = \log \frac{P(n\text{-gram})}{P(n_1\text{-gram}) \times P(n_2\text{-gram})} \tag{5}$$

The log LR is a formalization of independence which provides another good measure for the collocation between two POS/SC fragments. While sparseness could be a problem for MI, the LR function is a good complement to it. In applying the LR, two alternative hypotheses are examined.

$$\begin{aligned} &H_0\text{: } P(n_2\text{-gram} \mid n_1\text{-gram}) = p = P(\neg n_2\text{-gram} \mid n_1\text{-gram}) \\ &H_1\text{: } P(n_2\text{-gram} \mid n_1\text{-gram}) = p_1 \neq p_2 = P(\neg n_2\text{-gram} \mid n_1\text{-gram}) \end{aligned} \tag{6}$$

Table 3. Different measures of collocation between adjacent tag fragments

x_{n-3}	x_{n-2}	x_{n-1}	x_n	x_{n+1}	x_{n+2}	x_{n+3}	x_{n+4}	Measure of collocation	n-gram
		x_{n-1}	x_n					d_1: $\zeta(x_{n-1}, x_n)$	2-gram
			x_n	x_{n+1}				d_2: $\zeta(x_n, x_{n+1})$	2-gram
				x_{n+1}	x_{n+2}			d_3: $\zeta(x_{n+1}, x_{n+2})$	2-gram
	x_{n-2}	x_{n-1}	x_n					d_4: $\zeta(x_{n-2}x_{n-1}, x_n)$	3-gram
		x_{n-1}	x_n	x_{n+1}				d_5: $\zeta(x_{n-1}x_n, x_{n+1})$	3-gram
			x_n	x_{n+1}	x_{n+2}			d_6: $\zeta(x_n, x_{n+1}x_{n+2})$	3-gram
				x_{n+1}	x_{n+2}	x_{n+3}		d_7: $\zeta(x_{n+1}, x_{n+2}x_{n+3})$	3-gram
x_{n-3}	x_{n-2}	x_{n-1}	x_n					d_8: $\zeta(x_{n-3}x_{n-2}x_{n-1}, x_n)$	4-gram
	x_{n-2}	x_{n-1}	x_n	x_{n+1}				d_9: $\zeta(x_{n-2}x_{n-1}x_n, x_{n+1})$	4-gram
			x_n	x_{n+1}	x_{n+2}	x_{n+3}		d_{10}: $\zeta(x_n, x_{n+1}x_{n+2}x_{n+3})$	4-gram
				x_{n+1}	x_{n+2}	x_{n+3}	x_{n+4}	d_{11}: $\zeta(x_{n+1}, x_{n+2}x_{n+3}x_{n+4})$	4-gram
x_{n-3}	x_{n-2}	x_{n-1}	x_n					d_{12}: $\zeta(x_{n-3}x_{n-2}, x_{n-1}x_n)$	4-gram
		x_{n-1}	x_n	x_{n+1}	x_{n+2}			d_{13}: $\zeta(x_{n-1}x_n, x_{n+1}x_{n+2})$	4-gram
				x_{n+1}	x_{n+2}	x_{n+3}	x_{n+4}	d_{14}: $\zeta(x_{n+1}x_{n+2}, x_{n+3}x_{n+4})$	4-gram

If c_1, c_2, and c_{12} are the frequencies of n_1-gram, n_2-gram and the n-gram and N is the total number of POS/SC tags in our corpus, the log of the *LR* is calculated as follows:

$$\log LR = \log L(c_{12}, c_1, p) + \log L(c_2 - c_{12}, N - c_1, p) - \log L(c_{12}, c_1, p_1) - \log L(c_2 - c_{12}, N - c_1, p_2) \quad (7)$$

where $L(k,n,x) = x^k(1-x)^{n-k}$ and $p = \frac{c_2}{N}$ $p_1 = \frac{c_{12}}{c_1}$ $p_2 = \frac{c_2 - c_{12}}{N - c_1}$

Finally, the attributes MI_δ_{ij} (i.e. $MI\_d_i - MI\_d_j$) and LR_δ_{ij} (i.e. $LR\_d_i - LR\_d_j$) are introduced to measure the difference between the MI_d_i and MI_d_j. They quantify the collocation disparity between the tags around the focus point.

Here is an example illustrating the set of attributes described above. Take the focus point between RB and JJ in (1) as an example. MI_d_9 represents the MI between (VBZ DT RB) and JJ, i.e. MI(VBZ/DT/RB, JJ). Similarly, MI_d_8 measures the collocation of PRP/VBZ/DT and RB. In addition, two tags, PosHH and PosTT, are introduced to signal the beginning and the end of input line respectively. While the POS tag attributes are discrete, the collocation measures are continuous. The target attribute is either a chunking point or merging point.

Phrase Recognizer

The function of the phrase recognizer is to assign an SC tag to each chunk identified by the chunker. Again, in (1), "RB JJ" constitutes a phrase. The phrase recognizer

Table 4. Ambiguity in phrase recognition where `np`, `dj`, and `vp` represent noun phrase, single clause sentence and verb phrase respectively in the Tsinghua Chinese Treebank

`np → n v`	3.9%
`dj → n v`	91.2%
`vp → n v`	4.9%

Table 5. Attributes for the phrase recognizer

Attributes	*Meaning*
Sentence_Len	No. of words in the sentence
Phrase_Len	Phrase Length = total no. of words in the current chunk
Rel_Pos	Position of the first tag of the chunk (x_1) relative to the sentence (i.e. position of x_1 / total no. of tags at that level – 1)
POS1…POS3	First three tags (x_1 … x_3) of the current chunk
PPhrase	The tag right before the chunk
FPhrase	The tag right after the chunk
PhraseType (Target)	Syntactic Class (SC) of the chunk (e.g. `VP, NP, ADJP`, …)

has to classify the phrase as `ADJP` due to the training instances `ADJP → RB JJ`. The recognizer replaces the phrase with `ADJP`, as shown in (2). The phrase recognizer uses a classifier to learn the rule patterns[2]. The prediction can potentially be made by looking up a rule table. However, there are three obvious short-comings. First, some mappings from child nodes to the parent node are one-to-many. Here is an example from the Tsinghua Chinese Treebank. The sequence "`n v`" (i.e. noun + verb) can be found on the right-hand side of three grammar rules shown in Table 4.

Second, the classifier can predict the SC tag even when it encounters rules that are not covered in the training treebank. Third, the classifier allows the recognizer to take into account attributes more than just the tag values of the child nodes, e.g. tags preceding and following the phrase, phrase length, etc., which would be more difficult to accommodate in the table look-up method. Table 5 shows the attributes used in the training of the phrase recognizer.

4 Experiments and Results

An English parser and a Chinese parser were built. They were trained and tested using the Penn Treebank (Marcus *et al.* 1993) and the Tsinghua Chinese Treebank (Zhou 2003, 2004). Following the convention of previous studies, we pre-processed the trees

[2] Magerman's (1995) SPATTER chunk-based parser also assigns non-terminal node labels to chunks. It is a "lookup table based on the label of the internal node and the labels of the children."

Table 6. Size of training and testing data in two different treebanks

	English Penn Treebank			*Tsinghua Chinese Treebank*		
	Training Data	***Testing Data***	***Total***	***Training Data***	***Testing Data***	***Total***
Word tokens	1,014,129 (94.4%)	60,548 (5.6%)	1,074,677 (100.0%)	354,767 (79.3%)	92,687 (20.7%)	447,454 (100.0%)
Parse trees	39,832 (94.3%)	2,416 (5.7%)	42,248 (100.0%)	32,771 (66.9%)	16,211 (33.1%)	48,982 (100.0%)

in the Penn Treebank by removing NULL elements and functional tags and collapsing `ADVP` and `PRT` into `ADVP`. Table 6 shows the breakdown of training and testing data.

The Penn Treebank has 48 POS tags and 14 SC tags. The Tsinghua Chinese Treebank comes with a tagset consisting of 70 POS tags and 16 SC tags. The performance of the chunker, the phrase recognizer and the overall parsing is reported below. Tables 7 and 8 illustrate the results in our development tests and the overall parsing performance respectively.

Table 7. Development test results

	English Penn Treebank			*Tsinghua Chinese Treebank*		
	Training cases	***Testing cases***	***Accuracy***	***Training cases***	***Testing cases***	***Accuracy***
Chunker	4,931,561	297,411	96.3%	832,673	92,177	93.6%
Phrase Recognizer	3,469,890	210,520	99.2%	260,366	26,033	99.3%

Table 8. Parsing performance (LP = Labeled Precision, LR = Labeled Recall)

Penn Treebank			*Tsinghua Chinese Treebank*		
LP	LR	*F*-Score	LP	LR	*F*-Score
81.6%	79.0%	80.3%	83.5%	81.3%	82.4%

The development test results highlight the individual performance of the chunker and the phrase recognizer. Both modules are pretty robust. For example, the English chunker has an accuracy of 96.3%, and the phrase recognition for both languages is over 99% accurate[3]. As for the overall parsing, the English parser achieves an *F*-Score of 80.3% while the Chinese parser achieves an *F*-Score of 82.4%. It comes with no surprise that there is a performance gap between our English parser (80.3%) and state-of-the-art English parsers (~88–90%) when our parser does not utilize word token information at all. In fact, we do not expect the performance of our unlexicalized parser to outperform that of state-of-the-art lexicalized parsers. Nevertheless, our

[3] The development test results in Table 7 have not taken into account the propagation of errors from lower levels of a syntactic tree.

unlexicalized distituent parsing model performs with a reasonably good accuracy. It is noteworthy that the outcome of the Chinese version seems rather encouraging. Admittedly, as there are little published results that use the relatively new Tsinghua Chinese Treebank, it is difficult to directly benchmark the performance of our Chinese parser. However, we have reasons to believe that our parser is doing quite well. For example, using state-of-the-art lexicalized parsers and the Penn Chinese Treebank, Bikel (2004) and Huang (2009) obtain an *F*-score 81.2% and 82.2% respectively in parsing Chinese sentences. These two studies suggest that porting a parser from English to Chinese without major changes often results in noticeable degradation of performance. However, our current parser works well in parsing sentences from the Tsinghua Chinese Treebank, especially when one takes into account the relatively small size of the Tsinghua Chinese Treebank.

5 Further Work

The presented parser can be improved and optimized further along three dimensions.

Attribute Set Enhancement
We have been investigating adding three types of attributes. *First*, lexicalized parsing models typically percolate the lexical head token up the tree so that the parser can make use of the information to better determine how phrases are formed. To incorporate such propagation mechanism in an unlexicalized model, we can percolate the syntactic head POS up the tree. With the help of a head identification module, the mutual information of head POS can also be utilized to support the chunking-point decision making. *Second*, instead of simply describing a tree using the syntactic label of the top node (e.g. VP, NP), we will enrich the attribute set by including measures of the relative coordinates and the skewness of the phrase, or subtree, in a tree (or called *tree topological* features). Our preliminary study indicates that the tree topological features can provide useful information to improve parsing accuracy. *Third*, certain phrases are more difficult to parse than others. Coordinate structures, for example, in natural languages can span across many child nodes, e.g. $X \rightarrow Y_1,\ldots, Y_{n-1}$ and Y_n. It is not uncommon to find phrases with more than 5 conjuncts in the Tsinghua Chinese Treebank. Even worse, the scope of conjunctions can be rather ambiguous (Magerman 1994). In our preliminary error analysis, they are prone to be chunked incorrectly. Special attributes are probably needed to cater for these phrases. Besides attribute enhancement, we are looking into the integration of *n*-best chunking results for each chunking iteration.

Characteristics of Languages and Treebanks
The existing parser can be fine-tuned to accommodate the variations of different languages and treebanks. *First*, whereas English is largely a right branching language in syntactic tree structure, Chinese is famous for displaying mixed features of both left- and right-branching languages (Li and Thompson 1981). Collins (2003) used a distance measure to allow structural preference in parsing, and was shown experimentally to improve the accuracy noticeably. The tree topological features, as mentioned in the previous paragraph, will be integrated to address the language-specific

tendency. *Second*, the Penn Treebank-style annotation imposes a number of unary branching rules, e.g. SBAR → S, S → VP, NP → RB, etc. They are relatively difficult to predict because they are motivated for theoretical syntax reasons (e.g. null categories, movement traces, etc). Unary branching rules are not found in the Tsinghua Chinese Treebank. This may have made the chunking-point prediction in the Chinese parser more robust. *Third*, the POS tagset size of the English and Chinese treebank differ a lot, i.e. 48 (English Penn Treebank) vs. 70 (Tsinghua Chinese Treebank)[4]. The granularity of tagsets may affect the parsing performance, especially when our parser is highly dependent on POS information to estimate the values for the machine learning attributes. *Fourth*, commas in Chinese are generally a good indicator of phrasal boundary. Chinese has a special enumeration comma "、" to separate items in an enumerated list. The function is fulfilled by commas in English. The division of labor between commas and enumeration commas in the Tsinghua Chinese Treebank may lead to better chunking-point prediction in Chinese[5].

Benchmarking

It would be interesting to benchmark our existing setup with the same parsers with different attributes sets and treebanks. For example, a baseline parser can be set up to use only POS tag attributes so as to estimate quantitatively the contribution by information-theoretic attributes. Another possibility is to add word token attributes to our parser. It will enable us to know the performance difference between the unlexicalized parser and the lexicalized counterpart based on the same machine learning method. Besides, our unlexicalized model is predicted to be less sensitive to lexical variations. Currently, we only studied its performance on Wall Street Journal texts. It would be nice to compare its performance in other types of texts. One can turn to treebanks which are derived from a balanced corpus containing texts from different domains, e.g. Zhou and Sun (1999).

6 Conclusion

This study investigates the use of an unlexicalized parsing model to process English and Chinese. Through the novel method of chunk boundary identification, sentences are segmented based on various POS tags and their distituency measures. Some early studies found that distituency is a useful indicator of phrase boundaries. This paper has articulated a way to combine a heterogeneous set of attributes including linguistic attributes and information-theoretic attributes in refining chunking point detection using a machine learning algorithm. In the experiments, the English version obtains an *F*-score 80.3% and the Chinese version 82.4%. Although we do not expect the unlexicalized parsing engine to outperform lexicalized state-of-the-art parsers, the results of our parser are still very encouraging, especially in parsing Chinese sentences. Certainly, though our evaluation was conducted in English and Chinese, the

[4] For reader's reference, the tagset of the Penn Chinese Treebank used in Bikel (2004) and Huang (2009) is less fine-grained. It has only 33 tags.

[5] Incidentally, the Penn Chinese Treebank only uses one single tag to cover all punctuation marks. Bikel (2004) and Huang (2009) could not take advantage of the cues from different punctuation marks.

computational method is language-independent and can be easily adapted to different languages. Given the ever-evolving vocabularies in different languages, we have suggested a way to do *light parsing* in a word-free context (as contrasted with heavy parsing which crucially relies on word tokens) without being bogged down in various language genres.

Acknowledgments

The work described in this paper was partially supported by the grants from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project Nos. CUHK440607 and CUHK440609). The Tsinghua Chinese Treebank ver 1.0 has been made available by the Speech and Language Technologies R&D Center, the Research Institute of Information Technology of the Tsinghua University. We want to thank them for allowing us to use the data for training and testing in this study.

References

Abney, S.: Parsing by Chunks. In: Berwick, R., Abney, S., Tenny, C. (eds.) Principle-Based Parsing. Kluwer Academic, Dordrecht (1991)

Abney, S.: Partial Parsing via Finite-state Cascades. Natural Language Engineering 2, 337–344 (1996)

Bikel, D.: On the Parameter Space of Generative Lexicalized Statistical Parsing Models. PhD dissertation, University of Pennsylvania (2004)

Brill, E., Magerman, D., Marcus, M., Santorini, B.: Deducing Linguistic Structure from the Statistics of Large Corpora. In: Proceedings of the Workshop on Speech and Natural Language, Human Language Technology Conference, pp. 275–282 (1990)

Carreras, X.: Experiments with a Higher-order Projective Dependency Parser. In: Proceedings of the CoNLL Shared Task of EMNLP-CoNLL 2007, Prague, Czech Republic, pp. 957–961 (2007)

Charniak, E.: Statistical Techniques for Natural Language Parsing. AI Magazine 18(4), 33 (1997)

Charniak, E.: A Maximum-Entropy-Inspired Parser. In: Proceedings of NAACL 2000, pp. 132–139 (2000)

Chen, A.T., He, J.Z., Xu, L.J., Gey, F., Meggs, J.: Chinese Text Retrieval without Using a Dictionary. ACM SIGIR Forum 31, 42–49 (1997)

Church, K.: A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text. In: Proceedings of the 1st Conference on Applied Natural Language Processing, ANLP, pp. 136–143 (1988)

Church, K., Gale, W., Hanks, P., Hindle, D.: Parsing, Word Associations and Typical Predicate-Argument Relations. In: Proceedings of the Workshop on Speech and Natural Language, Cape Cod, Massachusetts, October 15-18 (1989)

Collins, M.: Three Generative, Lexicalised Models for Statistical Parsing. In: Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (and Eighth Conference of the European Chapter of the Association for Computational Linguistics), Madrid, pp. 16–23 (1997)

Collins, M.: Head-driven Statistical Models for Natural Language Parsing. Ph.D. thesis, University of Pennsylvania, Philadelphia (1999)

Collins, M.: Head-Driven Statistical Models for Natural Language Parsing. Computational Linguistics 29(4), 589–637 (2003)

Drábek, E., Zhou, Q.: Using Co-occurrence Statistics as an Information Source for Partial Parsing of Chinese. In: Proceedings of Second Chinese Language Processing Workshop, ACL 2000, Hong Kong, October 8, pp. 22–28 (2000)

Freund, Y., Schapire, R.E.: A Decision-Theoretic Generalization of On-line Learning and an Application to Boosting. Journal of Computer and System Sciences 55(1), 119–139 (1997)

Fung, P., Ngai, G., Yang, Y.S., Chen, B.F.: A Maximum-Entropy Chinese Parser Augmented by Transformation-Based Learning. ACM Transactions on Asian Language Information Processing 3(2), 159–168 (2004)

Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Springer, Heidelberg (2001)

Huang, L.Y.: Improve Chinese Parsing with Max-Ent Reranking Parser. Master project report (2009),
http://sca2002.cs.brown.edu/research/pubs/theses/masters/2009/huang.pdf

Klein, D., Manning, C.: Accurate Unlexicalized Parsing. In: Proceedings of the 41st Meeting of the Association for Computational Linguistics, pp. 423–430 (2003)

Kübler, S., McDonald, R., Nivre, J.: Dependency Parsing. Morgan & Claypool Publishers, San Francisco (2009)

Li, C., Thompson, S.: Mandarin Chinese—A Functional Reference Grammar. University of California Press, Berkeley (1981)

Magerman, D.: Natural Language Parsing as Statistical Pattern Recognition. PhD dissertation, Stanford University (1994)

Magerman, D.: Statistical Decision-tree Models for Parsing. In: Proceedings of the 33rd Annual Meeting on Association for Computational Linguistics, pp. 276–283 (1995)

Magerman, D., Marcus, M.: Parsing a Natural Language Using Mutual Information Statistics. In: Proceedings of AAAI 1990, 8th National Conference on AI, pp. 984–989 (1990)

Marcus, M., Santorini, B., Marcinkiewicz, M.: Building a Large Annotated Corpus of English: the Penn Treebank. Computational Linguistics 19(2), 313–330 (1993)

McDonald, R., Crammer, K., Pereira, F.: Online large-margin training of dependency parsers. In: Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL), Ann Arbor, MI, pp. 91–98 (2005)

Nivre, J., Scholz, M.: Deterministic Dependency Parsing of English Text. In: Proceedings of COLING 2004, Geneva, Switzerland, August 23-27, pp. 64–70 (2004)

Nivre, J., Hall, J., Kübler, S., Mcdonald, R., Nilsson, J., Riedel, S., Yuret, D.: The CoNLL 2007 Shared Task on Dependency Parsing. In: Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007, Prague, Czech Republic, pp. 915–932 (2007)

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)

Ratnaparkhi, A.: Learning to Parse Natural Language with Maximum Entropy Models. Machine Learning 34, 151–175 (1999)

Sagae, K., Lavie, A.: A Classifier-Based Parser with Linear Run-Time Complexity. In: Proceedings of the Ninth International Workshop on Parsing Technologies (IWPT), pp. 125–132 (2005)

Sagae, K., Lavie, A.: A Best-First Probabilistic Shift-Reduce Parser. In: Proceedings of the COLING/ACL on Main Conference Poster Sessions, Morristown, NJ, USA, pp. 691–698. Association for Computational Linguistics (2006)

Sang, E.: Transforming a Chunker to a Parser. In: Veenstra, J., Daelemans, W., Sima'an, K., Zavrel, J. (eds.) Computational Linguistics in the Netherlands 2000, pp. 177–188 (2001)

Schapire, R.E., Singer, Y.: BoosTexter: A Boosting-based System for Text Categorization. Machine Learning 39, 135–168 (2000)

Sproat, R., Shih, C.L.: A Statistical Method for Finding Word Boundaries in Chinese Text. Computer Processing of Chinese and Oriental Languages 4(4), 336–351 (1990)

Tsuruoka, Y., Tsujii, J.: Chunk Parsing Revisited. In: Proceedings of the 9th International Workshop on Parsing Technologies, pp. 133–140 (2005)

Xue, N., Xia, F., Chiou, F., Palmer, M.: The Penn Chinese TreeBank: Phrase Structure Annotation of a Large Corpus. Natural Language Engineering 11(2), 207–238 (2005)

Yamada, H., Matsumoto, Y.: Statistical Dependency Analysis with Support Vector Machines. In: Proceedings of the 8th International Workshop on Parsing Technologies (IWPT), Nancy, France, pp. 195–206 (2003)

Zhou, Q.: Build a Large-Scale Syntactically Annotated Chinese Corpus. In: Matoušek, V., Mautner, P. (eds.) TSD 2003. LNCS (LNAI), vol. 2807, pp. 106–113. Springer, Heidelberg (2003)

Zhou, Q.: Annotation Scheme for Chinese Treebank. Journal of Chinese Information Processing 18(4), 1–8 (2004) (in Chinese)

Zhou, Q., Sun, M.: Build a Chinese Treebank as the Test Suite for Chinese Parsers. In: Proceedings of the Workshop MAL 1999 and NLPRS 1999, Beijing, China, pp. 32–36 (1999)

Ontology-Based Semantic Interpretation as Grammar Rule Constraints

Smaranda Muresan

School of Communication and Information, Rutgers University
4 Huntington St, New Brunswick, NJ 08901
smuresan@rci.rutgers.edu

Abstract. We present an ontology-based semantic interpreter that can be linked to a grammar through grammar rule constraints, providing access to meaning during parsing and generation. In this approach, the parser will take as input natural language utterances and will produce ontology-based semantic representations. We rely on a recently developed constraint-based grammar formalism, which balances expressiveness with practical learnability results. We show that even with a weak "ontological model", the semantic interpreter at the grammar rule level can help remove erroneous parses obtained when we do not have access to meaning.

1 Introduction

Semantic parsing maps natural language utterances to formal representations of their underlying meaning. This differs from semantic role labeling [1], or other shallow semantic analysis tasks, which do not produce full formal meaning representations. Recently, several machine learning approaches have been proposed for mapping sentences to their meaning representations [2,3,4,5,6,7,8]. These approaches differ in the mount of annotation required — unsupervised methods that start from syntactic parses [8], supervised methods that require annotation of full sentences [2,3,5,7], supervised methods that require annotation of a small set of representative utterances that can be phrases, clauses or sentences [6]. Moreover, these approaches differ in the meaning representation languages they use — from λ-expressions [3,5,7] and command-like languages [2] to ontology-based representations [6] — and the integration, or lack thereof, of the meaning representations with grammar formalisms — Combinatory Categorial Grammars (CCGs) [9] are used by [3,7], and Lexicalized Well-Founded Grammars [10,11] are used by [6].

Simultaneously, in recent years, there has been significant interest in ontology-based natural language processing , starting from defining ontology-base semantic representations [12], to using ontologies in various applications, such as question answering [13,14], and building annotated corpora, such as the OntoNotes project [15].

In this paper, we present an ontology-based semantic interpreter that can be linked to a grammar through grammar rule constraints, providing access to meaning during parsing, generation and learning. The parser will take as input natural language text and will produce ontology-based semantic representations. We integrate this in a learning framework through the use of our Lexicalized Well-Founded Grammar formalism

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 137–149, 2010.

[10,11], which is a constraint-based formalism, which balances expressiveness with provable learnability results. We present several principles that allows for grammar reversibility and parsing termination (parsing and interpretation intertwine). The semantic interpreter can use either a weak "ontological model" based just on information regarding the semantic roles of verbs, prepositions, the attributes of adjectives, adverbs and also nouns that appear in noun-noun compounds, or a strong "ontological model" based on a hierarchy of concepts and roles. We show that even with a weak "ontological model", the semantic interpreter at the grammar rule level can help remove some of the erroneous utterance parses.

First, we review the Lexicalized Well-Founded Grammar formalism [10,11], emphasizing the representation of language expressions and how semantic composition and interpretation can be encoded as constraints at the grammar rule level. In Section 3, we present the ontology-based semantic interpretation (local vs global interpretation, principles, and the semantic interpreter). In Section 4 we discuss the issue of ambiguity, while in Section 5 we show how the semantic interpreter could be used to build terminological knowledge from text, and show preliminary results on how this interpretation at the grammar rule level can help remove some of the erroneous utterance parses obtained when we do not have access to meaning. We conclude in Section 6.

2 Lexicalized Well-Founded Grammar

Lexicalized Well Founded Grammar (LWFG) is a recently developed formalism that balances expressiveness with practical — and provable — learnability results [10,11] . Formally, Lexicalized Well-Founded Grammars are a type of Definite Clause Grammars (Pereira and Warren, 1980) in which (1) the context-free backbone is extended by introducing a partial ordering relation among nonterminals, 2) grammar nonterminals are augmented with strings and their syntactic-semantic representations, called *semantic molecules*, and (3) grammar rules can have two types of constraints, one for semantic composition (defines how the meaning of a natural language expression is composed from the meaning of its parts) and one for semantic interpretation (validates the semantic constructions at the rule level). The first property allows LWFG learning from a small set of annotated examples. LWFG's learning framework characterizes the "importance" of substructures in the model not simply by frequency, as in most previous work, but rather linguistically, by defining a notion of "representative examples" that drives the acquisition process. The last two properties make LWFGs a type of syntactic-semantic grammars.

2.1 Semantic Molecule

The *semantic molecule* is a syntactic-semantic representation of natural language strings $w' = \binom{h}{b}$, where h is the *head* of the semantic molecule and encodes information required for semantic composition, while b is the *body* of the semantic molecule and it is the semantic representation of the string.

Figure 1 shows examples of semantic molecules for an adjective, a noun and a noun phrase. When associated with lexical items, the semantic molecules are called *elementary semantic molecules* (Figure 1a). When the semantic molecules are built by the

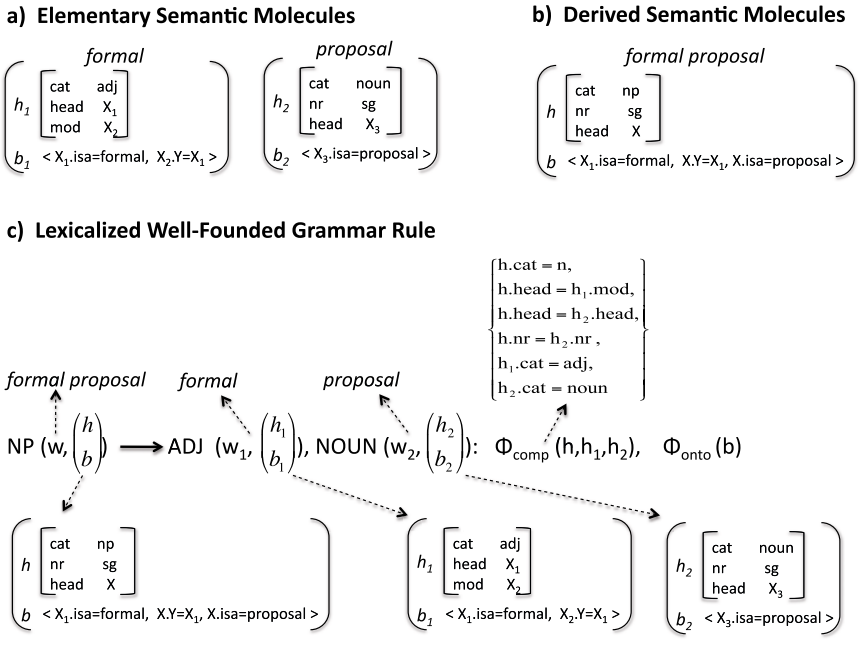


Fig. 1. Examples of two elementary semantic molecules (a), a derived semantic molecule (b) obtained by combining them, and a constraint grammar rule, together with the constraints for semantic composition and semantic interpretation, Φ_{comp} and Φ_{onto}, respectively (c). Φ_{comp} is applied to the heads of the semantic molecules, and is a system of equations, while $\Phi_{onto}(b)$ is the predicate which validates the semantic representation of the string corresponding to the left-hand side nonterminal on the ontology.

combination of others, they are called *derived semantic molecules* (Figure 1b). The head of the semantic molecule, h, is a flat feature structure that has at least two attributes: cat which encodes the syntactic category of the associated string, and head, which represents the head of the string. In addition, feature attributes for agreement and other grammatical features can be present (e.g., nr, pers). The set of attributes is finite and known *a priori* for each syntactic category.

The body of the semantic molecule, b, is a flat semantic representation, called OntoSeR (Ontology-based Semantic Representation). It is a logical form, built as a conjunction of atomic predicates $\langle concept \rangle.\langle attr \rangle = \langle concept \rangle$, where variables are either concept or slot identifiers in an ontology. For example, the adjective *formal* is represented as $\langle X_1.isa = fomal, X_2.Y = X_1 \rangle$, which says that the meaning of an adjective is a concept ($X_1.isa = formal$), which is a value of a property of another concept ($X_2.Y = X_1$) in the ontology.

The lexicon of a LWFG consists of words paired with their semantic molecules (Figure 1a). In addition to the lexicon, a LWFG has a set of constraint grammar rules, where the nonterminals are augmented with pairs of strings and their semantic molecules. These pairs are called syntagmas, and denoted by $\sigma = (w, w') = (w, \binom{h}{b})$, where w

is a natural language string, and $w\prime = \binom{h}{b}$ is its semantic molecule. An example of a LWFG rule for a noun phrase is given in Figure 1c. As it can be seen, grammar nonterminals are augmented with syntagmas. This rule generates the syntagma corresponding to the string *formal proposal* whose semantic molecule is given in Figure 1b.

2.2 Semantic Composition and Semantic Interpretation as Grammar Rule Constraints

In LWFGs, the semantic structures are composed by constraint solving, rather then functional application (with lambda expressions and lambda reduction). Moreover, the semantic interpretation can also be encoded as a constraint at the grammar rule level, providing access to meaning during parsing.

Thus, there are two types of constraints at the grammar rule level — one for *semantic composition* and one for *semantic interpretation*. The composition constraints Φ_{comp}, applied to the heads of the semantic molecules, form a system of equations that is a simplified version of "path equations" [16], because the heads are flat feature structures. These constraints are learned together with the grammar rules. An example of Φ_{comp} is given in Figure 1c.

The semantic interpretation constraints are applied to the body of the semantic molecule associated with the left-hand side nonterminal, interpreting/validating the semantic constructions at the grammar rule level. Assuming an ontology-based interpretation, Φ_{onto} is a predicate which can succeed or fail as a result of querying the ontology — when it succeeds, it instantiates the variables of the semantic representation with concepts/slots in the ontology. For example, given the phrase *formal proposal* in Figure 1c, Φ_{onto} succeeds and returns (X_1=FORMAL, X=PROPOSAL, Y=MANNER), where FORMAL, PROPOSAL, MANNER are concepts and slots in the ontology, respectively, while given the phrase *fair-hair proposal* it fails.

In the next section we discuss the ontology-based interpretation, including a formal definition of the OntoSeR representation, the issue of local vs. global semantic interpretation, several principles that govern the semantic representation and interpretation, and the local ontology-based semantic interpreter, Φ_{onto}.

3 Ontology-Based Semantic Interpretation

The $\Phi_{onto}(b)$ constraint can be seen as a *local semantic interpretation* at the utterance/grammar rule level, providing access to meaning during parsing/generation[1]. It is built using a meta-interpreter with *freeze* [17]. We give the details of this interpreter in Section 3.1.

Before we could talk about the semantic interpreter, and the principles that govern the semantic interpretation, we first give a formal definition for the ontology-based semantic representation, OntoSeR, and discuss the levels of representation needed to get from natural language utterances to knowledge: utterance, text, and ontology levels.

The formal definition of OntoSeR is given below. OntoSeR is a logical form, built as a conjunction of atomic predicates (AP), $\langle concept \rangle.\langle attr \rangle = \langle concept \rangle$. As can be seen

[1] Lexicalized Well-Founded Grammars are reversible grammars.

from this definition, the variables in OntoSeR are either concept IDs or attribute IDs in the ontology. The logical operator lop is the logical conjunction ($\wedge$). The coord operator is one of the linguistic coordinators, such as *and*, *or*, *but*. OntoSeR encodes both ontological meaning (concepts and relations between concepts) and extra-ontological meaning, such as tense, voice, aspect, modality (Figure 2 gives an example of representing tense information for a verb). OntoSeR can be seen as an ontology-query language, which is sufficiently expressive to represent many aspects of natural language and yet sufficiently restrictive to facilitate learning.

$$\begin{aligned}
\langle \text{OntoSeR} \rangle &\stackrel{\text{def}}{=} \langle \text{AP} \rangle \mid \langle \text{OntoSeR} \rangle \, \langle \text{lop} \rangle \, \langle \text{OntoSeR} \rangle \\
\langle \text{AP} \rangle &\stackrel{\text{def}}{=} \langle \text{conceptID} \rangle \, . \, \langle \text{attr} \rangle = \langle \text{concept} \rangle \\
\langle \text{AP} \rangle &\stackrel{\text{def}}{=} \langle \text{conceptID} \rangle = \langle \text{conceptID} \rangle \, \langle \text{coord} \rangle \, \langle \text{conceptID} \rangle \\
\langle \text{concept} \rangle &\stackrel{\text{def}}{=} \langle \text{conceptID} \rangle \mid \langle \text{conceptName} \rangle \\
\langle \text{conceptID} \rangle &\stackrel{\text{def}}{=} \langle \text{logicalVariable} \rangle \\
\langle \text{conceptName} \rangle &\stackrel{\text{def}}{=} \langle \text{lexicalWord} \rangle \\
\langle \text{attr} \rangle &\stackrel{\text{def}}{=} \langle \text{attrID} \rangle \mid \langle \text{attrName} \rangle \\
\langle \text{attrID} \rangle &\stackrel{\text{def}}{=} \langle \text{logicalVariable} \rangle \\
\langle \text{attrName} \rangle &\stackrel{\text{def}}{=} \langle \text{lexicalWord} \rangle \\
\langle \text{coord} \rangle &\stackrel{\text{def}}{=} \langle \text{lexicalCoord} \rangle \\
\langle \text{lop} \rangle &\stackrel{\text{def}}{=} \wedge
\end{aligned}$$

At the utterance/grammar level, *OntoSeR*$^-$ is the semantic representation that corresponds directly to a syntagma σ, before the ontology constraint Φ_{onto} is applied. Both the conceptIDs and attrIDs remain variables. After the application of Φ_{onto} during parsing, the assertional form K_σ of the syntagma σ is obtained. This representation is called *OntoSeR*$^+$. At this level, the attrIDs become constant, while the conceptIDs remain variables to allow further composition to take place (we are still at the utterance level). Both at the OntoSeR$^-$ and OntoSeR$^+$ levels, we can exploit the reversibility of the grammar since both these representations are used during parsing/generation. In Figure 2, we see an example of OntoSeR$^-$ and OntoSeR$^+$ for the utterance *John persuaded the doctor to examine her*. At OntoSeR$^-$, the attrIDs are still variables (i.e, the semantic roles of the verbs *persuaded* and *examine* are still variables: P1, P2, P3 and P4,P5, respectively), while at OntoSeR$^+$ they are instantiated with roles from the ontology (i.e., *ag*, *th*, *prop* and *exp*, *perc*, respectively).

The text level or discourse level representation, *TKR*, represents the asserted representations (K_d). The conceptIDs become constants, and no composition can happen at this level. However, we still have (indirect) reversibility, since TKR represents all the asserted OntoSeRs$^+$. Therefore, all the information needed for reversibility is still present. In Figure 3, we see the TKR for the same utterance *John persuaded the doctor to examine her*. We can see that TKR is the same as OntoSeR$^+$, except that the conceptIDs are constants (e.g., A becomes ˜1, B becomes ˜2).

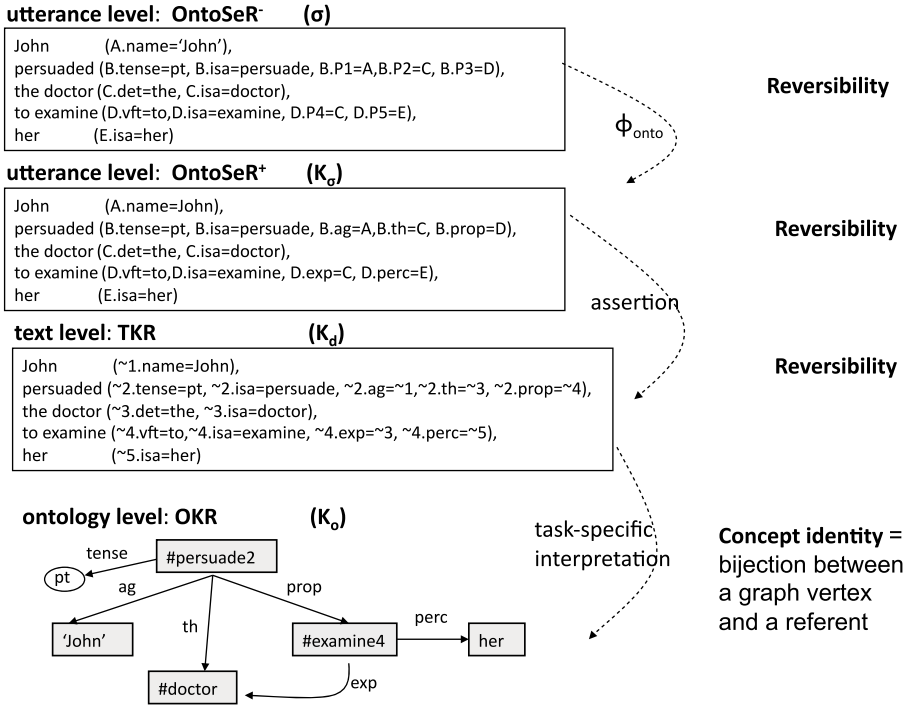


Fig. 2. Levels of representations for the utterance *John persuaded the doctor to examine her*

The knowledge representation at the ontology level, *OKR* (K_o), is obtained after filtering and task-specific interpretation. While Φ_{onto}, which happens at the utterance/grammar level, can be seen as *local semantic interpretation*, the interpretation from TKR to OKR can be seen as a *global semantic interpretation*. For the results presented in this paper, the task-specific interpretation is geared mainly towards terminological interpretation. OKR is a directed acyclic graph (DAG) $G = (V, E)$. Vertices, V are concepts (corresponding to nouns, verbs, adjectives, adverbs, pronouns, cf. Quine's criterion [18, page 496]), or values of extra-ontological properties, such as *past* corresponding to tense property. Edges, E, are semantic roles given by verbs, prepositions, adjectives and adverbs, or are extra-ontological properties, such as tense. At the OKR level we assume the *principle of concept identity* which means that there is a bijection between a vertex in OKR and a referent. For example, if we do not have pronoun resolution, the pronoun and the noun it refers to will be represented as two separate vertices in the graph. In Figure 2, we give an example of OKR for the same utterance *John persuaded the doctor to examine her*. We notice that vertices are either concepts/individuals or values of extra-ontological properties. Determiners, even if represented at the level of OntoSeR, they are not interpreted at the OKR level (they are filtered by the global level interpreter).

Parsing Reversibility Principle. OntoSeR^{-} guarantees parsing reversibility, preserving the entire string meaning. Thus, every syntagma, σ, is independent of the ontology-based interpretation (both local and global interpretation).

Uniform Representation Principle. OntoSeR^{+}, is independent of the knowledge level where the acquisition take place: ontology knowledge and discourse (text) knowledge. Thus, we consider that the assertional form K_σ of a syntagma σ is the same regardless of the asserting level (K_o, K_d), that is, ontology and discourse level, respectively. K_σ is dependent on the interpretation level given by Φ_{onto}. If $\sigma = (w, \binom{h}{b})$, b is guaranteed to preserve the whole meaning of w at the grammar level, while K_σ is dependent on Φ_{onto}, but independent of the knowledge level where the acquisition take place. Thus, the meta-interpreter which perform Φ_{onto} guarantees the interface with different Knowledge Representation Systems (KRS).

Natural Language as Problem Formulation Principle. The discourse (text) knowledge K_d, is only the logic-based problem formulation that can be further solved using logic as problem solving [19]. That is, the meta-interpreter Φ_{onto} does not deal with deep reasoning at the level of K_d assertion. In other words, we are concerned only with the meaning explicitly given in text. Thus, K_d can contain the representation of a paradox formulation in natural language, even if the reasoning about its solution cannot be emphasized. This principle applies only to the local semantic interpreter, Φ_{onto} and not to the global interpreter, where reasoning could take place. This principle assures the tractability of Φ_{onto}, which in turns assures the termination of parsing.

3.1 The Local Ontology-Based Semantic Interpreter

The local semantic interpretation is performed at the rule level through $\Phi_{onto}(b)$, which is built using a meta-interpreter with *freeze* [17]. Given the definition of OntoSeR in Section 3 and the notation $\Phi_{onto}(b) = b'$, the interpretation of OntoSeR is given below:

$$(AP)\,' \leftarrow (postpone\ (AP))\,'$$
$$(OntoSeR_1\ \langle \mathrm{lop} \rangle\ OntoSeR_2)\,' \leftarrow OntoSeR_1'\ \langle \mathrm{lop} \rangle\ OntoSeR_2'$$
$$postpone\ (AP) \leftarrow freeze\ (X \in var\ (AP)\ , AP)$$

The above definition entails that an atomic predicate, AP, is postponed through the $freeze$ predicate until at least one of its variables becomes instantiated. Thus our semantic interpreter is a meta-interpreter with $freeze$ [17]. This allows a nondeterministic efficient search in the ontology. The search strategy of the meta-interpreter is independent of the actual representation of the ontology, allowing an interface with any ontology at the level of atomic predicate meaning. The ontology-based interpretation is not done during the composition operation, but afterwards. Thus, for example, the head of the noun phrase *formal proposal* (Figure 1) does not need to store the slot Y, a

fact that allows us to use flat feature structures for representing the head of the semantic molecule. At this point, when Φ_{onto} is applied, the variable Y becomes instantiated with the value taken from the ontology (e.g., MANNER).

The meta-interpreter can be enhanced with generative ontology[2] axioms [20]: $X' \leftarrow X.isa = X'$, $(X.Y = Z)' \leftarrow X'.Y' = Z'$ (admissible concept rule), $(Y = Z) \leftarrow X.Y = X.Z$ (well-formedness principle for distinct simultaneous roles), $X.Y = Z \leftrightarrow Z.Y^{-1} = X$ (inversion principle), and also with a set of admissible affinities and role relations specified as atomic axioms. The latter refers to the ontologically admissible combinations of concepts and relations (e.g., $event.agt = substance$, $agt.isa = by$).

The OntoSeR is an ontology independent semantic representation, in the same way an ontology is a language independent logical structure. The meta-interpreter allows all the logic operators (i.e., conjunction, disjunction, negation) and provides the soundness of meaning. For negation, the meta-interpreter either adopt the negation as failure strategy of logic programming, or treats negation as atomic predicate that will be handled at the ontology level. The *freeze* interpreting technique provides the soundness of logic programs with negation as failure. Two predicates are implemented for asserting to and querying the ontology respectively. In the querying process, different OntoSeRs can have the same answer, thus transforming the problem of logical equivalence viewed as "meaning identity" [21] into equivalence viewed as concept identity. This ensures the computational tractability requirement for a semantic framework.

Having the local semantic interpreter, Φ_{onto} is important for the disambiguation required for some phenomena (e.g., prepositional phrase attachment, coordinations), and for the semantic interpretation of phenomena not usually analyzed by current broad-coverage grammars or statistical syntactic parsers (e.g., prepositions, noun-noun compounds). Some examples could be seen in the sentences given below (phrases of interest are underlined, and the semantic roles are given in square brackets):

(1) a. Senior U.S. officials were heading to Europe to present a new peace proposal to allies in Germany , Britain and France . [topic]
b. The authority does not want to make a formal proposal to Yankees club owner George Steinbrenner... [manner]
c. Who is that fair-hair proposal writer? [fair-hair modifies writer not proposal]

We discuss the issue of ambiguity in the next section, while in Section 5 we show some preliminary results of how Φ_{onto} could help.

[2] Starting from a skeleton ontology, generative ontologies are formed by rules for combining concepts using semantic roles (binary relations) as binders: "The role relations express possible relations among the nodes in the lattice constituting the ontology. Thereby they make possible the generation of an infinite number of ontological nodes in the lattice, thus establishing a generative ontology. [...] The notion of generative ontology is inspired by the generative grammar paradigm and provides semantic domains for a compositional ontological semantics for NPs containing PPs. In contrast to traditional logical semantics, which strongly emphasizes the semantic contribution of determiners, our ontological semantics places decisive weight on the conceptual semantics of the nominal parts of NPs and their modifiers such as PPs." [20].

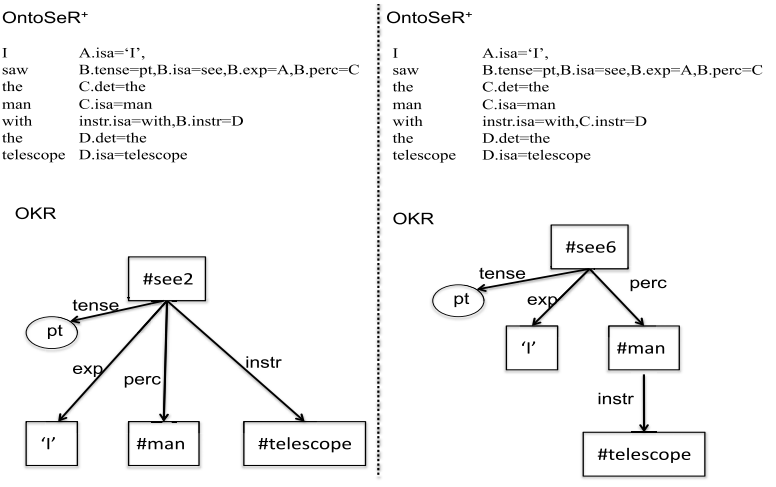


Fig. 3. Two OKRs for *I saw the man with the telescope*

4 Ambiguity

Natural language utterances in isolation could be highly ambiguous. We can have many representations (OntoSeRs/TKRs/OKRs) corresponding to the same utterance. In this case, the robust parser provides all alternatives. Let us consider the classical example:

(2) a. I saw the man with the telescope.

From Figure 3 we can see that this utterance has two OntoSeRs and two ontology level representations (OKRs). This is possible since there are two grammar rules from which this utterance can be derived, and the compositional constraints and the ontology constraints satisfy both alternatives. The ambiguity can be eliminated in this case only if we have discourse context, which will be handled by the global semantic interpreter. In this case, we would have two OntoSeRs and TKRs but only one OKR representation, since the global interpreter, which considers discourse context, will be able to remove the erroneous interpretation. The description of the global interpreter, which could implement discourse context, is outside the scope of this paper.

However there are cases where ambiguities can be eliminated by the use of grammar constraints, providing linguistic or semantic context:

(3) a. the two endocrine glands [located above the kidney] [that secrete hormones and epinephrine]
b. I saw the man with the blue shirt.

In the first example the second relative clause can be attached to the noun *kidney* or the noun *glands*. Since using LWFGs we can model agreement between the head noun and the verb in the relative clause, we have that the relative clause is attached to the noun *glands* (plural). This is achieved through the compositional constraints Φ_{comp}. In the second example, the ambiguity can be eliminated through semantic interpretation

given a strong semantic context that has hierarchies of concepts and roles, as well as selectional restrictions. This way, the Φ_{onto} constraint, based on this strong semantic context, allows only one interpretation: the prepositional phrase *with the blue shirt* is associated with the noun *man* and not with the verb *saw*. In the same way, polysemy can be handled.

5 Results

We have performed a pilot experiment, whose purpose is two-fold: 1) to show that the semantic representation, interpretation and parsing can be used to acquire knowledge from text and to query this knowledge using natural language questions, obtaining precise answers at the concept level; and 2) to show that the local semantic interpretation at the grammar rule level, Φ_{onto}, could help in disambiguation, even if it is based on a weak "ontological model."

The task was reduced to terminological knowledge, where the input text consists of definitions in the medical domain. The grammar was learned using the LWFG learning model described in [11, 6]. Regarding the lexical items, we have a total number of 13 lexical categories (i.e., preterminals, or parts of speech), 46 elementary semantic molecule templates that represent 24 types. For example, the verbs have 5 types of elementary semantic molecules, which gives a total number of 22 different templates (e.g., the type vtnsSem (finite, tensed verb) has three different templates for intransitive/transitive/ditransitive). For grammar learning, only a reduced lexicon is needed (e.g., only a few lexical items are given for every open word class, such as nouns (20), verbs (13, 6 of which are for raising and control verbs), adjectives (14), adverbs (9), proper nouns (4)). For the acquisition/querying experiment we automatically built a larger lexicon from COMLEX [22] and the UMLS lexicon [23], which is a medical lexicon. To learn this grammar we annotated 151 representative examples and 448 examples were used as a representative sublanguage for generalization. Annotating these examples requires knowledge about categories and their attributes. We used 31 categories (nonterminals) and 37 attributes (e.g., category, head, number, person). In this experiment, we chose the representative examples guided by the type of phenomena we wanted to modeled and which occurred in our corpus of medical definitions.

For the weak "ontological model", used only in the acquisition/querying experiment and not during grammar learning, we only used information regarding the semantic roles of verbs, prepositions, attributes of adjectives, adverbs and also nouns that appear in noun-noun compounds (i.e., no synonymy, or hierarchy of concepts and roles). For the semantic roles of verbs and prepositions we extracted the thematic roles from the "LCS Database" [24]. For adjectives and adverbs we used information from WordNet [25]. However, since we used medical definitions, these resources do not contain all the required information and thus we were forced to manually introduce this missing information (especially for adjectives, nouns, and specific roles of prepositions).

Acquisition of a pilot terminological knowledge base. In this experiment we tested the use of the learned grammar and of the semantic interpreter based on a weak "ontological model" to build a pilot terminological knowledge base. Without local semantic validation, Φ_{onto}, the average number of syntagmas (OntoSeR$^-$) obtained by

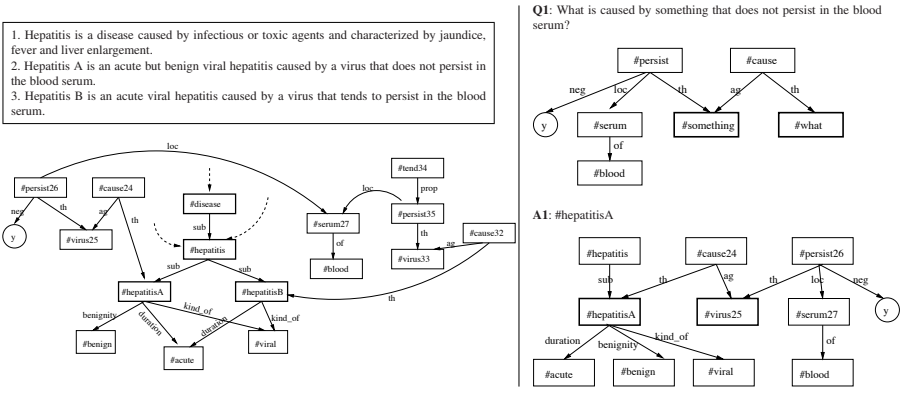


Fig. 4. Example of acquisition and natural language querying of terminological knowledge

the parser is 2.53 per definition. After Φ_{onto} is applied, the average number of different syntagmas (OntoSeR$^+$) obtained for a definition is 2.00. This result shows that even with a weak "ontological model" our semantic interpreter helps remove some erroneous parses. However, it is not enough to obtain only the correct semantic analysis in all cases. Thus, we developed the system to allow a user to manually select the correct OKR, which was then added to the knowledge base. The selection of the OKR-level of representation for human validation is due to the fact that this representation is much more "readable" for a user than the OntoSeR$^-$ and OntoSeR$^+$ levels (as can be seen from Figure 2). This mode of operation allows the semi-automatic creation of OKR-annotated resources, with user validation. Building such a knowledge base could be important for further developing the semantic interpreter towards strong "ontological model", by automatically building an ontology (hierarchy of concepts, and roles) from natural language definitions in different domains. In future work, we will use richer ontologies for our local semantic interpreter.

NL-querying of the acquired knowledge base. For this experiment, we created a benchmark of 29 questions. The type of questions we used are "Who did what to whom?", that is only questions regarding the verbs' arguments. Since in our knowledge base we obtained a hierarchy of concepts (an example of hierarchy is given in Figure 4), the questions can be related to this hierarchy: e.g., the question *Which are viral diseases?* has as answer #HepatitisA and #HepatitisB, even if their direct parent is #hepatitis and not #disease. Since OKR is a direct acyclic graph, the NL-querying is reduced to a graph matching problem. A question is a subgraph of the utterance graph where the wh-word substitutes the answer concept. An answer is a vertex in the OKR of an utterance, together with all the edges incident from/to it. We have experimented both with precise and vague questions. An example of a vague question is *What is caused by something that does not persist in the blood serum?*, where *something* is considered as a variable concept that will match a vertex in the OKR. We obtain precise answers at the concept level (see example in Figure 4). A practical advantage of being able to handle vague questions is that we can obtain all the concepts that are in a particular

relation with other concepts, or that have particular properties. For questions we have an average of 6.06 syntagmas per question at the OntoSeR$^-$ level (i.e., without Φ_{onto} validation). After semantic validation, we have an average of 2.35 syntagmas per question. In this experiment though, even if the weak "ontological model" is not always enough to eliminate incorrect semantic representations of questions, we only obtain the correct answer(s), since we match the OKRs of these questions against the manually validated knowledge base.

6 Conclusions

In this paper we have presented an ontology-based semantic interpreter that is linked to a grammar through grammar rules constraints, providing access to meaning during language processing. We presented several principles that govern the semantic representation and interpretation, principles which are important for parsing reversibility and termination. In a pilot experiment, we showed that the interpreter could be used to acquire terminological knowledge and to query the knowledge using natural language questions, obtaining precise answers at the concept level. We also showed that even with a weak "ontological model", the semantic interpreter is useful to remove some of erroneous utterance parses obtained when we do not have access to meaning. In future work, we plan to use a stronger "ontological model" based on hierarchy of concepts and roles, as well as to enhance the "ontological model" with weights/probabilities.

References

1. Carreras, X., Marquez, L.: Introduction to CoNLL-2004 shared task: Semantic role labeling. In: Proceedings of The Eight Conference on Natural Language Learning (2004)
2. Ge, R., Mooney, R.J.: A statistical semantic parser that integrates syntax and semantics. In: Proceedings of CoNLL 2005 (2005)
3. Zettlemoyer, L.S., Collins, M.: Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In: Proceedings of UAI 2005 (2005)
4. He, Y., Young, S.: Spoken language understanding using the hidden vector state model. Speech Communication Special Issue on Spoken Language Understanding in Conversational Systems 48(3-4) (2006)
5. Wong, Y.W., Mooney, R.: Learning synchronous grammars for semantic parsing with lambda calculus. In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, ACL 2007 (2007)
6. Muresan, S.: Learning to map text to graph-based meaning representations via grammar induction. In: Coling 2008: Proceedings of the 3rd Textgraphs workshop on Graph-based Algorithms for Natural Language Processing, Manchester, UK, August 2008, pp. 9–16 (2008)
7. Zettlemoyer, L., Collins, M.: Learning context-dependent mappings from sentences to logical form. In: Proceedings of the Association for Computational Linguistics, ACL 2009 (2009)
8. Poon, H., Domingos, P.: Unsupervised semantic parsing. In: Proceedings of EMNLP 2009 (2009)
9. Steedman, M.: Surface Structure and Interpretation. The MIT Press, Cambridge (1996)
10. Muresan, S.: Learning constraint-based grammars from representative examples: Theory and applications. Technical report, PhD Thesis, Columbia University (2006)

11. Muresan, S., Rambow, O.: Grammar approximation by representative sublanguage: A new model for language learning. In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, ACL (2007)
12. Nirenburg, S., Raskin, V.: Ontological Semantics. MIT Press, Cambridge (2004)
13. Basili, R., Hansen, D.H., Paggio, P., Pazienza, M.T., Zanzotto, F.: Ontological resources and question answering. In: Workshop on Pragmatics of Question Answering, held jointly with NAACL 2004 (2004)
14. Beale, S., Lavoie, B., McShane, M., Nirenburg, S., Korelsky, T.: Question answering using ontological semantics. In: ACL 2004: Second Workshop on Text Meaning and Interpretation (2004)
15. Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., Weischedel, R.: Ontonotes: The 90% solution. In: Proceedings of HLT-NAACL 2006 (2006)
16. Shieber, S., Uszkoreit, H., Pereira, F., Robinson, J., Tyson, M.: The formalism and implementation of PATR-II. In: Grosz, B.J., Stickel, M. (eds.) Research on Interactive Acquisition and Use of Knowledge, pp. 39–79. SRI International, Menlo Park (1983)
17. Saraswat, V.: Concurrent Constraint Programming Languages. PhD thesis, Dept. of Computer Science, Carnegie Mellon University (1989)
18. Sowa, J.F.: Knowledge Representation: Logical, Philosophical, and Computational Foundations. Brooks Cole Publishing Co., Pacific Grove (1999)
19. Kowalski, R.A.: Logic for Problem Solving. North-Holland Publishing Co., Amsterdam (1979)
20. Jensen, P.A., Nilsson, J.F.: Ontology-based semantics of prepositions. In: Proceedings of ACL-SIGSEM Workshop: The Linguistic Dimensions of Prepositions and their Use in Computational Linguistics Formalisms and Applications (2003)
21. Shieber, S.: The problem of logical-form equivalence. Computational Linguistics 19(1), 179–190 (1994)
22. Grishman, R., Macleod, C., Meyers, A.: COMLEX syntax: Building a computational lexicon. In: Proceeding of 15th International Conference on Computational Linguistics (COLING 1994), Kyoto, Japan (1994)
23. Lindberg, D., Humphreys, B., McCray, A.: The unified medical language system. Methods of Information in Medicine 32, 281–291 (1993)
24. Dorr, B.J.: Large-scale dictionary construction for foreign language tutoring and interlingual machine translation. Machine Translation 12(4), 271–322 (1997)
25. Miller, G.: WordNet: An on-line lexical database. Journal of Lexicography 3(4), 235–312 (1990)

Towards a Cascade of Morpho-syntactic Tools for Arabic Natural Language Processing

Slim Mesfar[1,2]

[1] RIADI, University of Manouba, Tunisia
[2] LASELDI, University of Franche-Comté, France
mesfarslim@yahoo.fr

Abstract. This paper presents a cascade of morpho-syntactic tools to deal with Arabic natural language processing. It begins with the description of a large coverage formalization of the Arabic lexicon. The built electronic dictionary, named "El-DicAr", which stands for "**El**ectronic **Dic**tionary for **Ar**abic", links inflectional, morphological, and syntactic-semantic information to the list of lemmas. Automated inflectional and derivational routines are applied to each lemma producing over 3 million inflected forms. El-DicAr represents the linguistic engine for the automatic analyzer, built through a lexical analysis module, and a cascade of morpho-syntactic tools including: a morphological analyzer, a spell-checker, a named entity recognition tool, an automatic annotator and tools for linguistic research and contextual exploration. The morphological analyzer identifies the component morphemes of the agglutinative forms using large coverage morphological grammars. The spell-checker corrects the most frequent typographical errors. The lexical analysis module handles the different vocalization statements in Arabic written texts. Finally, the named entity recognition tool is based on a combination of the morphological analysis results and a set of rules represented as local grammars.

Keywords: Arabic language, lexical analysis, agglutinative morphology, automatic vocalization, Named Entities Recognition, NooJ linguistic platform.

1 NooJ: A Linguistic Development Platform

NooJ is a linguistic developmental environment which can analyze texts of several million words in real time. It includes tools to construct, test and maintain large-coverage lexical resources, as well as morphological and syntactic grammars. Dictionaries and grammars are applied to texts in order to locate morphological, lexicological and syntactic patterns, remove ambiguities, and tag simple and compound words. NooJ can build lemmatized concordances of large texts from Finite-State or Context-Free grammars, and can accordingly perform cascading transformation operations on texts, in order to annotate the text, or to generate paraphrases.

NooJ is used to formalize five levels of linguistic phenomena: spelling, lexicon, morphology, syntax and semantics. For each level, NooJ proposes a methodology, one or more suitable formalisms, and one or more automatic text analyzers. For example, for the morphological level, NooJ provides two formalisms to describe

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 150–162, 2010.

inflection and derivation, a formalism to describe the lexical morphology and a formalism to write morphological productive rules (e.g. to formalize the creation of neologisms).

Indeed, NooJ is used as a linguistic platform, an information retrieval system, an environment to teach second languages, a terminological extractor, and a tool to teach computational linguistics [17].

Given the explosion of Arabic resources available on-line, with more than 30,000 websites in Arabic, and more than 400 million users[1], we recognized the need to develop an Arabic component for the NooJ platform, which would process and take advantage of these readily available data. We started building the Arabic NooJ module with the purpose of providing automatic analysis of texts written in standard Arabic. This module is formalized as a cascade of morpho-syntactic tools that are used to describe vocabulary and transformational syntax according to the theory of Chomsky [5] and Harris [9], and help to better understand the Arabic language.

Since each linguistic analysis must go through a first step of lexical analysis, which consists in testing membership of each word of the text to the Arabic vocabulary [16], the development of the morpho-syntactic tools cascade begins with formalization of the Arabic vocabulary.

2 Arabic Lexicon Formalization

The NooJ lexical module described throughout this paper relies on some morphological operators performing transformations inside strings, and morphological graphs describing grammatical rules for morphological analysis.

Generally, transformations inside strings are based on the use of some generic predefined commands such as <B> (i.e. keyboard Backspace), <L> (i.e. keyboard Left arrow), <R> (i.e. keyboard Right arrow), … Although these generic commands are predefined in NooJ, we can add new commands. For example, for Arabic, as a highly inflectional and derivational language, it was necessary to define three new operators (<T>, <M> and <Z>) in order to reduce the number of the different verbal inflectional paradigms.

These morphological commands can be associated with two argument types; either a number (e.g. <L3>: go left 3 times) or a "W" (e.g. <LW>: go to beginning of word). They operate on a letter pile, requiring a O(n) transformation time. So, they guarantee a correspondence transformation in a linear time.

Although traditional Arabic grammarians distinguish only three main lexical subgroups (nouns, verbs and particles), we verified that this classification is limited in computational linguistics for the formalization of the Arabic lexicon. So, the class of particles has been extended to include grammatical morphemes, which actually belong to other classes, such as pronouns and demonstratives. This extension resulted in reorganizing four subsets of the Arabic lexicon: verbs, nouns, pronouns and function words [12] and [11].

[1] Sources: http://www.ethnologue.com/ and http://en.wikipedia.org/wiki/Arabic_language

2.1 Verbs

The dictionary of verbs contains 10,000 fully vowelled entries. Since automatic combination between roots and patterns leads to the generation of virtual lemmas or leaves a large number of lexical entries unrepresented and considering that each Arabic root can combine with only a subset of the potential patterns [8], we chose to build a dictionary of lemmas to avoid such problems evoked within the Xerox lexical analyzer. In our case, each entry represents a third person, singular, masculine, perfect verb. These verbs are associated with an inflectional description (among 130 hand-encoded inflectional paradigms for all the verbs).

By inflectional description we refer to the set of possible transformations which produce all inflected forms for a lexical entry (lemma). These inflectional descriptions represent the mood (indicative, subjunctive, jussive or imperative), the voice (active or passive), the gender (masculine or feminine), the number (singular, plural or dual) and the person (first, second or third). On average, there are 122 inflected forms per lexical entry.

2.2 Nouns

We built a dictionary which contains 15,000 primitive nouns[2], such as "كُرْسِيّ" (korsiyy – a chair). Each entry represents a singular noun form deprived of its final vowel. We added into the same dictionary some plural forms which do not have a singular corresponding form, such as "مَخَاوف" (maKaawif – dangers, perils). We also associated derivational descriptions to verbs as described above. Generated forms represent the deverbals[3], such as "إسم الفاعل" (ism al-faa'il - active participle), "إسم المفعول" (ism al-maf'ool - passive participle) or "مصدر" (maSdar - infinitive form) [8].

These nouns, deprived of their final vowel, are associated with inflectional descriptions to generate all inflected nominal forms labeled with linguistic information, such as gender (masculine, feminine or neutral), number (singular, dual and plural) and case (nominative, accusative or genitive). In addition, to generate plural forms from nominal entries, we had to develop about 125 paradigms when describing masculine regular plural, feminine regular plural and irregular or broken plural (جمع التكسير). These paradigms were carefully developed in order to treat certain specificities of the Arabic plural, such as the difference between plurals of small numbers and collective plurals, such as "شَهْر" (shahr – a month), which can have two plural forms: "أَشْهُر" (ashhur – less than 12 months) and "شُهُور" (shuhoor – 12 months and more) [12].

2.3 Pronouns

The pronoun class was introduced as a first extension of the traditional decomposition of the Arabic lexicon. It includes some forms which do not comply with any derivation rule and were considered as nouns. Pronouns form a closed list of words in which we distinguish demonstrative pronouns, such as " أسماء الإشارة " (asmaa' al-ishaarah);

[2] A primitive noun is a noun that does not derive from a verb.
[3] A deverbal is a noun which is derived from a verb.

relative pronouns, such as " أسماء موصولة " (asmaa' mawSoolah); and proclitic "personal" pronouns, such as " ضمائر منفصلة " (Damaa'ir munfaSilah).

2.4 Function Words

Traditionally, function words serve to situate facts or objects in relation to a time or a place. They play a key role in the coherence and sequencing of a text, such as particles that designate a time بعد (ba'da - after), قبل (qabla - before), منذ (munThu - since) or a place حيث (Haythu - where). According to their semantics and function in the sentence, function words are used in sentences expressing an introduction, explanation, consequence, among others. [10]. In Arabic, function words include prepositions, such as "في" (fii - in) or "على" ('alaa - on); coordinating conjunctions, such as "ثم" (thumma - then); adverbs, such as "أبدا" ('abadan - ever) or "بسرعة" (bisor'aah - quickly); and quantifiers, such as "كل" (kulla - all) or "بعض" (ba'Da - a part of).

All the above-mentioned lemmas are listed in "**El-DicAr**", the **El**ectronic **Dic**tionary for **Ar**abic. Automated inflectional and derivational routines are applied to this list producing over 3 million inflected forms. "**El-DicAr**",associates each lexical entry with a set of relevant information and identifies the inflectional paradigms that allows the automated generation of all inflected forms. The linguistic information represents lemma, grammatical category, inflectional information (gender, number, time, mood, etc.), syntactic information (e.g., transitivity: +Tr) and the distributional and semantic information (e.g. Human, Concrete, Country, Sea, etc.).

3 Morphological Analysis and Automatic Tokenization

3.1 Agglutinative Structure of Arabic Tokens

Most Arabic tokens have a complex structure, in which case it is designated as "maximal word form". This designation was given by D. Cohen [6] to a word form that can correspond to a succession of one or more proclitics, a radical and one or more enclitics. Radicals, designated as "minimal word form", themselves are forms which have been inflected or derived from a lemma. An Arabic token can correspond to a whole French sentence. For instance, the agglutinated form "أَسَتَتَذَكَّرُونَنَا" ('asatataThakkaroonanaa → 'a + sa + tataThakkaroona + naa) can be translated into "will you remember us?". In this section, we describe the two types of clitics (proclitics and enclitics) which can be added to a "minimal word form to produce an agglutinative form [4].

3.1.1 Proclitics and Enclitics

In Arabic grammar, we can enumerate 10 proclitics and 13 enclitics that can be, respectively, added to a "minimal word form" as a prefix or a suffix in order to make a "maximal word form". Whereas a compound word form can have only one enclitic, it can contain more than one proclitic. When combined together, proclitics can give a special syntactic meaning (coordination, emphasis...).

Proclitics can be combined together and form a compound proclitic. A robust analysis should identify the syntactic category of each of these components of such

compound forms. This will be particularly useful and even essential for parsing. Proclitics are classified into four categories depending on the potential position within a compound proclitic. The combination of these proclitics is governed by two types of constraints:

- **Order relationship:** Each proclitic is inconsistent, in a strict order, with a proclitic in the same position [8] Similarly, a proclitic, which occupies a position of precedence over another one on the classification above, has no chance to follow it in a "maximal word form".
- **Compatibility rules:** Proclitics respecting the order relationship are not necessarily compatible with each other, for syntactic and semantic reasons [7]. In this regard, we note that the combination of proclitic belonging to four different positions is uncommon. For example, a construction such as "أفبالبيت" decomposable as "أ + ف + بـ + الـ + بَيْتِ" ('a + fa + bi + l + bayti – and + is it + in + the + house ?) is rarely used in texts.

In addition to these compatibility rules between proclitics, other rules are needed to verify the compatibility of proclitics with the rest of word form components (radical and enclitics). A study of the combination of morphemes of agglutinated form words has been undertaken by K. Beesley [3] and S. Mesfar [15].

3.1.2 Lexical Analysis and Agglutination

The complex structure of the Arabic word, described above, derives from the inflectional and agglutinative phenomena that characterize the language. These phenomena cause serious problems for the automatic analysis of Arabic; they increase the rate of ambiguity by introducing additional ambiguities in the segmentation of words. Indeed, an Arabic word can have several possible analyses into: proclitic(s), inflected form and enclitic.

For example, the word form "أوحل" may have three potential analyses:

- **1st analysis :**

1st proclitic	2nd proclitic	Inflected form
أ ('a) – question mark	و (wa) – coordinating conjunction	حَلَّ : preterit verb (Halla – resolve, happen, occur or settle) or a noun (Hall – a solution)

- **2nd analysis :**

Proclitic	Inflected form
أ ('a) – question mark	وَحَلَ : preterit verb (waHal – to make it muddy), وَحْل : a noun (waHl – mud)

- **3rd analysis :**

Inflected form
أوحل : preterit verb ('awHala – throw in the mud)

Given the abundance of such ambiguous word forms, it is essential to build a tokenization system to be able to identify, parse their different morphemes and deal with the complexity of their potential agglutinative structure.

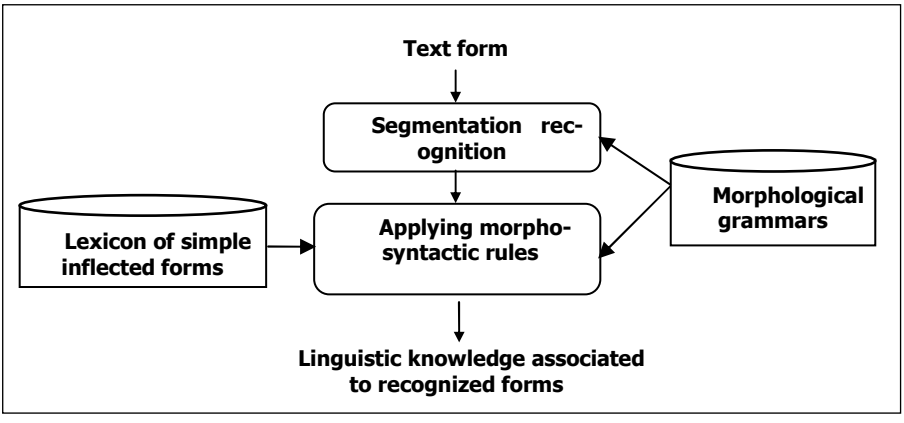


Fig. 1. Chain of a text form morphological analysis

3.2 Morphological Analysis and Grammatical Rules Definition

The Arabic language is a strongly agglutinant language; its morphological analyzer should separate and identify the component morphemes of the input word, labeling them somehow with sufficient information to be useful for the tasks at hand.

We start our analysis with the application of a decomposition system, implemented via a NooJ morphological grammar, to each word of the text to identify its radical and affixes. In the second step, grammars (finite-state transducers) produce lexical constraints checking the validity of segmentation thanks to a dictionary lookup. So, these grammars associate the recognition of a word to lexical constraints, working only with valid combinations of the various components of the form. Typically there are several output strings, each representing a possible analysis of the input word.

We continue the description of the morphological analysis of Arabic within the linguistic platform NooJ by detailing the different lexical constraints implemented inside morphological grammars.

- *Morphological constraints:* they consider morphological incompatibilities which would have to be generated from a direct decomposition. We proceed by the application of some morphological transformations (addition of affixes, deletion, substitution, etc.) which can be combined together to deal with more complex morphological phenomena.
- *Constraints on the syntactic properties of verbs:* they verify the mark "+Transitive" of verbs in the dictionary. Indeed, the transitivity of a verb is directly related to the possibility of its suffixation. Such agglutination will be only permitted for direct transitive verbs and indirect transitive ones conjugated at the singular third person [1].
- *Orthographical constraints:* they look at letters which have orthographical variation during agglutination. We can cite the case of the letter "T" which can be written in two different orthographies with the same pronunciation and also the hamza with six different orthographies ("ء", "أ", "إ", "ؤ", "ئ", "ـئـ") for the same pronunciation.

- *Phonological constraints:* These constraints, generally combined with morphological ones, maintain a consonance inside agglutinated forms. They deal with the compatibility of the declension of radical and attached suffix.

4 Automatic Morpho-syntactic Analysis

4.1 Automatic Vocalization

The formalized inflection of verbs, primitive nouns and deverbals allows recognition of all the corresponding inflected terms; the lookup algorithm of NooJ uses finite-state machines, which make possible simultaneous recognition (i.e. without any additional computing) both of vowelled, partially vowelled or unvowelled forms.

In fact, the omission of diacritics in a written form can lead to numerous distinct fully vowelled words. For example, the unvowelled form "ktb" is supposed to have multiple vocalized annotations, our lookup algorithm based on finite state machines is able to return, at the same time, fifteen fully vowelled forms including nouns and verbs (in the active, passive and imperative form).

Moreover, each recognized form is associated by the lookup algorithm of NooJ a set of linguistic information: lemma, grammatical category, gender and number, syntactic information (e.g. +Transitive) and distributional information (e.g. +Human).

Furthermore, the proposed new algorithm to look through finite-state transducers has affected the NooJ's lexical engine in parsing efficiently other Semitic languages, such as Hebrew.

4.2 Lexical Ambiguity Reduction

The ambiguity is one of the central problems of a morpho-syntactic analysis especially for Arabic where the analyzers are frequently faced with situations of ambiguity at different levels:

- the lexical level: the ambiguity is related to the segmentation into lexical units and the homography;
- the syntactic level: the ambiguity is more related to the richness and syntactic constructs and their multiple interpretations;
- the semantic level: the ambiguity is related to the ability to match more than one meaning to a form.

These problems are causing ambiguities, mainly by morpho-syntactic phenomena specific to the Arabic language such as vowels, flexion and derivation and agglutination. In this paper, we focus on lexical ambiguity related to the automatic vocalization and we will only concentrate on reduction of the ambiguity of some verbal forms. In fact, we distinguished different kinds of contextual information that can be used for disambiguation depending on the preceding form which can be:

- A relative pronoun: in this case, we reduce the ambiguity over 73.3% of cases;
- A subjunctive particle: in this case, we reduce the ambiguity over 70% of cases;
- A jussive particle: in this case, we reduce the ambiguity over 80% of cases.

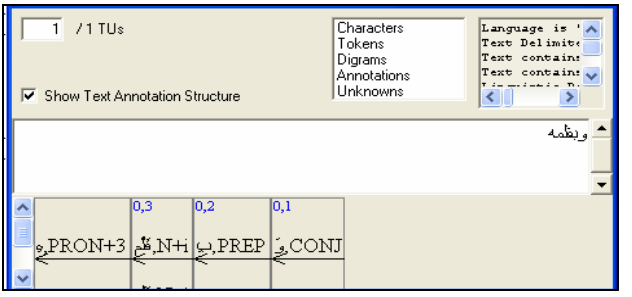


Fig. 2. NooJ's Text Annotation Structure

4.3 Token's Annotation Structure

NooJ's linguistic engine uses an annotation system. An annotation is a pair (position, information) that states that a certain sequence of the text has certain properties. When NooJ processes a text, it produces a set of annotations, stored in the Text Annotation Structure (TAS); annotations are always kept synchronized with the original text file, which is never modified [18].

For instance, we consider the agglutinated form «وَبِقَلَمِهِ» (wabiqalamihi – and with his pen). Its morphological analysis leads to the following TAS which shows that the mentioned agglutinated form can be segmented as a succession of:

- A coordinating conjunction: "وَ" (wa – and) annotated as CONJ;
- A preposition: "بِ" (bi – with) annotated as PREP;
- A nominal form: "قَلَم" (qalami – pen) annotated as N+i (noun, genitive form);
- A personal pronoun: "هِ" (hi – his) annotated as PRON+3 (pronoun, 3rd person);

4.4 Spelling Checker and Corrector

Given the problems caused by spelling errors as part of the robustness of the automatic processing task, we have implemented finite state cascade transducers to deal with some frequent spelling errors such as:

- Letter confusion at the beginning of a word form :"ا" (alif) vs. "أ" (hamza) ;
- Letter confusion at the end of a word form : "ي" (ya') vs. "ى" (alif maqSoorah) or "ة" (t - taa' marbooTah) vs. "ه" (h - haa') ;
- Letter inversion : "ـًا" vs. "ـاً" ;

5 Automatic Arabic Named Entity Recognition

As a first step of the processing, we used the lexical module of the linguistic platform NooJ for vocabulary formalization and tokenization. Then, we evaluated the lexical coverage of our Arabic module on LASELDI's[4] corpora described in Section 6.1. Using our lexical and morphological resources, the lexical analysis of these corpora

[4] LASELDI: LAboratoire de SEmio-Linguistique, Didactique et Informatique, University of Franche-Comté, Besançon, France.

shows about 92% coverage. The unrecognized forms include 85% of transliterated named entities, about 11% of borrowing terms such as "ميتافيزيقا" (miitaafiiziiqaa - metaphysics) as well as some spelling mistakes.

This analysis showed that the majority of unrecognized forms are proper names (names of people, organizations or localities). Although these unrecognized forms are words or sequences of words called named entities (NEs) which cannot be found in common dictionaries, they encapsulate important information that can be useful for the semantic interpretation of texts.

Since so far there are no defined standards for writing or transliterating proper names[5], the simple lookup approach was impossible to adopt. In fact, it is impossible to enumerate all proper names in lists, to collect and maintain these lists, to deal with name variants and finally to resolve the resulting ambiguity. So, we built a named entities recognition system based on the syntactic module of NooJ and using its syntactic grammars. These used local grammars represent predefined rules based on internal and external [13] evidence in named entity recognition where:

- *Internal evidence:* is taken from within the sequence of words that includes the name, such as the content of lists of proper names (gazetteers).
- *External evidence:* is provided by the context in which a name appears.

The adequacy of this solution was retained within the last MUC conference. It will be developed for Arabic Named Entity Recognition within the developmental environment NooJ, the tool used for identifying and categorizing Arabic NEs.

5.1 Problems with Arabic Named Entity Recognition

In addition to lack of obvious clues such as initial capitalized letters to indicate the presence of a proper name, there are some specific problems related to Arabic named entity recognition.

- *Non-vocalization[6]:* Non-vocalization can affect a named entity recognition system when potential vocalizations can lead to different senses which can designate trigger words for two or more different NE type such as the case of unvowelled form "مؤسسة" (mo'ass'sah) that can be considered as trigger word for an organization name ("مُؤَسَّسَة" [mo'assasah – a company]) as well as trigger word for a person name ("مُؤَسِّسَة" [mo'assisah – a founder, fem.]).
- *Delimitation problems:* Delimitation problems are related to a lack of information about unknown words within NEs, an antonomastic usage where proper names are substituted with a phrase or conversely as well as the presence of some homonyms[7] which increases ambiguity when trying to mark NE constituents such as "أَشْرَف" ('ashrafa) which can be a first name, an inflected verbal form meaning "he supervised", an elative adjective which means "the most honorable", etc.

[5] To a large extent, transcription systems depend on the origins of their authors, their individual writing methods, their educational background or the writing norms of their native countries.

[6] Non-vocalization means the absence of short vowels in common Arabic texts. It leads to a high degree of ambiguity. In theory, only the Koran, and children's books are fully vowelled.

[7] A homonym is a word that has the same pronunciation and/or spelling as another word, but a different meaning.

5.2 Automatic Named Entity Recognition Process

Our Arabic Named Entity Recognition system is a two-step process. Initially, we try to collect the maximum of information for contextual recognized forms. Then, this information will be used within syntactic grammars to locate relevant sequences.

- *Morphological analyzer:* the morphological analyzer, described in the Section 3, splits the agglutinated form to identify the attached affixes (conjunctions, prepositions, personal pronouns, etc.). If there was no such tokenization functionality, then these affixes will not be recognized anywhere. This has the effect that agglutinated lexical markers will not be recognized since the system no longer correctly tokenizes these forms and associates them with useful linguistic information.
- *Named Entity Recognizer:* The Named Entity Recognition system within NooJ is based on the use of some knowledge sources:

  - Gazetteers: They are lexical marker lists, containing names that are identified and listed beforehand and have been classified into named-entity types. Lists of names are employed for locations, personal titles, organizations, dates/times and currencies. We also use lists of trigger words which indicate that the surrounding tokens are probably named entity constituents and may reliably allow determination of the type or even the subtype of the named entity (e.g. religious and political person names are considered as subtypes of the person names category). These lists of triggers were produced manually, tagged as result of the morphological analysis and used in NER grammar rules.

  - Grammars: They are compiled into Finite-State transducers, Context-Free grammars (stack automata) and ERTNs (enhanced Recursive Transition Networks). A syntactic grammar[8] represents word sequences described by manually created rules, and then produces some kind of linguistic information such as type of the recognized NE.

A grammar rule is generally made of, at least, a trigger word, some tagged words and occasionally unknown words in order to group together elements pertaining to the same entity. Sequences of words can be accurately tagged given an appropriate context especially if a trigger word or an entry from gazetteers disambiguates the sequence.

The preponderance of unknown words within NEs induces a lack of information; added to problems of determination of stop words that allow knowing where to stop, which increases boundary errors. NooJ syntactic grammars respect some heuristics when applying rules. They locate the "longest match" for one grammar and "all matches" for the whole of grammars.

The extracted named entities are displayed into a concordance window to give users a quick overview of the contents of documents. This is particularly useful when applied to large document collections especially when sorting, identifying and filtering out bad concordances, as well as producing statistics on the contents of the whole corpora. This would, also, allow us to extend our gazetteers and syntactic rules in order to enlarge the set of identified expressions.

[8] We give an example of a syntactic grammar in Fig. 5.

- *Evaluation of NER system:* Traditionally, the scoring report compares the answer file with a carefully annotated file. The system was evaluated in terms of the complementary precision (P) and recall (R) metrics. Briefly, precision evaluates the noise of a system while recall evaluates its coverage. These metrics are often combined using a weighted harmonic called the F-measure (F).

$$\mathbf{P= \#\ of\ correct\ detected\ entities\ /\ \#\ of\ detected\ entities.} \quad (1)$$

$$\mathbf{R= \#\ of\ correct\ detected\ entities\ /\ \#\ of\ entities\ manually\ annotated.} \quad (2)$$

$$\mathbf{F = 2\ P\ R\ /\ (P+R).} \quad (3)$$

The evaluation carried out on parts of our corpora gives the following scores:

Table 1. Experiments on our corpora

		Precision : P	Recall : R	F-mesure : F
TIMEX		97,2%	94,7%	95,9%
NUMEX		97,6%	93,9%	95,7%
ENAMEX	Person names	93,2%	80,4%	86,3%
	Organizations	91,3%	77,5%	83,8%
	Localizations	80,1%	71,7%	75,6%

Despite the problems described above, the used techniques seem to be adequate and display very encouraging recognition rates. According to the results of the developed system, the named entities extraction grammars follow the Zipf law. Indeed, a minority of the rules may be sufficient to cover a large part of the patterns and ensure coverage. However, many other rules must be added to improve the recall.

6 Evaluation

6.1 General Characteristics of the Evaluation Corpus

The evaluation corpus in composed of two sub-corpora:

- A journalistic corpus composed of more than 1,000 journalistic articles discussing general news topics and covering various subjects of politics, economics, culture and sport activities. The corpus includes more than two million word forms.
- A corpus of stories, containing about 200 stories published on the Arabic Writers Union website (cf. http://www.awu-dam.org). This corpus includes romances, narrations and fictional stories. It contains a total of 4,586.439 word forms.

6.2 Evaluation and Lexical Coverage

During the formalization of Arabic language, we used to work, repeatedly, on test and validation tasks. Indeed, before proceeding to a further step, we make some

verification routines on the current step results. In the table below, we show the results of experiments, related to the lexical coverage, obtained after each step:

- **1st step:** Tokenization process and electronic dictionaries application;
- **2nd step:** addition of the spelling error grammars;
- **3rd step:** application of named entity recognition grammars.

Table 2. Lexical coverage evaluation

	Journalistic corpus	**Stories' corpus**
1st step	84.17 %	85.32 %
2nd step	93.73 %	94.14 %
3rd step	95.4 %	95.69 %

After these three evaluation steps, the final list of "Unknows" contains 57% of forms that are components of named entities which remain unrecognized because of insufficiency of contextual or structural information, 21.6% of foreign words written in Arabic alphabet, 17.3% of typographical errors and 4.1% isolated letters.

7 Conclusion

In this paper, we presented a series of morpho-syntactic tools for Arabic natural language processing. In addition to a large coverage electronic dictionary, we described our morphological analyzer that tokenizes agglutinated forms and identifies the component morphemes, our lexical analyzer based on the finite state technology that handles the different vocalization statements in Arabic written texts without any additional computing and a rule-based named entity recognition module that detects proper names in context. Although impressive, the displayed results could be improved. These improvements could settle some problems mainly related to the named entities recognition step. Among these problems, we cite:

- The abundance of the spelling variation for transcribed named entities, which is induced by the absence of conventions for their writing. In fact, transliteration and transcription of foreign names do not obey to special writing rules. Since we observed that there are some spelling variants that are specially related to long vowels transcription, we plan to develop some morphological grammars to automatically deal with alternative spellings.
- The problems of entity delimitation are mainly introduced by the high degree of ambiguity coming from the morpho-lexical analysis step. We propose to perform a new syntactic processing to better "understand" the syntactic structures of sentences and carry out a morpho-syntactic disambiguation before the named entities identification task.

The cascade of morpho-syntactic tools[9], described above, provides a detailed and comprehensive analysis for the different lexical forms, compositions, morphological or syntactic structures. In addition, they are used into a browser-based application, NooJ4Web, that allows an on-line analysis of static as well as dynamic texts [14].

References

1. Achour, H.: Contribution à l'étude du problème de la voyellation automatique de l'arabe. PhD Thesis, Paris7 University (1998)
2. Beesley, K.: Arabic Finite-State Morphological Analysis and Generation. In: Proceedings of the16th International Conference on Computational Linguistics (COLING 1996), Copenhagen, Denmark, pp. 89–94 (1996)
3. Beesley, K.: Arabic Finite-State Morphological Analysis and Generation of Arabic at Xerox Research: Status and Plans in 2001. In: Proceedings of ACL/EACL 2001 Workshop, ARABIC Language Processing: Status and Prospects, Toulouse, France, pp. 3–19 (2001)
4. Bohas, G., Guillaume, J.P., Kouloughli, D.E.: The Arabic Linguistic Tradition. Reprinted version in the Georgetown Classics in Arabic Language and Linguistics of the 1990 edition. Georgetown University Press, Washington (2006)
5. Chomsky, N.: Aspects of the Theory of Syntax. Trad. J.-C. Milner (1975)
6. Cohen, D.: Essai d'une analyse automatique de l'arabe. In: Cohen, D. (ed.) Etudes de linguistique sémitique et arabe, pp. 49–78. Mouton, Paris (1970)
7. Dichy, J.: Pour une lexicomatique de l'arabe: l'unité lexicale simple de l'inventaire du mot. META - Journal de traduction 42(2), 291–306 (1997)
8. Dichy, J.: On lemmatization in Arabic. A formal definition of the Arabic entries of multilingual lexical databases. In: Proceedings of ACL/ EACL 2001, Workshop, ARABIC Language Processing: Status and Prospects, Toulouse, France, pp. 52–65 (2001)
9. Harris, Z.S.: Transformational Theory. Language 41(9), 363–401 (1985)
10. Kadri, Y., Benyamina, A.: Un système d'analyse syntaxico-sémantique du langage arabe non voyellé. Engineer software thesis, Oran University (1992)
11. Khoja, S., Garside, R., Knowles, G.: A tagset for the morpho-syntactic tagging of Arabic. In: Proceedings of the International conference CL 2001, Lancaster, UK, pp. 341–349 (2001)
12. Kouloughli, D.E.: Lexique fondamental de l'arabe standard moderne. l'Harmattan, Paris (1991)
13. Mac Donald, M.: Internal and external evidence in the identification and semantic categorization of proper names. In: Corpus processing for Lexical Acquisition, pp. 21–39. Massachussetts Institute of Technology (1996)
14. Mesfar, S.: NooJ4Web: an on-line concordance service. In: Proceedings of the 10th International NooJ conference, pp. 173–189. Cambridge Scholars Press (2007)
15. Mesfar, S.: An Automatic morpho-syntactic analyzer and a named entities recognition system for standard Arabic. PhD Thesis, Franche-Comté University, France (2008)
16. Revuz, D.: Dictionnaires et lexiques: méthodes et algorithmes. PhD Thesis, Paris7 University, France (2001)
17. Silberztein, M.: NooJ's Dictionaries. In: Proceedings of the 2nd Language and Technology Conference (LTC 2005), Poznan, Poland, pp. 128–133 (2005)
18. Silberztein, M.: An Alternative Approach to Tagging. In: Kedad, Z., Lammari, N., Métais, E., Meziane, F., Rezgui, Y. (eds.) NLDB 2007. LNCS, vol. 4592, pp. 1–11. Springer, Heidelberg (2007) (invited talk)
19. Silberztein, M.: NooJ Manual (2010), `http://www.nooj4nlp.net`

[9] These tools are available for free download to be used by linguists, computer scientists, etc.

An Open-Source Computational Grammar for Romanian

Ramona Enache, Aarne Ranta, and Krasimir Angelov

Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
enache@student.chalmers.se, aarne@chalmers.se, krasimir@chalmers.se

Abstract. We describe the implementation of a computational grammar for Romanian as a resource grammar in the GF project (Grammatical Framework). Resource grammars are the basic constituents of the GF library. They consist of morphological and syntactical modules which implement a common abstract syntax, also describing the basic features of a language. The present paper explores the main features of the Romanian grammar, along with the way they fit into the framework that GF provides. We also compare the implementation for Romanian with related resource grammars that exist already in the library. The current resource grammar allows generation and parsing of natural language and can be used in multilingual translations and other GF-related applications. Covering a wide range of specific morphological and syntactical features of the Romanian language, this GF resource grammar is the most comprehensive open-source grammar existing so far for Romanian.

1 Preliminaries

GF[1] [1] is a grammar formalism, which uses type theory to express the semantics of natural languages, for multilingual grammar applications. The GF resource grammars [2] are the basic constituents of the GF library, on top of which applications are built. Notable applications that use GF are the verification tool KeY, for the generation of natural language from the formal language OCL, the dialogue system research project TALK and the educational project WebALT, for generating natural language for mathematical exercises in different languages, and performing multilingual translations.

The two main operations that are regularly performed with resource grammars are the generation of natural language, based on a term in the abstract syntax (linearization) and parsing. Multilingual translation is achieved as a combination of these two processes.

A GF resource grammar basically consists of the abstract syntax, which is a set of rules common to all grammars, and provides the structure of the grammar, and the concrete syntax, which implements the elements of the abstract syntax in the given language, considering its specific features. The abstract syntax provides consistency for the resource library, also ensuring grammatically correct

[1] http://www.grammaticalframework.org

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 163–174, 2010.

multilingual translations. Resource grammars are general-purpose, as they capture the basic traits of the language. Domain-specific applications use a more restricted domain ontology. In this case, there is more emphasis on the semantical aspect, than in the case of general-purpose grammars. In both cases, only syntactically correct constructions can be generated and parsed.

So far the resource library contains 15 languages : English, French, Italian, Spanish, Catalan, Swedish, Norwegian, Danish, Finnish, Russian, Bulgarian, German, Interlingua (an artificial language), Polish and Romanian. The last two languages were added in 2009. Considering the Romance languages (French, Italian, Spanish and Catalan) and the Scandinavian ones (Swedish, Norwegian and Danish), as the languages from the same family shared many similarities, they were each implemented as families in the resource library. In this way, in the Romance and Scandinavian module, all the similar features are grouped together, along with an interface that declares the differences among the languages. Regarding syntactical features, members of the same family share more than 75% of the code, through the implementation of the family module.

Although Romanian is a member of the Romance family, it was implemented independently, due to significant differences between it and the existing Romance languages in the GF resource library.

2 Main Categories

Each resource grammar features a complete set of paradigms for the inflectional morphology of the main categories, namely nouns, adjectives, verbs, numerals and pronouns.

In the abstract syntax, lexical entries are represented as nullary functions (constants). They are given linearizations in the concrete syntax, typically of tables with all the inflection forms. For example: `fun airplane_N : N` from the abstract syntax is linearized in the Romanian resource grammar as `lin airplane_N = mkN "avion"` where the function `mkN` generates all the 12 flexion forms needed for a noun, as well as its the gender.

Special categories are the relational nouns, adjectives and verbs, where we specify the case of the object, and the preposition that binds it with the relational category. For example: `fun forget_V2 : V2` will be linearized as `forget_V2 = dirV2 (v_besch18 "uita")` where `v_besch18` indicates the group of conjugation for the given verb, according to [3]; the name is a reference to Bescherelle, which is the resource used for implementing verb conjugations for most languages in the Romance family. The function `dirV2` indicates that the verb is transitive, and the corresponding object will in the Accusative cases, with no binding preposition (direct object).

2.1 Nouns

Romanian nouns (N) inflect in case, number and species (definite or indefinite form). The definite article is enclitical, while the indefinite article is proclitical.

In the other Romance languages, both the definite and indefinite articles are proclitical. For example:

om → *om***ul** → **un** *om*
man → the man → a man

There are 5 cases: Nominative, Accusative, Dative, Genitive and Vocative, but due to syncretism between Nominative-Accusative and Genitive-Dative, nouns have at most 3 different forms for case inflexion. Other Romance languages have just one inflectional case; case distinctions are expressed by prepositions. For example in French *de* is used for Genitive, and *a* for Dative.

While the other Romance languages have two genders, Romanian has three: Masculine, Feminine, and additionally Neuter. However, the Romanian Neuter has been the center of some linguistic disputes, as it behaves like Masculine for Singular and as Feminine for Plural, from the agreement point of view. This feature allows us to consider only the basic two genders in the syntactic part of the grammar, when reasoning about agreement between nouns and adjectives and noun phrases and verbs.

Another distinguishing feature of Romanian is the Animacy feature, which plays an important role in syntax, particularly for clitic doubling. Inanimate nouns do not have a special form for Vocative. However, compared to the gender which is inherent, animacy can be changed according to use, most frequently from Inanimate to Animate. The features of nouns also apply to adjectives.

In the Romanian resource grammar, the noun is represented as

```
N = {s: Number => Species => ACase => Str;
     g: NGender; a: Animacy};
```

where

```
NGender = NMasc | NFem | Nneut;
Species = Def | Indef;
ACase = ANomAcc | AGenDat | Avoc;
Animacy = Animate | Inanimate;
```

The syntax of parameters in GF follows the model of declaring an algebraic datatype in functional languages, where the elements of the disjunction are constructors of the type. The representation of the noun is a record with three fields, where the `s` field is a multidimensional table storing the 12 forms of the noun. Each of the parameters separated by `=>` defines a new dimension of the table. A function would, hence, need 12 strings, along with a gender and an animacy attribute for a complete representation of a noun.

However, we provide special functions, named smart paradigms, that build the complete representation using at most 3 parameters. These functions can infer the animacy attribute, gender, and declension forms of a noun. The most common ones are the functions that use the Singular and Plural Nominative Indefinite forms of a noun, but other combinations of forms are also considered.

Since for the Vocative case there are no well-established rules, we provide a function that sets this field to a particular value, in case it cannot be inferred

by the default rules. Regarding the Animacy feature, since by default nouns are assumed to be inanimate, we provide a function for this case, too.

The gender can be automatically inferred from the last letter of the word with a precision of 77%, on the 186 nouns in the GF Lexicon. The great majority of nouns ending in *-ă*,*-e* or *-a* for the singular form have feminine gender[4]. It is considerably harder to distinguish between masculine and neuter nouns if we have just the singular form, but [4], offers some patterns that characterize masculine words, which are statistically rarer than the neuter ones.

In case the smart paradigm takes both the singular and the plural form as arguments, it can normally differentiate masculine from neuter, as the plural form of neuter nouns ends in *-e* or *-uri*, while masculine nouns always have plural forms ending with *-i*.

For the implementation of a Noun Phrase there are more specific details of the language to take into account.

```
NP= {s: NCase => {comp: Str; clit: Clitics => Str};
     a: Agr; indForm: Str; nForm: NForm; isPronoun: Bool};
```

where `NForm = HasClit | HasRef Bool`.

Because pronouns do not have case syncretism, the 5 cases need to be represented separately (`NCase` parameter). The agreement consists of number, gender (Feminine or Masculine) and person.

The parameter `NForm` indicates whether the noun phrase is in referential form and develops clitic doubling (`HasClit`), or, in the absence of clitic doubling, if it is in referential form or not (`HasRef True` or `HasRef False`). We mention that clitic doubling implies referential form, while the reverse does not hold. Nouns in referential form need to be preceded by the preposition *pe* when they act as Direct Objects in Accusative. Although the use of noun phrases in referential form and clitic doubling is very context-dependent in some cases, and subject to discussions in others, we chose the approach suggested in [5]. So, for referential form and clitic doubling, we considered pronouns, animate proper nouns and animate nouns determined by adjectives or possessive pronouns.

Regarding fields from the representation of `NP`:

- `nForm` indicates if the noun phrase needs to be doubled by a clitic in the situations when this phenomenon occurs.
- `isPronoun` is relevant for the clitic doubling situations, because the basic form of the pronoun will be ignored, and the noun phrase will just be represented by the clitic. Situations where clitic doubling also occurs for pronouns are possible, but less common. They are not handled in the resource grammar, since they are meaning dependent, and their semantical role is to emphasize the pronoun.
- `indForm` is used to cover another distinguishing feature in Romanian, which is the usage of the definite/indefinite form of the noun depending on context [5]. Some Accusative prepositions(like *la* - to, *de pe* - on/from) require the indefinite form of a noun phrase, in case it consists of a noun, which is not followed by an adjective or a determiner. For dealing with this case `indForm`

stores the suitable form of the noun, to be used if preceded by such a preposition. For example:

de pe deal - on the hill (Definite form conflict because of preposition *de pe*)
de pe un deal - on a hill (Indefinite form for noun)
*de pe deal***ul** *mare* - on the big hill (Definite form for noun + adjective)

The intermediate category between nouns (N) and noun phrases (NP), sometimes called "N bar", is in the GF resource grammar library called common noun, CN. It consists of a noun, possibly with adjectives, adverbs, relative clauses, appositional attributes, and complements for relational nouns.

Noun phrases can be further formed from common nouns followed by determiners, which give the number of the noun phrase and also select the definite/indefinite form of the common noun.

Proper nouns (PN) have different inflection forms and behavior towards clitic doubling, depending on their animacy. The representation of proper nouns is

```
PN = {s: NCase => Str ; g: Gender ; n: Number; a: Animacy};
```

Proper nouns thus inflect for case and have inherent gender, number, and animacy. Smart paradigms for proper nouns can infer these properties, setting animacy to the animate by default.

2.2 Adjectives

Adjectives (A) are represented in the resource grammar as

```
A = {s: AForm => Str};
```

where `AForm = AF Gender Number Species ACase | AA`.

Considering the agreement of adjectives with nouns, we just consider the two genders Feminine and Masculine. For neuter nouns, we choose the Masculine or Feminine form, depending on the number, on syntactical level. The constructor `AF` builds the representation table of an adjective, consisting of 24 forms, while `AA` maps an adjective to its corresponding adverb, which in most cases has the same form as the adjective for Masculine Singular. The complete representation of an adjective thus consists of 25 forms, but smart paradigms need at most 5 forms to infer them all. Adverbs (Adv) are inflectionally invariant in Romanian, as in most languages in the resource library.

One of the main difficulties of giving complete inflection rules for adjectives was the presence of phonetical mutations. They are not predictable from the lexical structure, being rather dependent on the etymology and age of the word. Neological words do not usually develop phonetical mutations. Compared to nouns, adjectives need more forms of the word, so it is important to be able to determine the effects of phonetical mutations in a systematic way.

When building the forms for all genders and numbers of the adjective, two main mutations can occur: o→oa (masculine singular → feminine singular and

plural) and e→ea (masculine singular → feminine singular). For example, for o→oa : *frumos*(Masc Sg), *frumoasă*(Fem Sg), *frumoase*(Fem Pl, "beautiful") and for e→ea : *drept*(Masc Sg), *dreaptă*(Fem Sg, "right"). These changes affect the second or third last letter in the stem.

The default behavior of the adjectives does not feature the phonetical mutations, for which special functions are provided. In the given lexicon, 80% of the 54 adjectives have default behavior. Also, for 60% of them, just the Masculine Singular Indefinite Nominative form is needed in order to build the whole representation table, using the provided declension rules.

The degrees of comparison are formed on syntactical level, as they do not change the basic form of the adjective.

2.3 Verbs

The category of verbs (V) is by far the most complex one from the Romanian resource grammar. On morphological level the table of a verb is defined as:

```
VForm = Inf | Indi Temps Number Person | Subjo Number Person
      | Imper Number | Ger | PPast Gender Number Species ACase;
```

where `Temps = Presn | Imperf | PSimple | PPerfect` represent the tenses for Indicative and Subjunctive that cannot be formed analytically on syntactical level. The past participle behaves like an adjective, as in the other Romance languages.

The representation of a verb on morphological level consists of 62 forms:

- Present, Imperfect, Perfect Simple and Past Perfect: 6 forms for each
- Infinitive: 1 form
- Conjunctive: 1 form, corresponding to the third person singular, as the other forms are identical to the present ones, except for the irregular verb *a fi* (to be) which will be treated separately.
- Imperative: 1 form, corresponding to the 2nd person singular, as the 2nd person plural has the same form as for present.
- Past Participle: 24 forms as for ordinary adjectives.
- Gerund: 1 form

The 6 forms required for the first four tenses are motivated by the fact that verbs have different forms for the Cartesian Product of the 3 persons (1, 2 and 3) and 2 numbers (singular and plural).

There are 4 conjugation groups, based on the last 1–2 letters:

1. verbs ending in -*a* (which do not belong to the 2^{nd} group) are in the 1^{st} group (example *a lucra* - to work)
2. verbs ending in *ea* (where *e* and *a* belong to the same syllable and are not preceded by *h*) belong to the 2^{nd} group (example *a părea* - to seem)
3. verbs ending in *e* belong to the 3^{rd} group (example *a zice* - to say)
4. verbs ending in *i* or *î* belong to the 4^{th} group (example *a iubi* - to love, *a hotărî* - to decide)

Each of these groups is divided into 4–14 subgroups, which use different affixes to form tenses and moods. Most of the verbs from the Romanian vocabulary belong to the 1^{st} and 4^{th} group. Most of the irregular verbs belong to the 2^{nd} group, while the verbs from the 3^{rd} group are most likely to develop phonetical mutations.

We have implemented the most complete taxonomy of Romanian verbs freely available so far [3], which consists of 140 groups of verbs, and also built a smart paradigm that distinguishes the most frequent 10 groups.

The behavior of a verb cannot be inferred from its lexical structure, as it depends on its etymology. For example, *a ara* ("to plough") belongs to Group 1, subgroup 6, while *a nara* ("to narrate"), belongs to Group 1, subgroup 1. Although they look very similar, the first is of Latin origin, while the other is a neological word imported from French.

An interesting feature of Romanian is the absence of auxiliary verbs for building composite tenses. Romance languages require the verb "to have" / "to be" for building the past form (French : *j'ai dormi* "I have slept", *je suis parti* "I have left"). In Romanian, some particles are used for this purpose. They are the same for all the verbs, and they cannot be used independently as verbs. For example, for the past form *am, ai, a, am, aţi, au* that originate in the verb "to have" *am, ai, a, avem, aveţi, au*, but the two are not identical.

Before proceeding with the structure of a verb phrase (VP), a discussion on clitics in Romanian is needed. There are 4 types of clitics that can follow a verb:

- Accusative (direct object)
- Dative (indirect object without preposition)
- Accusative (reflexive verbs)
- Dative (reflexive verbs)

At most two clitics in different cases out of the four can occur in a verb phrase. The representation of clitics in the Romanian resource grammar is: `Clitics = Normal | Composite | Short | Imperative` where each of the parameters represents a different instance of the clitic, as follows:

- `Normal`: the form that the clitic takes when following a verb in a tense/mood that is not composed with an auxiliary beginning with a vowel (Example : *Eu* **te** *întreb* "I ask **you**")
- `Short`: the form corresponding to composed tenses and moods(Example : *Eu* **te**-*am intrebat* "I asked **you**")
- `Composite`: the form that the clitic takes when combined with another clitic, following a verb in a tense that is not composed (Example: **Ţi-l** prezint "I present **him to you**")
- `Imperative`: the form of the clitic that is used for the Imperative form of the verb. (Example: *Intreaba*-**ma! "Ask me!"**)

As shown in [6], the order of the clitics is always Dative–Reflexive–Accusative, and the Reflexive clitic, when present, acts as a Dative or Accusative, according to its case. The `Imperative` clitics are always placed after the verb in Imperative

mood, while in the other cases they are placed before the verb in the given order, with the only exception of the `Short` clitic for 3rd Person Singular Feminine, which always occurs after the verb.

When combining two clitics for a non-composite tense/mood, both of them are used in their `Composite` form. For a composite tense, the first one is used with the `Composite` form, while the other one is used with `Short` form. If the second clitic is the 3^{rd} Person Singular Feminine, then the first clitic is used with the `Short` form also. In the current implementation, an extra field is used in order to count the number of clitics in a verb phrase. However, in case that two clitics occur, we need to know the case of the reflexive clitic, in order to use it with the right form for a composite tense/mood.

For efficiency reasons, the clitics are stored in the structure of the noun phrases and are transferred to verb phrases in the complementation process.

Having these preliminaries, we can proceed with the representation of the verb phrase. It results from combining transitive verbs with complements, or from intransitive verbs directly.

```
VP = {s: VForm => Str; isRefl: Agr => RAgr;
      nrClit: VClit; pReflClit: Clitics; isFemSg: Bool;
      neg: Polarity => Str; clAcc: RAgr; clDat: RAgr;
      comp : Agr => Str; ext : Polarity => Str};
```

where

- `s` stores the forms of the verb which were built on the morphological level.
- `isRefl` corresponds to the reflexive clitics of the verb phrase.
- `nrClit` counts the number of clitics, while `pReflClit` keeps track of the proper form of the reflexive clitic, when combined with another clitic for a composite tense/mood.
- `clAcc` and `clDat` store the clitics for the Accusative and Dative case.
- `isFemSg` keeps track of whether the verb phrase has an 3^{rd} Person Singular Feminine Accusative clitic.
- `neg` is used to express the polarity.
- `comp` stores the objects of the verb phrase, while the `ext` field stores the secondary phrases, introduced by the verb phrase.

The current implementation of clitics uses the above-mentioned structures and parameters for efficiency reasons, and may look artificial, but the problem of clitics is complex in any language, and requires solutions that are both expressive and efficient.

On syntactical level, a distinguishing feature of Romanian is the lack of infinitives, and the use of verbs in Subjunctive Present instead. For this, case agreement is needed, and the current implementation makes the agreement between the verb and the subject of the phrase or the direct object, depending on the grammatical context. For example:

Eu *vreau să merg* "**I** want to go"
Eu **o** *rog să cumpere* "I ask **her** to buy"

In the first case, the verb agrees with the subject, while in the second case, it agrees with the object.

2.4 Numerals

Numerals in Romanian follow the decimal system, as all the other Romance languages. The cardinals composed with the digits 1 or 2 have different forms for Masculine and Feminine. The ordinals inflect in gender and case, but do not normally have forms for plural. For numerals between 11 and 19, alternative formal and informal forms exist, and the grammar generates both of them. The formal form is however used as default.

A distinguishing feature of numerals is the taxonomy of size: `Size = sg | less20 | pl` . There is a difference between numerals from 2 to 19 and numerals which are greater than 20 on syntactical level: an extra preposition is added when combining a numeral greater than 20 with a noun phrase. For example:

zece oameni "ten people"
treizeci **de** *oameni* "thirty people"

2.5 Sentences

Regarding the formation of clauses and sentences, Romanian is very similar to the other Romance languages. Its structure is SVO where the predicate agrees with the subject in number and person (gender for passive voice or predicates formed by copula + adjective).

The inverse topicalization VOS is used for interrogative sentences introduced by an interrogative pronoun, and for relative clauses. For example:

Ion vede pe cineva "John sees somebody"
Pe cine vede Ion ? "Who does John see?"
Casa pe care o vede Ion "The house that John sees".

The interrogative pronoun *cine*(who), as it is animated, requires the preposition *pe*, when it acts as a direct object, but it does not develop clitic doubling. On the other hand, the noun *casa* ("house"), although not animated, is doubled by the corresponding clitic (*o*) in the relative clause that determines it.

3 Evaluation

The Romanian resource grammar was added to the GF library in September 2009, after almost 4 months of work. It consists of 20 modules that cover morphological and syntactical features of Romanian, which are written in the GF language. The size of the code is 5892 lines, which is above the average of the GF library [2].

Resource grammars can be embedded in programming languages like Haskell and Java. This is achieved by compiling the resource grammar to PGF [7], a

portable grammar format, which will imported and processed by the host language. The PGF form of a grammar is also a measure of its complexity, as it reflects the number of rules that the grammar uses and the way they combine. For example, the first implementation of clitics in Romanian, which was the intuitive approach, that just kept the clitics and a boolean parameter, keeping track of whether a given clitic is present or not, made the resource grammar was so complex, that the PGF file could not be generated by the GF compiler, as the number of rules was too big. The current implementation of clitics reduced the number of rules 200 times for the verb category, and 4369 times for the complementation function. This made possible the generation of a PGF file for Romanian, which can be used for parsing, multilingual translations and other related applications. Because of the complexity of the morphology and of the clitics, the Romanian resource grammar has one of the highest number of rules in the library.

There is a trade-off between the expressive power of a resource grammar and its efficiency. Our approach covers, as we showed, many specific features of Romanian, but there are still constructions that the current grammar does not cover. One of them is the presence of clitic doubling in a relative phrase that contains a nested verb phrase sequence. For example: *Maşina pe care mă roagă ei să* **o** *cumpăr* ("The car that they beg me to buy"), where the clitic refers to *maşina* ("the car"), is generated as *maşina pe care mă roagă ei să cumpăr*. This can be understood to have the same meaning, but it is not correct in standard Romanian. The solution to this problem would require an additional field for verb phrases. This would bring about an increase in the number of rules for verbs and functions that involve verbs that would make it impossible, in the current implementation of PGF, to generate the PGF format file for Romanian. We prefered to make the grammar as expressive as possible, in the current context of the GF compiler and resources, but still keeping it reasonably efficient, so that it can be used in GF-related applications.

4 Future Work

An obvious direction for future work is improving the efficiency of the grammar, making it possible to add features that are not currently covered because of complexity issues.

A big step towards a more expressive grammar would be adding a bigger lexicon, perhaps by import from other open source projects.

Another main direction is to derive an application grammar for Romanian for the projects that use GF, like WebALT, TALK or KeY.

5 Related Projects

The number of open-source projects that attempt to give a formal characterization of the Romanian language is relatively small, and they deal mostly with the morphological features of the language.

Roric-Ling[2], describes paradigms for inflectional morphology of nouns, adjectives and verbs. The rules cover a small lexicon (almost 100 entries), but there are many other cases of inflection which are not treated. For verb conjugation, around 30 forms of the verb are needed as input. Our approach has a wider coverage of the morphology, also featuring smart paradigms, which require considerably smaller inputs.

Another significant project that deals with Romanian morphology is the spell checker from Open Office[3]. It features a comparable set of rules for inflection of noun and adjectives, and a large database.

The EGLU project [8] features the most comprehensive implementation of the Romanian morphology, a large database, but it does not have such a wide coverage for the syntax part, and, to our knowledge, no treatment of clitics. It also has the possibility of performing automated POS tagging and morphological analysis.

Liviu Ciortuz described and implemented a HPSG kernel for Romanian in his PhD thesis [9], elaborating on NPs, VPs and some aspects about clitics. Our work features more aspects of the grammar and is a part of a large multilingual framework.

The LinGO Matrix [10] and Pargram [11] projects, are similar to the GF project and they both feature a computational grammar for Romanian, but they are still under construction, and were not available for a more detailed comparison.

Regarding the theoretical study of Romanian clitics, we mention the work of P. Monachesi [12].

Other computational linguistic resources for Romanian are related to the areas like machine-learning, aquisition of corpora, POS taggers and lemmatisers, word sense disambiguators and others [13].

6 Conclusions

The current resource grammar integrates Romanian in the GF setting, expressing the main features of the language. However it is not complete, as it cannot parse arbitrary sentences, or generate all the possible constructions. The morphology is complete, in the sense that it covers the main categories and their possible declensions/conjugations, and can always be applied to a bigger lexicon, or used in application grammars for any new domain.

Romanian was not integrated in the Romance module, because of some significant differences from the other languages in the family. Some of these are the enclitical definite article, different forms of nouns and adjectives for case triggered declension, and the animacy hierarchy for nouns. Another key feature of the grammar is the problem of clitics and clitic doubling, which is considerably different from the languages that were already present in the GF resource library.

[2] http://phobos.cs.unibuc.ro/roric/morpho/demo.html
[3] http://extensions.services.openoffice.org/node/1392

The Romanian resource grammar in GF provides substantial coverage of both morphological and syntactical aspects of the language, and is so far the most comprehensive computational grammar for Romanian.

References

1. Ranta, A.: Grammatical Framework: A Type-Theoretical Grammar Formalism. The Journal of Functional Programming 14(2), 145–189 (2004)
2. Ranta, A.: The GF Resource Grammar Library. Linguistic Issues in Language Technology 2 (2009)
3. Barbu, A.M.: Conjugarea Verbelor Româneşti. Editura Coresi, Bucureşti (2007)
4. Perkowski, J.L., Vrabie, E.: Covert Semantic and Morphophonemic Categories in the Romanian Gender System. Slavic and East European Journal 30 (1986)
5. Chiriacescu, S., von Heusinger, K.: Pe-marking and Referential Persistence in Romanian. In: SinSpeC - Working Papers of the SFB 732, Incremental Specification in Context (2009)
6. Klein, U.: The syntax of Romanian clitic pronouns, University of Bielefeld (2007) (manuscript)
7. Angelov, K., Bringert, B., Ranta, A.: PGF: A Portable Run-Time Format for Type-Theoretical Grammars. Journal of Logic, Language and Information (2009) (to appear)
8. Tufiş, D., Barbu, A.M.: A Reversible and Reusable Morpho-Lexical Description of Romanian. In: Tufiş, D., Andersen, P. (eds.) Recent Advances in Romanian Language Technology, pp. 83–93. Editura Academiei Române, Bucureşti (1997)
9. Ciortuz, L.V.: DF—A Feature Constraint Concurrent System with application to Natural Language Processing. PhD thesis, University of Lille (1996)
10. Bender, E.M., Flickinger, D., Oepen, S.: The grammar matrix: an open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars. In: COLING 2002 on Grammar engineering and evaluation, Morristown, NJ, USA, pp. 1–7. Association for Computational Linguistics (2002)
11. Butt, M., Dyvik, H., King, T.H., Masuichi, H., Rohrer, C.: The parallel grammar project. In: COLING 2002 on Grammar engineering and evaluation, Morristown, NJ, USA, pp. 1–7. Association for Computational Linguistics (2002)
12. Monachesi, P.: Clitic placement in the romanian verbal complex. In: Clitics in Phonology, Morphology and Syntax. John Benjamins, Amsterdam (2000)
13. Cristea, D., Forascu, C.: Linguistic resources and technologies for romanian language. The Computer Science Journal of Moldova 14(1), 34–73 (2006)

Chinese Event Descriptive Clause Splitting with Structured SVMs

Junsheng Zhou[1,2], Yabing Zhang[3], Xinyu Dai[3], and Jiajun Chen[3]

[1] Department of Computer Science and technology, Nanjing Normal University, Nanjing, China
[2] Jiangsu Research Center of Information Security & Confidential Engineering, Nanjing, China
[3] State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
{Zhoujs,Zhangyb,Dxy,Chenjj}@nlp.nju.edu.cn

Abstract. Chinese event descriptive clause splitting is a novel task in Chinese information processing. Different from English clause splitting problem, Chinese event descriptive clause splitting aims at recognizing the high-level clauses. In this paper, we present a Chinese clause splitting system with a discriminative approach. By formulating the Chinese clause splitting task as a sequence labeling problem, we apply the structured SVMs model to Chinese clause splitting. Compared with other two baseline systems, our approach gives much better performance.

Keywords: Chinese clause splitting, Partial parsing, Structured SVMs.

1 Introduction

Chinese event descriptive clause splitting is the task of splitting a complex Chinese sentence into several clauses [1], which is a novel task in Chinese information processing. This task is important for various tasks such as syntactic parsing, machine translation, aligning parallel text and transformation from natural language sentences into logical forms. Chinese event descriptive clause splitting is deeper level of partial parsing, which is the task of recovering only a limited amount of syntactic information.

In English, there is a similar clause splitting problem presented as a shared-task problem in CoNLL-2001 [2]. The goal of English clause splitting problem is to identify embedded clauses in text. Considering the difficulty of English clause splitting, the shared task was divided into three parts: identifying clause starts, recognizing clause ends and finding complete clauses. Many machine learning approaches have been developed for English clause splitting. These methods include boosting decision trees and decision graph, neural networks, memory-based learning, statistical, and symbolic learning [3][4][5]. Carreras applied the Adaboost algorithm and improved clause identification by using global inference on the top of the outcome clauses hierarchically learned by local classifiers [6]. Then, Carreras used a discriminative model for it [7]. They applied a global learning algorithm, FR-Perceptron [8] to recognize the structure of clauses. The FR-Perceptron method shows the best result for English clause splitting now. Other approaches such as Maximum Entropy, and Winnow are

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 175–183, 2010.

applied for clause splitting too [9]. Recently, Nguyen et al. presented a CRFs-based framework approach to clause splitting [10], and achieved a result competitive with the state-of-the-art results of clause splitting.

The problem of Chinese event descriptive clause splitting is similar to the third part in the shared-task problem in CoNLL-2001, and Chinese event descriptive clause splitting aims at recognizing the high-level clauses [1]. However, there is little work to date on Chinese event descriptive clause splitting problem.

We present a discriminative approach to Chinese event descriptive clause splitting problem. We formulate Chinese clause splitting as a sequence tagging problem, and learn a discriminative tagger from labeled data using a structured support vector machine (SVM) [11][12].

2 Chinese Event Descriptive Clause Splitting Problem

The input to the event descriptive clause splitting splitter is a complete Chinese sentence that is correctly segmented and labeled the part-of-speech (POS) tags. Then the event descriptive clause splitting algorithm recognizes the left and right boundaries of every event descriptive clause to form a sequence of event descriptive clauses. Here is an example of a sentence and its event descriptive clauses obtained from Tsinghua treebank:

[只有/c 自身/rNP 硬/a] ，/wP [才/d 能/vM 对/p 不良/a 风气/n 、/wD 腐败/a 现象/n 敢/vM 抓/v 敢/vM 管/v] ，/wP [不/dN 怕/v “/wLB 鬼/n ”/wRB] ，/wP [不/dN 信/v 邪/a] ，/wP [敢/vM 摸/v “/wLB 老虎/n ”/wRB 屁股/n] 。/wE

The brackets “[” and “]” in the sentence specify the left and right boundaries of each event descriptive clause respectively.

For English clause splitting problem, a clause splitter is intended to be used after a POS tagger and a chunk parser. Chunks are sequences of consecutive words in the sentence which form the basic syntactic phrases, subject to the constraints that chunks cannot overlap or have embedded chunks. In a correct syntactic tree, clause boundaries are always at some chunk boundaries. However, in Chinese clause splitting task, Chunk tags are not provided for Chinese clause splitter.

In general, an English clause is leaded by an antecedent, which is obviously a formal mark for clause. In contrast to English clauses, there are not any particular marks between Chinese clauses. In Chinese clause splitting, punctuators are often viewed as the separators between clauses. But the use of punctuator is very flexible in Chinese. For instance, the punctuators can be used for separating the functional chunks such as subject, predicate and object. It can also be applied to separating the conjuncts in a functional chunk. Chinese clause splitting is a difficult task.

3 Structured SVMs

The structured support vector machine (SVM) generalizes the Support Vector Machine classifier that supports binary classification, multiclass classification and regression. the structured SVM allows training of a classifier for general structured output labels. Structured classification is the problem of predicting y from x in the case where y having a meaningful internal structure. Elements $y \in Y$ may be, for instance, sequences, trees, or graphs. The major problem for the structured SVMs is the modification of multiple classifications to the very large number of labels problem. To solve the problem, Tsochantaridis et al. [11] presented a re-scaling method for the SVM optimization problem and viewed it as discriminative classification by employing several loss function and maximization methods. As is typical of discriminative approaches, a feature vector $\Psi(x, y)$ needs to be created to represent a candidate y and its relationship to the input x. In the framework of structural SVMs [11], training the parameters can be formulated as the following optimization problem [12].

$$\begin{aligned} &\min \ \frac{1}{2}\| w \|^2 + C\sum_i \xi_i \\ &\text{s.t.} \quad w \cdot \Psi(x_i, y_i) - w \cdot \psi(x_i, y) \geq L_{i,y} - \xi_i, \ \forall i, y \in Y \setminus y_i; \end{aligned} \tag{1}$$

The training objective is to weight the features using a vector w so that the correct tag sequence receives more weight than the incorrect sequences. The constraints state that the score $w \cdot \Psi(x_i, y_i)$ of the true output y_i must be greater than the score of all alternative alignments y by a difference of $L_{i,y}$. $L_{i,y}$ is a loss function that measures how different the two output y_i and y are. ξ_i is a slack variable shared among constraints from the same example, since in general the problem is not separable.

For most structured problems, the number of constraints in optimization problem (1) is huge and it is unfeasible to solve the quadratic program directly. However, it has been shown that the cutting plane algorithm can be used to efficiently approximate the optimal solution of this type of optimization problem [12]. The cutting plane algorithm starts with an empty set of constraints, adds the most violated constraint among the exponentially many during each iteration, and repeats until the desired precision is reached.

4 Chinese Event Descriptive Clause Splitting with Structured SVMs

Due to the fact that Chinese clauses are often split at the locations of punctuators, we formulate the Chinese event descriptive clause splitting task as a sequence labeling problem.

4.1 Annotation Method

According to the definition of Chinese event descriptive clause, a clause is a sequence of words separated by a punctuator such as comma, semicolon and interrogation. But these punctuators have very flexible usage, as described in section 2, not limiting to the clause separator. By interpreting every sequence of words separated by a punctuator as a block, we can formulate the Chinese event descriptive clause splitting task as a sequence labeling problem where a block is similar to a token to be labeled in POS tagging problem. For example, the input is the following sentence:

只有/c 自身/rNP 硬/a ，/wP 才/d 能/vM 对/p 不良/a 风气/n 、/wD 腐败/a 现象/n 敢/vM 抓/v 敢/vM 管/v ，/wP 不/dN 怕/v “/wLB 鬼/n ”/wRB ，/wP 不/dN 信/v 邪/a ，/wP 敢/vM 摸/v “/wLB 老虎/n ”/wRB 屁股/n 。/wE

By finding specific punctuators occurring in the sentence, not including the double quotation marks, the complete sentence should be divided into the following six blocks:

Block 1: “只有/c 自身/rNP 硬/a ，/wP”

Block 2: “才/d 能/vM 对/p 不良/a 风气/n 、/wD”

Block 3: “腐败/a 现象/n 敢/vM 抓/v 敢/vM 管/v ，/wP”

Block 4: “不/dN 怕/v “/wLB 鬼/n ”/wRB ，/wP”

Block 5: “不/dN 信/v 邪/a ，/wP”

Block 6: “敢/vM 摸/v “/wLB 老虎/n ”/wRB 屁股/n 。/wE”

For every divided block, we need to make a decision whether the block forms an independent clause. So, the Chinese clause splitting task is converted into a sequence labeling problem.

We employ a structured SVM that predicts tag sequences, called an SVM Hidden Markov Model, or SVM-HMM [13]. This approach can be considered an hidden Markov model (HMM) because the Viterbi algorithm is used to find the highest scoring tag sequence for a given observation sequence. But it discriminatively trains models that are isomorphic to an kth-order HMM using the structured SVMs formulation. The scoring model employs a Markov assumption: each tag's score is modified only by the tag that came before it. In sequence tagging each input $x = (x_1, \ldots, x_n)$ is a sequence of feature vectors, and $y=(y_1, \ldots, y_n)$ is a sequence of labels $y_i \in \{1..k\}$ of matching length. Given the trained feature weight vector, the SVM-HMM tags new instances $x = (x_1, \ldots, x_n)$ according to:

$$\arg\max_y \left\{ \sum_{i=1}^{n} \left[\sum_{j=1}^{k} (x_i \bullet w_{y_{i-j} \cdots y_i}) + \varphi_{trans}(y_{i-j}, \ldots, y_i) \bullet w_{trans} \right] \right\} \tag{2}$$

In SVM-HMM model, the feature should be divided into two types: emission features and transition features. SVM-HMM learns one emission weight vector $w_{y_{i-j} \cdots y_i}$ for

each different kth-order tag sequence $y_{i-k} \cdots y_i$ and one transition weight vector w_{trans} for the transition weights between adjacent tags. When applying the SVM-HMM to Chinese clause splitting, the crux of the task is the design of suitable feature vectors and the loss function.

4.2 Loss Function

For Chinese clause splitting tasks, there are two possible loss functions: whole-sentence loss and Hamming loss. Whole-sentence loss gives credit only when the entire output sentence is correct: there is no notion of partially correct solutions. Hamming loss is more forgiving: it gives credit on a per label basis. To better express the difference between two outputs, we choose the Hamming loss as the loss function. For a true output y of length N and hypothesized output $\hat{y}$ (also of length N), the Hamming loss functions are given in Eq (3).

$$\ell^{Ham}(y, \hat{y}) = \sum_{n=1}^{N} 1[y_n \neq \hat{y}_n] \tag{3}$$

4.3 Emission Features

It is difficult to design a suitable set of features to capture the characteristic of Chinese clause, because we formulate a sequence of words (i.e. a block), not a single word, as a "token" to be tagged. According to definition of Chinese clause, a clause should include at least a predicate. But recognizing the predicates of Chinese sentence is difficult. Considering that the verb is a main type of POS that acts as predicate, we instead check whether every block includes a verb within it. By analyzing the instances in training data, we also find that the first and the last word and POS in every block play an important role in Chinese clause splitting.

We use the lexical and POS information within a fixed window based on blocks. We also consider different combinations of them from the same block or different blocks. The features are listed as follows:

- Word features:
  -- The first word in the block
  -- The last word in the block
- POS features:
  -- The first POS tag in the block
  -- The last POS tag in the block
- Punctuator features:
  -- Punctuation mark at the end of the block.
- Combinatorial features:
  -- The first word and the last word in the block
  -- The first POS tag and the last POS tag in the block

-- The last POS tag in the previous block and the first POS tag in the current block

- The features of inclusion of verb:
  -- Check if there exists a verb in the block.
  -- Check if there exist multiple verbs in the block.
  -- Check if there exists a noun occurring before the verbs in the block.
- The number of words appearing in the block.

4.4 Postprocessing with Rules

After interpreting a sequence of words separated by a punctuator as a block, we will make a decision whether the block forms an independent clause, to give the right boundary of every Chinese clause. That is, the left boundary of Chinese clause is implicitly decided. However, the double quotation marks is an exceptional punctuation mark acting as a separator of Chinese clauses, because the left double quotation mark sometimes leads to a left boundary of a clause, but it sometimes does not.

By analyzing the training data, we discover a rule: Either both of a pair of quotation marks or neither of them is in a clause. Based on that rule, postprocessing is done in detail as follows:

(1) If the text content between a pair of quotation marks is some simple words, the quotation marks should be contained in a single clause; If the content is a complete sentence, the quotation marks should not be contained in a single clause.

(2) If a left quotation mark is not in a clause, then the corresponding right quotation mark must not be in the clause.

(3) If a left quotation mark is in a clause, then the clause's end position must be behind the corresponding right quotation mark.

(4) If a right quotation mark is in a clause, then the clause's start position must be before the corresponding left quotation mark.

5 Experiments

This section will evaluate the effectiveness of our approach for Chinese event descriptive clause splitting through experiments.

5.1 Experimental Setting

We used the Tsinghua treebank that is published by Chinese Information Processing Society of China for ParsEval-2009 as data for training and testing the Chinese event descriptive clause splitting [1]. The data sets contain sentences with the words, the clause split solution, and tagged POS tags. The number of words and the number of clauses in the data sets are 186430 and 20429 respectively. The clauses vary in length from 1 to 71. The distribution of Chinese event descriptive clauses of different lengths is shown in figure 1.

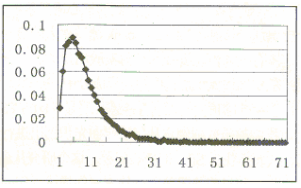


Fig. 1. The distribution of Chinese event descriptive clauses of different lengths in the data sets

The Chinese event descriptive clause splitting task of ParsEval-2009 included two types of test: open test and closed test. In the open tests systems could use any external data in addition to the training corpus to train their system. In closed tests, systems were only allowed to use information found in the training data. Absolutely no other data or information could be used beyond that in the training document. To verify the effectiveness of our approach itself, we are only interested in the closed test. That is, the set of features used by our system, as described in section 4.3, only exploited the information from the training data.

5.2 Experimental Results

For evaluating the task in a set of N sentences, the usual precision, recall and F1 measures are used:

$$\text{Precision} = \frac{\text{num of correctly recognized clauses}}{\text{num of recognized clauses}}$$

$$\text{Recall} = \frac{\text{num of correctly recognized clauses}}{\text{num of total clauses}}$$

$$\text{F1} = \frac{\text{Precision} \times \text{Recall} \times 2}{\text{Precision} + \text{Recall}}$$

To compare the effectiveness of our approach with other approaches, we first developed two baseline systems. The first baseline system is based on a decision tree classifier, and the second one uses the CRFs model [14] to split the Chinese clauses. Then we implemented the third system with the method proposed in this paper. In the first baseline system, the decision on whether the current block forms a complete clause is made independently with a decision tree classifier, and we use j48 algorithm in Weka as the decision tree classifier [15]. In the second baseline system, we also formulate the Chinese clause splitting task as a sequence labeling problem, and choose the CRFs model as the sequence tagging model, because a good result achieved by CRFs model for English clause splitting was reported in [10]. The results of the first baseline system and the second baseline system are shown in table 1. The second baseline system with CRFs model achieved better performance than the first baseline system with decision tree classifier.

Table 1. Results of two baseline systems

	Precision	Recall	F1
Decision tree	68.36	74.59	71.34
CRFs	73.37	76.64	74.97

Table 2. Results of our systems with different orders

	Precision	Recall	F1
1th-order model	75.15	76.99	76.06
2th-order model	75.95	78.82	77.36
3th-order model	76.36	80.02	78.15

When applying the SVM-HMM model to Chinese clause splitting, we can train different order models to express different length dependencies for both the transitions and the emissions. The results in table 2 show the performance of the applying 1th-order, 2th-order and 3th-order models to Chinese clause splitting task, respectively. We can observe that the higher order of model leads to a better performance. The fact also indicates that it is reasonable to treat the Chinese clause splitting task as a sequence tagging problem. Compared with other two baseline systems, our approach gives much better performance. The main reasons that the structured SVMs approach for Chinese clause splitting achieve better performance than CRFs are in the following ways. First, the CRFs model only optimizes log-0/1 loss over the structural output y, while the log-0/1 loss can not sufficiently capture the difference between the predicted output and the true output for clause splitting. Second, the CRFs model is not able to exploit higher order markov property to describe the dependencies for both the transitions and the emissions as the structured SVMs model.

6 Conclusion

In this paper, we explore a discriminative approach for Chinese event descriptive clause splitting task. By formulating the Chinese clause splitting task as a sequence labeling problem, we apply the structured SVMs model to Chinese clause splitting. As far as we know, this is the first work to Chinese clause splitting problem. We compare the approach proposed in this paper with two baseline systems and the experimental results show that our approach achieves a much better result. We will try to select more useful feature functions into the existing sequence tagging model in future work.

Acknowledgement

This work is supported by the National Natural Science Foundation of China under Grant No.60673043, the National High-Tech Research and Development Plan of China under Grant No. 2006AA010109, the National Basic Research Program of China under Grant No. 2010CB327903, and the Natural Science Foundation of Jiangsu Higher Education Institutions of China under Grant No. 07KJB520057.

References

1. Zhou, Q., Li, Y.: Chinese Chunk Parsing Evalution Tasks. Advances of computational Linguistics in China, pp. 130–135. Tsinghua University Press (2009) (in Chinese)
2. Tjong Kim Sang, E.F., Dejean, H.: Introduction to the CoNLL 2001 shared task: Clause identification. In: Daelemans, W., Zajac, R. (eds.) Proceedings of CoNLL, pp. 53–57 (2001)
3. Carreras, X., Màrquez, L.: Boosting Trees for Clause Splitting. In: Proceedings of the CoNLL 2001 Shared Task (2001)
4. Carreras, X., Màrquez, L., Punyakanok, V., Roth, D.: Learning and Inference for Clause Identification. In: Elomaa, T., Mannila, H., Toivonen, H. (eds.) ECML 2002. LNCS (LNAI), vol. 2430, p. 35. Springer, Heidelberg (2002)
5. Hammerton, J.: Clause identification with Long Short-Term Memory. In: Proceedings of CoNLL-2001, Toulouse, France, pp. 61–63 (2001)
6. Tjong Kim Sang, E.F.: Memory-based clause identification. In: Proceedings of CoNLL 2001, Toulouse, France, pp. 67–69 (2001)
7. Carreras, X., Màrquez, L., Castro, J.: Filtering-Ranking Perceptron Learning for Partial Parsing. Machine Learning Journal, Special Issue on Learning in Speech and Language Technologies 60(1-3), 41–71 (2005)
8. Collins, M.: Discriminative training methods for hidden markov models: theory and experiments perceptron algorithm. In: Proceeding of EMNLP, Philadelphia, PA, USA, pp. 1–8 (2002)
9. Hachey, B.C.: Recognizing clauses using symbolic and machine learning approaches. Master thesis. University of Edinburgh (2002)
10. Van Nguyen, V., Le Nguyen, M., Shimazu, A.: Clause Splitting with Conditional Random Fields. Information and Media Technologies 4(1), 57–75 (2009)
11. Tsochantaridis, I., Joachims, T., Hofmann, T., Altun, Y.: Large Margin Methods for Structured and Interdependent Output Variables. Journal of Machine Learning Research (JMLR), 1453–1484, September 6 (2005)
12. Joachims, T., Finley, T., Yu, C.-N.: Cutting-Plane Training of Structural SVMs. Machine Learning 77(1), 27–59 (2009)
13. Altun, Y., Tsochantaridis, I., Hofmann, T.: Hidden Markov Support Vector Machines. In: Proceedings of International Conference on Machine Learning, ICML (2003)
14. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proceedings of the International Conference on Machine Learning (ICML), pp. 282–289. Morgan Kaufmann Publishers, San Francisco (2001)
15. Witten, I.H., Frank, E.: Data mining: practical machine learning tools and techniques with Java implementations. Morgan Kaufmann, San Francisco (2000)

An Experimental Study on Unsupervised Graph-based Word Sense Disambiguation

George Tsatsaronis[1], Iraklis Varlamis[2], and Kjetil Nørvåg[1]

[1] Department of Computer and Information Science,
Norwegian University of Science and Technology
{gbt,Kjetil.Norvag}@idi.ntnu.no
[2] Department of Informatics and Telematics, Harokopio University of Athens
varlamis@hua.gr

Abstract. Recent research works on unsupervised word sense disambiguation report an increase in performance, which reduces their handicap from the respective supervised approaches for the same task. Among the latest state of the art methods, those that use semantic graphs reported the best results. Such methods create a graph comprising the words to be disambiguated and their corresponding candidate senses. The graph is expanded by adding semantic edges and nodes from a thesaurus. The selection of the most appropriate sense per word occurrence is then made through the use of graph processing algorithms that offer a degree of importance among the graph vertices. In this paper we experimentally investigate the performance of such methods. We additionally evaluate a new method, which is based on a recently introduced algorithm for computing similarity between graph vertices, P-Rank. We evaluate the performance of all alternatives in two benchmark data sets, Senseval 2 and 3, using WordNet. The current study shows the differences in the performance of each method, when applied on the same semantic graph representation, and analyzes the pros and cons of each method for each part of speech separately. Furthermore, it analyzes the levels of inter-agreement in the sense selection level, giving further insight on how these methods could be employed in an unsupervised ensemble for word sense disambiguation.

1 Introduction

Word Sense Disambiguation (WSD) addresses the problem of selecting the most appropriate sense for a word, among several offered from a dictionary or a thesaurus, with respect to its context. WSD algorithms are used in several natural language processing tasks, such as machine translation, and speech processing, and the performance of the disambiguation procedure is critical to their success [6]. WSD has also been reported to boost performance of text retrieval, document classification, and document clustering tasks [13,20]. All these findings, strengthen the need for fast and accurate WSD algorithms.

The various solutions found in the WSD bibliography face the tradeoff between unsupervised and supervised methods. The former usually offer fast execution time but low accuracy, while the latter suffer from the *knowledge acquisition bottleneck* problem because they require extensive training in a large amount of manually annotated

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 184–198, 2010.

data. Unsupervised graph-based WSD techniques [2,24,35,26,38] have been attracting a wider focus lately, mainly because they have managed to truncate the accuracy gap from the supervised methods. The key to these methods' achievement is the rich semantic model that they employ. More specifically, they map the words to be disambiguated and their respective candidate senses to graphs, which are enhanced with nodes and semantic edges from word thesauri (e.g., WordNet). On top of this representation, they use a node ranking or node activation algorithm, which after several iterations concludes to the best candidate sense for each word, which is usually the highest ranked sense node after the convergence of the vertices' values.

In this paper, we compare the performance of several unsupervised graph-based WSD methods. We also apply for the first time a new vertices similarity measure, capitalizing on the structural similarity of the graph vertices. In the experimental evaluation we use the English WordNet [10] as our lexical database, and the data from the Senseval 2 [31] and 3 [36] *English all words* task as a benchmark. We present the comparative results of several vertex ranking algorithms [4,8,17], and vertex similarity algorithms [42]. The contributions of this work can be summarized in the following: (a) thorough experimental evaluation and analysis of the performance of seven state of the art unsupervised graph-based WSD methods, (b) application -for the first time- of the node similarity algorithm P-Rank [42], in the word sense disambiguation task, (c) generalized comparison and analysis against state of the art WSD approaches, both supervised and unsupervised, offering an experimental survey of the current top methods in word sense disambiguation, and (d) analysis of the methods inter-agreement in the sense selection level, that can give further insight into a possible inclusion of those methods in an ensemble of approaches.

The rest of the paper is organized as follows: Section 2 discusses the related work, and gives a short overview of the state of the art in word sense disambiguation. Section 3 presents in detail the graph construction and graph processing algorithms and their application in WSD, and also discusses the space and time complexity of the examined methods. Section 4 experimentally evaluates the compared approaches and illustrates the advantages of each method per part of speech (POS). Furthermore, it generalizes the comparison against top performing WSD methods in the Senseval 2 and 3 data sets. Finally, Section 5 concludes and provides pointers to future work.

2 Related Work

2.1 Supervised Word Sense Disambiguation

The field of WSD is a well studied research area [15,28], mainly because the application of WSD may improve the performance of several tasks, like machine translation and text classification. A crucial component in such critical applications is the achieved accuracy of the underlying WSD system. In general, supervised WSD methods outperform their unsupervised rivals but they require extensive training in large data sets. Recent research results [28] show that the accuracy of state of the art supervised WSD methods is above 60% with an upper bound reaching 70% for all words, fine-grained WSD, while the accuracy of unsupervised methods is usually between $45 - 60\%$.

Supervised WSD approaches that report interesting performance results comprise the works of Pedersen [33], Florian et al. [11], and Carpuat et al. [40]. Pedersen uses an ensemble of 9 classifiers selected from a set of 81 Naive Bayes classifiers and requires at least one training instance for each different sense of the target word that exists in the lexicon. Similarly, Florian et al. use an ensemble of 6 different classifiers (Naive Bayes, Transformation-base learning, etc.) and report similar requirements for training samples. Carpuat et al. use a method that exploits a nonlinear kernel principal component analysis (KPCA) technique [40]. The KPCA-based model acts as the voting mechanism over a set of classifiers that learn to predict the correct sense and decides on which of the suggested senses should be selected.

State of the art results in supervised WSD have been reported by the SenseLearner system of Mihalcea and Csomai [23], the Simil-Prime system introduced by Kohomban and Lee [18], and the system developed by Hoste et al. [14]. In [23] the authors suggest the construction of seven semantic models, which are trained using the Timbl memory based learning algorithm. The Simil-Prime method [18] is trained to disambiguate words into generic semantic classes, and consequently casts the generic semantic classes back to finer grained senses, using heuristical mapping. The major drawback of this method is the use of heuristics, which cannot guarantee that finer senses will not be missed. Another drawback is the fact that it uses a decision-tree based implementation of the k-nn classifier, which raises the execution cost (mainly the space complexity) since many training examples need to be reexamined for each target word. The memory-based learning approach proposed by Hoste et al. uses voting among word-experts to decide on the correct sense. The method stores all instances in memory during training and testing, which results in both high space and time complexity.

Finally, we should mention the winners of the Senseval 2 and 3 *All English Words Task* which were the supervised WSD systems *SMUaw* [21] and *GAMBL* [9] respectively. *SMUaw* was based on pattern learning from sense-tagged corpora and instance-based learning with automatic feature selection. In the cases where the existing patterns failed to disambiguate a word and no more training data existed, the method selected the most frequent sense for the word, which resulted in high recall levels, but affected precision. In *GAMBL* word experts are trained using memory-based classifiers, that learn to predict the correct sense of each word, thus requiring extensive training.

2.2 Unsupervised Word Sense Disambiguation – The Graph-Based Methods

The list of unsupervised WSD methods is long and comprises corpus-based [41], knowledge-based, such as Lesk-like [19] and graph-based [2,24,35,26,38] methods, as well as ensembles [5] that combine several methods. From all types of unsupervised WSD methods, we focus on the graph-based ones, which demonstrate high performance and seem to be a promising solution for unsupervised WSD. The first step of graph-based WSD methods relies in the construction of semantic graphs from text. The graphs are consequently processed in order to select the most appropriate meaning[1] of each examined word, in its given context.

[1] In the remaining of the paper, the words *concept*, *sense*, and *synset* may be used interchangeably to describe the meaning of a word, among the several offered by a dictionary or a word thesaurus.

One of the most influential WSD works in this direction is the disambiguation algorithm of Sussna [37], which uses the WordNet graph as a basis and examines all nouns in a window of context and assigns a sense to each noun in a way that minimizes a semantic distance function among all selected senses. In [1], Agirre and Rigau introduced and applied a similarity measure based on conceptual density between noun senses. The measure was based on WordNet's is-a hierarchy and measured the similarity between a target noun sense and the nouns in the surrounding context. More recently, Banerjee and Pedersen [3] suggested an adaptation of the original Lesk algorithm for the WordNet graph. In [24] and [38] authors use WordNet as a graph, defining the vertices as synsets and the edges as the semantic relations connecting synsets. Both methods construct synset graphs from text in the first step. Then, the former method applies the PageRank algorithm to rank the synset vertices whereas the latter employs a spreading activation technique to process the network (*SAN*) and selects the most active sense nodes after the spreading of the activation as the senses disambiguating the respective words. In [27] Navigli introduced a different graph construction method, the Structural Semantic Interconnections (SSI-HITS), in which all candidate senses are connected and consequently ranked using the HITS algorithm. SSI-HITS is based on a measure that maximizes the degree of mutual interconnection among a set of senses. Finally, in [2] Agirre and Soroa use the PageRank algorithm, instead of HITS, and a wider knowledge-base (WordNet and Extended WordNet [25] relations).

Examining unsupervised graph-based WSD from another perspective, Sinha and Mihalcea [35] propose an unsupervised graph-based method for WSD, based on an algorithm that computes graph centrality of nodes in the constructed semantic graph. To measure the centrality of the nodes, they use the indegree, the closeness, and the betweenness of the vertices in the graph, as well as PageRank. They also employ five known measures of semantic similarity or relatedness to compute the similarity of the nodes in the semantic graph, based on an idea initially presented by Patwardhan et al. [32]. Similarly, in [29], Navigli and Lapata explore several measures for analyzing the connectivity of semantic graph structures in *local* (i.e., per individual node) or *global* (i.e. for the whole graph) level. They evaluate in-degree and eigenvector centrality, maximum flow, compactness, graph entropy and edge density. They conclude that local measures perform better than global measures for the WSD task.

The examination of related literature revealed a wide variety of options in unsupervised graph-based WSD techniques. In the following of this study, we examine more closely the empirical evaluations of these methods, and analyze the reasons behind a boost in performance, which can be either the levied semantic representation or the graph processing technique itself. Furthermore, we examine the interagreement of these methods in the selection of senses when the same graph representation is employed. For this reason we implement four graph processing techniques (PageRank, SAN, HITS and P-Rank) and evaluate their performance in the same semantic representation.

3 Assigning Senses to Words in Semantic Graphs

This section presents the four graph processing methods that were selected for evaluation: SAN [8], PageRank [4], HITS [17] and P-Rank [42]. Though three of those

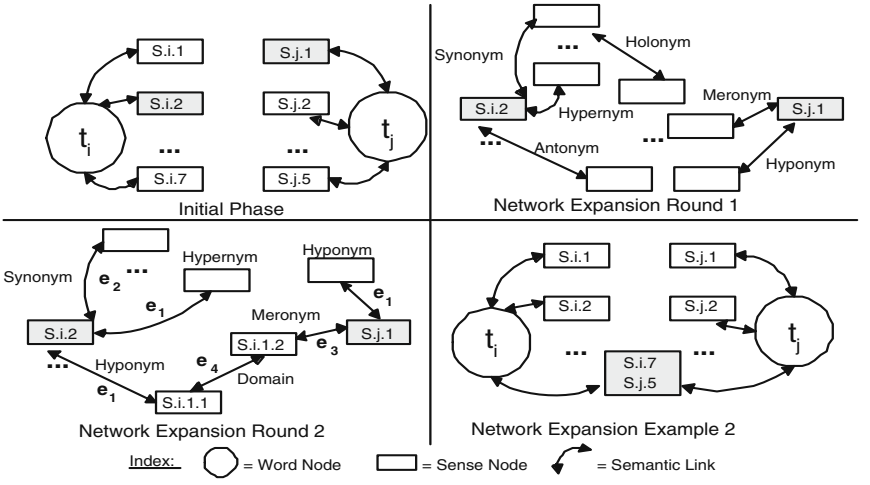


Fig. 1. Semantic Network Construction for Spreading of Activation

methods have been applied before in WSD (SAN-based WSD [38,39], PageRank-based WSD [2,35], and HITS-based WSD [29]), they have never been evaluated in tandem using the same semantic representation of text. Thus, in order to provide a comparative evaluation, we used the same semantic representation (i.e., the same graph) for all methods. More specifically, we adopt the semantic network construction method that was introduced in [38]. The method utilizes all of the available semantic relations in WordNet 2.0. Furthermore, it employs a novel weighting scheme for the edges connecting the sense nodes.

3.1 Semantic Graph Construction

The semantic network construction method of Tsatsaronis et al [38] creates a semantic network for each sentence, that contains only the words that have entries in WordNet and assumes that these words have been tagged with their parts of speech (POS). The method, as depicted in Figure 1, initially adds the word nodes and their senses to the network (*initial phase*). Consequently, it adds all the thesaurus senses which are semantically related to the existing senses of the network (*expansion round 1* for the senses *S.i.2* and *S.j.1* respectively). Expansion continues iteratively until there is a path between every pair of the initial word nodes. In this step, the network ceases growing and is considered as *connected*. If there are no more senses to be expanded and the respective network is not connected, the words of that sentence cannot be disambiguated, which means a loss in coverage. Once the network is ready to be processed, the weights of each edge are added (*expansion round 2*). The weights are based on the frequency of occurrence of each edge type in the constructed network [38]. During the construction of the networks, it might be the case that two words share the same senses. This case is depicted in *expansion example 2* of Figure 1. In this case, a single sense node is added to the network (i.e., the sense nodes in the network represent WordNet synsets).

Apart from the specific method that we employ for constructing semantic graph, several other alternatives exist in the literature. In [39] the authors utilize the gloss words of the WordNet entries to construct semantic graphs. The network constructed in [24] is very similar, since it is based on some of the WordNet relations between senses, but differs in that it defines additional composite semantic relations (called *xlinks*). In [2], the authors use additional relations from the Extended WordNet and manual disambiguations of its glosses for the different entries. Finally, in [29], the network construction approach has an allowed upper bound on the length of the semantic paths. The main reason behind our selection is that the selected method incorporates all of the explicit semantic relations in WordNet and adopts an edge weighting scheme that takes into account the importance of each edge type. The selected method has been first evaluated in [38] against the approaches of [39] and [24] in graph construction and is evaluated again in this study against more recent techniques. Results show that the selected method performs better or equally well than other graph construction methods, for the same graph processing method. For example, in [38], it was compared against the representation of Veronis and Ide [39] and an accuracy improvement was reported.

3.2 Spreading of Activation (SAN) Method

The method introduced by Tsatsaronis et al. in [38], relies on spreading of activation in semantic networks (*SAN*) for WSD and it was based on an initial approach by Veronis and Ide [39] for constructing SAN for WSD. The constructed graph is processed with an iterative spreading activation strategy incorporating the fan-out and the distance constraints, as described by Crestani [8]. More specifically, the nodes initially have an activation level of 0, except for the input word nodes, whose activation is 1. In each iteration, every node propagates its activation to its neighbors, as a function of its current activation value and the weights of the edges that connect it with its neighbors. At each iteration p every network node j has an activation level $A_j(p)$ and an output $O_j(p)$, which is a function of its activation level, as shown in equation 1.

$$O_j(p) = f(A_j(p)) \tag{1}$$

The output of each node affects the next-iteration activation level of any node k towards which node j has a directed edge. Thus, the activation level of each network node k at iteration p is a function of the output, at iteration $p-1$, of every neighboring node j having a directed edge e_{jk}, as well as a function of the edge weight W_{jk}, as shown in equation 2. Although this process is similar to the activation spreading of feed-forward neural networks, the reader should keep in mind that the edges of SANs are bi-directional (for each edge, there exists a reciprocal edge). A further difference is that no training is involved in the case of SANs.

$$A_k(p) = \sum_j O_j(p-1) \cdot W_{jk} \tag{2}$$

Unless a function for the output O is chosen carefully, after a number of iterations the activation floods the network nodes. We use the function of equation 3, which incorporates fan-out and distance factors to constrain the activation spreading; τ is a threshold value.

$$O_j(p) = \begin{cases} 0 & , \text{ if } A_j(p) < \tau \\ \frac{F_j}{p+1} \cdot A_j(p) & , \text{ otherwise} \end{cases} \quad (3)$$

Equation 3 prohibits the nodes with low activation levels from influencing their neighboring nodes. The factor $\frac{1}{p+1}$ diminishes the influence of a node to its neighbors as the iterations progress (intuitively, as "pulses" travel further). Function F_j is a fan-out factor, defined in equation 4. It reduces the influence of nodes that connect to many neighbors.

$$F_j = (1 - \frac{C_j}{C_T}) \quad (4)$$

C_T is the total number of nodes, and C_j is the number of nodes directly connected to *j* via directed edges from j.

3.3 PageRank (PR) Method

In this work we also investigate on the potential of applying PageRank in the semantic networks shown in Figure 1. Thus, we designed another WSD algorithm (*PR*) that processes the constructed networks with PageRank. The PageRank formula that we used is a simple variation of the original PageRank equation, which takes into account edge weights as well. This variation was first introduced by Mihalcea et al. in [24]. Equation 5 shows the original PageRank formula and Equation 6 shows its weighted variation that we use to process the networks. $S(V_i)$ (and $WS(V_i)$ respectively) is the PageRank value of vertex V_i, d is the damping factor, $Out(V_j)$ is the number of outgoing links from vertex V_i and w_{ij} is the weight of the edge connecting vertices V_i and V_j.

$$S(V_i) = (1-d) + d \sum_{j \in In(V_i)} \frac{S(V_i)}{|Out(V_j)|} \quad (5)$$

$$WS(V_i) = (1-d) + d \sum_{V_j \in In(V_i)} \frac{w_{ij}}{\sum_{V_k \in Out(V_j)} w_{jk}} WS(V_j) \quad (6)$$

The *SAN* method can then be easily modified to process the constructed networks with equation 6, instead of spreading of activation. As a damping factor (d) we set 0.85, as in the original formula by Brin and Page [4], and we did not optimize this parameter. After the PageRank values stabilize, the sense nodes with the highest PagerRank scores for each target word are selected to disambiguate each word occurrence. The difference between this new PageRank-based WSD method and the method of Mihalcea et al. [24] is the semantic representation of the sentences used. In Section 4 we show that this difference in the semantic representation is important and yields an increase of almost 5% in the disambiguation accuracy. Furthermore, regarding the difference with the PageRank-based WSD algorithm introduced by Agirre and Soroa [2], this relies not only in the semantic representation of text, but also in the used PageRank formula. More specifically, Agirre and Soroa bias the PageRank execution to concentrate the initial probability mass uniformly over the word nodes that constitute the context of the word to be disambiguated.

3.4 HITS Method

In the same adopted semantic network representation we also utilize the HITS algorithm as a means of ranking the sense nodes and disambiguating text. Initially the algorithm was introduced by Kleinberg [17], and its idea is based on identifying the authorities (the most important pointed nodes in a graph) and the hubs (the nodes that point to authorities). The algorithm preceded PageRank, and it has several disadvantages, like the fact that is prone to clique-attack (i.e., densely connected neighborhoods of the graph can aggregate large scores). Its application in WSD is thus interesting, so as to investigate on how this affects the results of the task.

In HITS, each graph node has a pair of values (its hub and its authority score). Initially these values are set to 1. Then the algorithm runs in steps iteratively, to update the hub and the authority scores for each node, following the authority and the hub update rules respectively, shown in Equations 10 and 9.

$$authority(p) = \sum_{q \in In(p)} hub(q) \tag{7}$$

$$hub(p) = \sum_{r \in Out(p)} authority(r) \tag{8}$$

where *authority(i)* of a node *i* is its authority value, and *hub(i)* is its hub value, *In(i)* is the set of nodes that link to *i*, and *Out(i)* the set of nodes that *i* links to. Since our graph has edges on weights, we are using a modification of Equations 9 and 10, that take into account the edge weights. The equations are modified as follows:

$$authority(p) = \sum_{q \in In(p)} w_{q,p} \cdot hub(q) \tag{9}$$

$$hub(p) = \sum_{r \in Out(p)} w_{p,r} \cdot authority(r) \tag{10}$$

where $w_{i,j}$ is the edge weight of the edge leaving from i and linking to j. Eventually, after a large number of iterations, the authority and the hub values may converge if a normalization is used, which divides at each step each authority value by the sum of the authority values and each hub value by the sun of the hub values. In practice, we are using a small threshold (i.e., 10^{-4}) which acts as a criterion of change from step to step during the iterations, and when the changes affecting the authority and the hub values do not surpass it for any node in the graph, we assume that the values have converged. Eventually, the sense node with the highest authority value is selected as the most appropriate sense for each word.

3.5 P-Rank Method

The P-Rank measure [42] (Penetrating Rank) is a very recently introduced measure of structural similarity for nodes in an information network. It enriches a former successful measure of node similarity in information networks, SimRank [16]. In their paper, the authors prove that P-Rank is a unified structural similarity network, under which

all state of the art similarity measures, including CoCitation, Coupling, Amsler and SimRank, are just its special cases. In this work, it is for the first time that P-Rank is applied for WSD. The basic idea behind P-Rank is that two vertices in an information network are similar, if they are referenced by similar vertices, and they also reference similar vertices. P-Rank is recursive, and it executes over all pairs of vertices in a given graph. Let a graph G and two vertices a, b. Also let $R_k(a,b) = R_k(b,a)$ denote the P-Rank similarity value for the pair of vertices (a,b), at iteration k. Then, P-Rank can be formalized as shown in Equation 11:

$$R_{k+1}(a,b) = \lambda \cdot \frac{C}{|I(a)||I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} R_k(I_i(a), I_j(b)) + (1-\lambda) \cdot \frac{C}{|O(a)||O(b)|} \sum_{i=1}^{|O(a)|} \sum_{j=1}^{|O(b)|} R_k(O_i(a), O_j(b)) \quad (11)$$

where $I(n)$ of a vertex n is the set of its incoming neighbors, $I_i(n)$ is the i_{th} element of this set, and the respective holds for the $O(n)$ notation. The parameter $\lambda \in [0,1]$ balances the relative weight of in- and out-link directions, and is usually set to 0.5. $C \in [0,1]$ is a damping factor for in- and out-link directions, usually taking the value of 0.8, according to the authors. Finally, the number of iterations needed, for the vertex pairs similarity values to converge, is reported to be empirically at $ln(n)$, where n is the number of vertices in the graph. Since our semantic networks have weights on their edges, we are using a modification of Equation 11 to accommodate our weighting scheme. Thus, we modify the definition of $|I(a)|$, and $|O(a)|$ of a vertex a, as follows:

$$|I(a)| = \sum_{i \in Incoming(a)} w_{i,a} \quad (12)$$

$$|O(a)| = \sum_{j \in Outgoing(a)} w_{a,j} \quad (13)$$

where *Incoming(a)* and *Outgoing(a)* are the lists of the incoming and outgoing neighbors of a. Then, the sums in equation 11 are of course modified to run over the respective $|Incoming(a)|$ and $|Outgoing(a)|$. After the convergence of the similarity values between all pairs of vertices, the correct sense for each word is the sense node having the highest similarity with the respective word node in our networks.

3.6 WSD Methods Complexity

With regards to the complexity of the four methods, in [38] it was shown that the construction time of the semantic networks is $O(n \cdot k^{l+1})$ where n is the number of words we disambiguate, k is the maximum branching factor of the used thesaurus nodes and l is the maximum semantic path length in the thesaurus. The time complexity of *SAN* is $O(n^2 \cdot k^{2l+3})$. The time complexity of *PR* is $O(n^2 \cdot k^{\frac{3}{2}l+3})$ in the worst case, where the network has $n \cdot k^{\frac{l}{2}+1}$ nodes and $n \cdot k^{l+2}$ edges, and similar is the time complexity of the

Table 1. Occurrences of polysemous and monosemous noun (N), verb (V), adjective (Adj.), adverb (Adv.) and total (All) words of WordNet 2 in Senseval 2, and 3

	Senseval 2					*Senseval 3*				
	N	**V**	**Adj.**	**Adv.**	**All**	**N**	**V**	**Adj.**	**Adv.**	**All**
Mono.	260	33	80	91	464	193	39	72	13	317
Poly.	813	502	352	172	1839	699	686	276	1	1662
Av. Poly.	4.21	9.9	3.94	3.23	5.37	5.07	11.49	4.13	1.07	7.23
Av. Poly. (P. only)	5.24	10.48	4.61	4.41	6.48	6.19	12.08	4.95	2.0	8.41

HITS and the method. The time complexity of P-Rank in the worst case is even larger; $O(n^4)$ [42], since it runs over all pair combinations of vertices. Its space complexity though, is the same with the rest of the algorithms. The space complexity at the worst case is equal to the complexity required in memory to construct the semantic networks, and for the disambiguation of n words is equal to $O(n^2 \cdot k^{2l+3})$.

4 Experimental Evaluation

In this section we proceed with an empirical evaluation of the performance of the four methods, which examines two criteria: (1) the accuracy of the methods in two benchmark data sets, and (2) the inter-agreement rate of the methods in the sense selection level, in the same data sets. In order to evaluate the examined methods we use the Senseval 2 [31] and 3 [36] *All English Words Task* data sets for testing. These data sets were manually annotated with the correct senses by human annotators, before the respective competitions were conducted.[2] In Table 1 we present the statistics of those data sets, including average polysemy of words, both with (Av. Poly.) and without (Av. Poly. (P. only)) taking into account monosemous words. Senseval 2 is easier to disambiguate than Senseval 3, as the average polysemy is larger in the latter. Adverbs are very easy to disambiguate and are usually excluded from the evaluation (e.g., Senseval 3 has only 13 adverb occurrences with average polysemy close to 1). The verb POS is the most difficult to disambiguate, since a typical verb has more than 8 different senses from WordNet.

Regarding the lower and upper bounds of WSD methods in those data sets, a straightforward lower bound is to select randomly a sense for each word occurrence. This disambiguation method would produce an accuracy of around 20% for Senseval 2 and SemCor, and 14% for Senseval 3. A reasonable upper bound, as stated in [28], would be the interannotator agreement or intertagger agreement (ITA), that is, the percentage of words tagged with the same sense by two or more human annotators. The interannotator agreement on coarse-grained (lexicons with few and clearly distinct senses for each lemma are used), possibly binary (two senses per lemma), sense inventories is calculated around 90% [12,29], whereas on fine-grained, WordNet-style sense inventories, where there are many senses per lemma and which are often hard to distinguish, the inter-annotator agreement is estimated between 67% and 80% [7,30,36].

[2] http://www.senseval.org/

Table 2. Overall and per POS accuracies (%) of WSD methods in Senseval 2, and 3 (*All English Words Task* data sets) for all POS, excluding adverbs

Method	Senseval 2				Senseval 3			
	N	V	Adj.	All	N	V	Adj.	All
SAN	53.9	31.7	59.0	49.5	50.8	36.5	58.0	46.8
PR	69.5	37.2	59.0	58.8	61.8	47.3	60.6	56.7
HITS	69.1	36.6	59.1	58.3	69.2	40.4	66.7	57.4
P-Rank	51.3	27.31	57.4	45.6	60.6	29.9	67.8	52.1
Mih05	57.5	36.5	56.7	52.0	*n/a*	*n/a*	*n/a*	51.8
Agi09	70.4	38.9	58.3	59.5	64.1	46.9	62.6	57.4
Nav07	*n/a*	*n/a*	*n/a*	*n/a*	61.9	36.1	62.8	52.5
FS	74.0	42.4	63.1	63.7	70.9	50.7	59.7	61.3

4.1 Empirical Evaluation of Unsupervised Graph-Based WSD Methods

Table 2 shows the accuracy of the four methods for all POS in the two data sets. We have also added in the comparison, results from related methods with regards to unsupervised graph-based WSD. These are: the method of Mihalcea et al. [22] (Mih05), the method of Agirre and Soroa [2] (Agi09), and the results from the work of Navigli and Lapata [29] (Nav07). For this latter work, because the authors test and compare several graph connectivity measures, the table contains the numbers of their *KPP* measure, which was shown by their analysis to be the best performing graph-based measure overall. Note also, that adverbs are omitted in the comparison, since they are very few in number in the Senseval competitions, compared to the rest POS. Whenever results were not available, because they were not reported in the literature, an entry *n/a* exists in the respective cell. Finally, we have also added in the comparison a simple heuristic method (*FS*) that always selects the first sense of the target word from WordNet (i.e., the most frequent) to conduct the disambiguation. Though this method is usually reported as a baseline for the supervised systems (the unsupervised systems' baseline is the random assignment of senses), we have added it into the comparison, so that practitioners of WSD have a clear idea of the performance the unsupervised WSD systems can offer against the supervised ones.

As Table 2 shows the *SAN* method has stable performance, obtaining an accuracy very near 50%, overall for all POS. The *PR* method shows impressive increase in accuracy over the method of Mihalcea et al., which is due to the different semantic representation used through the constructed semantic networks, since the PageRank formula remaind the same in both cases. The *HITS* method performs overall better than *SAN* and its performance is very close to the *PR* method. In fact, *HITS* seems to be performing equally (Senseval 2) or better (Senseval 3) for the noun POS than *PR*, and the same holds for the adjective POS. For the verb POS, *PR* performs overall better than *HITS*. The *P-Rank* method does not seem to perform very well against the rest unsupervised graph-based techniques, but this is in accordance with the results reported by Navigli and Lapata [29], who reported lower results than other graph-based methods for the *betweenness* and the *indgree* measures of structural similarity in semantic graphs. The method of Agirre and Soroa also performs very well, and is in fact the best

Table 3. *SAN*, *PR*, *HITS*, and *P-Rank* methods' pairwise inter-agreement (%) in Senseval 2 and 3 (*All English Words Task* data sets) for all POS, excluding adverbs

Pair	Senseval 2				Senseval 3			
	N	V	Adj.	All	N	V	Adj.	All
SAN - PR	51.51	35.74	54.16	47.86	53.17	49.48	49.83	51.21
SAN - HITS	52.42	23.89	57.55	39.51	50.6	40.38	50.16	46.68
SAN - P-Rank	50.84	27.16	63.46	46.77	66.52	32.94	69.04	55.37
PR - HITS	62.56	34.93	64.32	55.54	60.36	44.64	66.88	55.57
PR - P-Rank	50.55	30.95	67.3	48.1	68.2	30.58	71.42	55.78
HITS - P-Rank	53.88	23.8	59.61	46.83	67.78	31.76	69.04	54.17

unsupervised graph-based method in Senseval 2, and has the same performance with *HITS* in the Senseval 3 data set. The performance difference between these two methods is not statistically significant at the 0.95 confidence level, if one examines their overall accuracy in the respective data sets. The *KPP* measure of Navigli and Lapata cannot match the accuracy of *PR*, *HITS* and the method of Agirre and Soroa. Regarding the *FS* method, though simple, outperforms every other compared method. It obtains very high accuracies, always above 60% for all POS. Its performance in nouns and adjectives is impressive, but in the verbs, due to their large average polysemy, the performance drops dramatically, compared to the rest POS. In another interesting unsupervised approach, Pedersen and Kolhatkar [34] perform disambiguation in Senseval 2 and 3, using measures of semantic relatedness. Their best reported results in F-Measure were 59% for Senseval 2 and 54% for Senseval 3, performance which is almost the same with PR and HITS in Senseval 2, but slightly worse in Senseval-3.

One additional comparison we would like to make regarding the four studied methods (*SAN*, *PR*, *HITS*, and *P-Rank*) is to examine the percentage of times the four methods agree in the sense selection level. Previous studies have shown that the ensemble of methods can lead to increased WSD accuracy [5]. A prerequisite is that the methods do not agree very often, so that there is a potential benefit from the ensemble. In this direction, we have measured their pair-wise inter-agreement rate (i.e., the percentage of the same sense assignment to the total sense assignments performed). Table 3 shows the inter-agreement rate for all pairwise combinations of the four methods, separately for each POS, and for each of the two examined data sets. The aim of this analysis is to investigate whether a potential combination of any subset of the four methods in an ensemble of unsupervised methods (e.g., [5]), would be expected to yield interesting results. The performance of the ensemble is strongly related to the pluralism of suggestions of the underlying WSD methods. As the table shows, the pairwise inter-agreement rate of the four methods is always lower or very close to 70% in all cases. The lowest inter-agreement rates are reported for the verb POS, which is an expected outcome, since the verbs are more polysemous than the rest POS. The lowest inter-agreement rates are reported for the *SAN-PR* and *SAN-HITS* pairs. This means that in a possible ensemble, the combination of *SAN* with *PR* or *HITS* could boost the overall performance. In parallel, we can observe from the table that all the methods seem to agree more in a pairwise manner in Senseval 3 than in Senseval 2. This is an interesting finding, because Senseval 3 is more polysemous than Senseval 2, and maybe the reverse was expected. A possible interpretation is

Table 4. Accuracies (%) on Senseval 2 and 3 *All English Words Task* data sets, excluding adverbs

Dataset	SenseLearner	Simil-Prime	SSI	WE	FS	PR	HITS	Agi09
Senseval2	64.82	65.00	n/a	63.2	63.7	58.8	58.3	59.5
Senseval3	63.01	65.85	60.4	n/a	61.3	56.7	57.4	57.4

that in the case of Senseval 3 the networks are larger, and thus more densely connected, and so the applied measures recognize more easily the most important vertices. Overall, these findings show that the combination of these four graph-based measures has great potential due to their relatively low level of inter-agreement.

4.2 Comparison with State of the Art WSD Methods

In this section we generalize the comparison of the unsupervised graph-based WSD methods with the state of the art results reported in the WSD literature, independently of the type of methods used. In this direction, we compare the top 3 methods from Table 2, namely *PR*, *HITS* and the method of Agirre and Soroa, with the highest results reported i the WSD literature for Senseval 2 and 3. Thus, we compare with the methods of Mihalcea and Csomai [23] (SenseLearner), Kohomban and Lee [18] (Simil-Prime), Navigli [26] (SSI), and Hoste et al. [14] (WE), in the Senseval 2 and 3 data sets. Table 4.2 shows the respective accuracies, where available. We can also refer to the unsupervised enseble method of Brody et al. [5] only in the noun POS of Senseval 3 data set, since their method's evaluation is limited to that. Brody et al. report an accuracy of 63.9% in Senseval 3 nouns (Senseval 2 is N/A) with an upper bound of their ensemble close to 70%. From the results of Table 4.2 we can observe that the top performing method appears to be the *Simil-Prime*, with very high overall accuracy, equal or above to 65%. We have to note though, that the latter cannot disambiguate adjectives and adverbs, residing in the FS method to perform the task for these two POS. It is also obvious from the results table, that though the unsupervised graph-based WSD techniques cannot match the accuracy of the rest of the methods, they have clearly closed the gap very much towards a possible match in the near future.

5 Conclusions and Future Work

In this work we presented an experimental study of unsupervised graph-based WSD techniques. The aim was to analyze the performance of known techniques for processing semantic graphs, keeping the same semantic representation, so that the comparison is compatible. In our comparison we included a spreading of activation method, the PageRank and HITS algorithms, as well as, for the first time, the P-Rank structural similarity measure for vertices in information networks. A thorough experimental evaluation was conducted in two benchmark WSD data sets. We also compared against other known unsupervised graph-based WSD techniques, that do not use the same semantic representation, as well as against the top reported results in the two data sets. Furthermore, we analyzed the pairwise inter-agreement rate between the examined methods, and we showed that it is low for most pairs, leading to the conclusion that several of these methods can form up an ensemble. Our study also showed that the gap in accuracy between

supervised methods and unsupervised graph-based techniques, has been truncated over the last years, constituting a solid evidence that there is still room for improvement, given also the fact that thesauri, like WordNet, will keep developing. In the future we plan to design unsupervised ensembles of graph-based methods, taking advantage of the relatively low inter-agreement rate and aiming at a high accuracy learner for the task.

References

1. Agirre, E., Rigau, G.: Word sense disambiguation using conceptual density. In: Proc. of COLING, pp. 16–22 (1996)
2. Agirre, E., Soroa, A.: Personalizing pagerank for word sense disambiguation. In: Proc. of EACL, pp. 33–41 (2009)
3. Banerjee, S., Pedersen, T.: Extended gloss overlaps as a measure of semantic relatedness. In: Proc. of IJCAI (2003)
4. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. Computer Networks and ISDN Systems 30, 1–7 (1998)
5. Brody, S., Navigli, R., Lapata, M.: Ensemble methods for unsupervised wsd. In: Proc. of COLING/ACL 2006, pp. 97–104 (2006)
6. Chan, Y., Ng, H., Chiang, D.: Word sense disambiguation improves statistical machine translation. In: Proc. of ACL (2007)
7. Chklovski, T., Mihalcea, R.: Exploiting agreement and disagreement of human annotators for word sense disambiguation. In: Proc. of RANLP (2003)
8. Crestani, F.: Application of spreading activation techniques in information retrieval. Artificial Intelligence Review 11, 453–482 (1997)
9. Decadt, B., Hoste, V., Daelemans, W., van den Bosch, A.: Gambl, genetic algorithm optimization of memory-based wsd. In: Proc. of the Senseval3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text (2004)
10. Fellbaum, C.: WordNet – an electronic lexical database. MIT Press, Cambridge (1998)
11. Florian, R., Cucerzan, S., Schafer, C., Yarowsky, D.: Combining classifiers for word sense disambiguation. Natural Language Engineering 8(4), 327–341 (2002)
12. Gale, W., Church, K., Yarowsky, D.: Estimating upper and lower bounds on the performance of word-sense disambiguation programs. In: Proc. of the ACL 1992, pp. 249–256 (1992)
13. Gonzalo, J., Verdejo, F., Chugur, I.: Indexing with wordnet synsets can improve text retrieval. In: Proc. of the COLING/ACL Workshop on Usage of WordNet for NLP (1998)
14. Hoste, V., Daelemans, W., Hendrickx, I., van den Bosch, A.: Evaluating the results of a memory-based word-expert approach to unrestricted word sense disambiguation. In: Proc. of the ACL Workshop on Word Sense Disambiguation (2002)
15. Ide, N., Veronis, J.: Word sense disambiguation: the state of the art. Computational Linguistics 24(1), 1–40 (1998)
16. Jeh, G., Widom, J.: Simrank: a measure of structural-context similarity. In: Proc. of KDD, pp. 538–543 (2002)
17. Kleinberg, J.: Authoritative sources in a hyperlinked environment. Journal of the ACM 46(5), 604–632 (1999)
18. Kohomban, U., Lee, W.: Learning semantic classes for word sense disambiguation. In: Proc. of ACL, pp. 34–41 (2005)
19. Lesk, M.: Automated sense disambiguation using machine-readable dictionaries: How to tell a pine cone from an ice cream cone. In: Proc. of the SIGDOC Conference, pp. 24–26 (1986)

20. Mavroeidis, D., Tsatsaronis, G., Vazirgiannis, M., Theobald, M., Weikum, G.: Word sense disambiguation for exploiting hierarchical thesauri in text classification. In: Jorge, A.M., Torgo, L., Brazdil, P.B., Camacho, R., Gama, J. (eds.) PKDD 2005. LNCS (LNAI), vol. 3721, pp. 181–192. Springer, Heidelberg (2005)
21. Mihalcea, R.: Word sense disambiguation with pattern learning and automatic feature selection. Natural Language Engineering 1(1), 1–15 (2002)
22. Mihalcea, R.: Unsupervised large-vocabulary word sense disambiguation with graph-based algorithms for sequence data labeling. In: HLT (2005)
23. Mihalcea, R., Csomai, A.: Senselearner: Word sense disambiguation for all words in unrestricted text. In: Proc. of ACL, pp. 53–56 (2005)
24. Mihalcea, R., Tarau, P., Figa, E.: Pagerank on semantic networks with application to word sense disambiguation. In: Proc. of COLING (2004)
25. Moldovan, D., Rus, V.: Logic form transformation of wordnet and its applicability to question answering. In: Proc. of ACL, pp. 394–401 (2001)
26. Navigli, R.: Online word sense disambiguation with structural semantic interconnections. In: Proc. of EACL (2006)
27. Navigli, R.: A structural approach to the automatic adjudication of word sense disagreements. Natural Language Engineering 14(4), 547–573 (2008)
28. Navigli, R.: Word sense disambiguation: A survey. ACM Computing Surveys 41(2), Article 10 (2009)
29. Navigli, R., Lapata, M.: Graph connectivity measures for unsupervised word sense disambiguation. In: Proc. of IJCAI, pp. 1683–1688 (2007)
30. Palmer, M., Dang, H., Fellbaum, C.: Making fine-grained and coarse-grained sense distinctions, both manually and automatically. Journal of Natural Language Engineering 13(2), 137–163 (2007)
31. Palmer, M., Fellbaum, C., Cotton, S.: English tasks: All-words and verb lexical sample. In: Proc. of Senseval-2, pp. 21–24 (2001)
32. Patwardhan, S., Banerjee, S., Pedersen, T.: Using measures of semantic relatedness for word sense disambiguation. In: Gelbukh, A. (ed.) CICLing 2003. LNCS, vol. 2588, pp. 241–257. Springer, Heidelberg (2003)
33. Pedersen, T.: A simple approach to building ensembles of naive bayesian classifiers for word sense disambiguation. In: Proc. of NAACL, pp. 63–69 (2000)
34. Pedersen, T., Kolhatkar, V.: WordNet:: SenseRelate:: AllWords - A Broad Coverage Word Sense Tagger that Maximimizes Semantic Relatedness. In: Proc. of NAACL/HLT, pp. 17–20 (2009)
35. Sinha, R., Mihalcea, R.: Unsupervised graph-based word sense disambiguation using measures of word semantic similarity. In: Proc. of ICSC (2007)
36. Snyder, B., Palmer, M.: The english all-words task. In: Proc. of Senseval-3, pp. 41–43 (2004)
37. Sussna, M.: Word sense disambiguation for free-text indexing using a massive semantic network. In: Proc. of CIKM (1993)
38. Tsatsaronis, G., Vazirgiannis, M., Androutsopoulos, I.: Word sense disambiguation with spreading activation networks generated from thesauri. In: Proc. of IJCAI, pp. 1725–1730 (2007)
39. Veronis, J., Ide, N.: Word sense disambiguation with very large neural networks extracted from machine readable dictionaries. In: Proc. of COLING, pp. 389–394 (1990)
40. Wu, D., Su, W., Carpuat, M.: A kernel pca method for superior word sense disambiguation. In: Proc. of ACL, pp. 637–644 (2004)
41. Yarowsky, D.: Word-sense disambiguation using statistical models of roget's categories trained on large corpora. In: Proceedings of COLING, pp. 454–460 (1992)
42. Zhao, P., Han, J., Sun, Y.: P-Rank: a comprehensive structural similarity measure over information networks. In: Proc. of CIKM, pp. 553–562 (2009)

A Case Study of Using Web Search Statistics: Case Restoration

Silviu Cucerzan

Microsoft Research
1 Microsoft Way, Redmond, WA 98052, USA
silviu@microsoft.com

Abstract. We investigate the use of Web search engine statistics for the task of case restoration. Because most engines are case insensitive, an approach based on search hit counts, as employed in previous work in natural language ambiguity resolution, is not applicable for this task. Consequently, we study the use of statistics computed from the snippets generated by a Web search engine, and we show that such statistics can achieve performance similar to corpus-based approaches. We also note that the top few results returned by a search engine may not the most representative for modeling phenomena in a language.

1 Introduction

This work has a dual goal: to address the task of case restoration via Web n-gram statistics and to investigate a framework of using Web search engines for NLP tasks.

1.1 Case Restoration

Case restoration (also known as *truecasing*) is a lexical disambiguation task that addresses the problem of adding or restoring capitalization information to a text that misses or has inconsistent such information.

This task is specific to languages that employ the Latin alphabet. While Romans only used uppercase letter forms, languages employing the Latin alphabet nowadays also use lowercase forms, which have been added to the alphabet in the Middle Ages. The vast majority of such languages capitalize the first letter of words in proper nouns and the first letter of the first word in a sentence. Additionally, there exist many other language-dependent capitalization rules. For example, all nouns are capitalized regardless of their position in the sentence in German, the days of the week are capitalized in English (e.g., "Thursday"), but not in French (e.g., "jeudi"), the names of languages are capitalized in English (e.g., "Japanese"), but not in Romanian (e.g., "japoneza"), etc. When such orthographic rules exist and are followed consistently, they can be successfully incorporated directly in a rule-based truecasing system or learned by using frequency-based estimates (such as MLE) from a large corpus.

Nevertheless, for many words, either no clearly specified capitalization rules exist or rules are applied inconsistently in practice (e.g., the capitalization of organization names, titles, or the first word in the document body differs widely across corpora).

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 199–211, 2010.

Furthermore, there are numerous *polysemous words*, which account for multiple meanings, that must be capitalized differently based on the intended meaning. For example, "us" and "aids" should be in lowercase or in uppercase in the following two Web text fragments about the AIDS crisis in Somalia, depending on their meaning (the casing information was removed to mimic the input to a case restoration system):

- *"it strikes us that the number of children orphaned by aids since the beginning of epidemic has exceeded 11 million* […]*"*;
- *"the annual aids for somalia has not exceeded us $ 50 million, according to un sources in nairobi* […]*"*[1]

With the modern information explosion and the widespread use of acronyms (e.g., "CICLing", "TidBITS"), technical terms (e.g., "HttpWebRequest"), catchy names ("eBay", "iPod"), foreign words (e.g., "Jade de Lugo"), references (e.g., "In Proc. of CoNLL 2001"), and hyperlinks (e.g., "Vote: Pullout an option?"), the body of text available in one language nowadays suffers of numerous capitalization irregularities.

As observed by Lița et al. [10], truecasing is an extremely useful tool in cross-corpora normalization, in order to produce consistent, case-sensitive, text information, which can be further used in information retrieval and information extraction tasks. One particular casing problem encountered by these tasks for most languages/texts is that the first word in each sentence is uppercased due to orthographic rules and not because of semantics. While such orthographic rules help humans and NLP systems in identifying sentence boundaries (e.g., [13], [15]), they introduce uncertainty on a different level. For example, Church [3] notes that "proper nouns and capitalized words are particularly problematic" in part-of-speech tagging and noun-phrase chunking. Many other NLP applications, such as named-entity recognition and machine translation, benefit from recovering the true case of the first word in the sentence once the text is split into sentences. For example, the disambiguation of instances of words such as "Hair" and "Traffic" (which can either be common nouns or refer to the Broadway show, respectively the 1970s rock band) occurring in the first position of a sentence, is simplified when true case information is available.

Mikeev [12] successfully addressed the problem of recovering true case information for words in positions where they must be capitalized (98.5% accuracy on a test set from The New York Times 1996 corpus) by a system based on the observation that "ambiguous things are usually unambiguously introduced at least once in text unless they are part of common knowledge". Prior to Mikeev's work, Yarowsky [19] trained a Bayesian model of context on 400 occurrences of the words "turkey" and "Turkey" (200 for each) drawn from the Associated Press newswire and tested it on an additional 50 examples of each, obtaining 100% accuracy. For other examples, Yarowsky reported that the typical performance was generally close to 90%.

Church [4] investigated the effect of text normalization, including collapsing of all case forms to one form, in information retrieval, but the findings on whether case information (as existing in the current corpora) helps were inconclusive.

[1] Note that "aids" is improperly used in plural form in this Web example. Truecasing such misspellings (which are relatively frequent in Web documents) increases the difficulty of the task, and makes it especially challenging for systems trained on a fixed trusted corpus.

Liță et al. [10] implemented a statistical, language-model-based truecaser, trained on news text, which achieved an accuracy of about 98% on news articles, and 96% on an English set of human translation of Chinese news articles. They showed that using such a system to preprocess the input or postprocess the output of various systems can substantially improve their performance: 26% F-measure improvement for named entity recognition, a factor of 8 in automatic content extraction, and a relative BLUE score improvement of 80.2% in machine translation.

1.2 Using Web Statistics in NLP Tasks

Previous work in natural language ambiguity resolution has shown that Web frequencies of word n-grams as estimated by search engines can be used to overcome data sparseness and provide a reliable general source of lexical and syntactic information. For example, Grefenstette [8] used Web frequencies to obtain counts for candidate translations for word compounds from English to French. Zhu and Rosenfeld [20] analyzed the use of language models that interpolate n-gram corpus frequencies and n-gram Web frequencies in speech recognition. Keller et al. [9] showed that there is a strong correlation between Web frequencies as estimated by search engines and corpus frequencies of adjective-noun, noun-noun, and verb-object pairs. Cucerzan and Yarowsky [5] observed that a search engine's estimated number of document hits for word bigrams can be used successfully to predict grammatical gender of nouns in languages that have this grammatical subclass.

Using similar Web statistics for truecasing is not feasible because the major Web search engines (Google, Yahoo, Bing, and Ask) employ case-insensitive indexes and thus, they cannot provide estimates for different case forms of the words in given contexts (and even independent of context). One possible solution to this problem, which we propose in this work and describe in detail in Section 3, is to extract statistics from the snippets returned by a search engine for queries that contain the words of interest.[2] The technique of employing a ranker and a snippet generation system of a search engine to retrieve and compile text data for computing occurrence statistics was previously employed successfully in question answering (e.g., [1], [14]). Mihalcea [11] also employed search engine snippets to build a corpus annotated with sense tags for the word sense disambiguation task, while Sumita and Sugaya [17] used search engine snippets to build contextual models for acronym disambiguation.

2 Problem Formulation and Methodology

We distinguish four types of case representations of words: all lowercase, first letter in uppercase, all uppercase, and mixed case (the latter accounting for $2^n - 3$ possible case representations of a word of length n). Thus, we can regard the task of truecasing as a classification problem, in which we have to label each word of length n in a text with one of 2^n labels. In practice, only one or two mixed case forms cover the overwhelming majority of occurrences of a word (as shown further in Section 3). In this work, we discard from the candidate space all case labels that are not present in the snippets returned by the Web search engine.

[2] The snippets are short paragraphs of text, typically two or three sentence fragments from the documents retrieved, that contain instances of the queried words.

We employ the typical scenario used in automatic speech recognition and machine translation, by presuming that the input text is in lowercase and we are tasked with restoring its case information. Note that when the input text is already cased but the case is unreliable, the input capitalization and the estimated defect rate could be used to determine a prior. However, such an approach was beyond the scope of this paper. Here is an overview of the approach investigated:

Step 1: Tokenize the text. Because our goal is to build a system that works in most languages that use the Latin alphabet, we only rely on punctuation and spaces for tokenization. The exceptions to the rule are periods and apostrophes, which we do not regard as separators and remove only when present at the beginning or end of the words (in order not to split acronyms, possessives, and contractions).

Step 2: To predict the case form of each token, extract the n-grams of length three or less containing it and gather search result snippets returned by a Web search engine for these n-grams submitted as quoted queries. We retrieve as many as 200 Web results per query, from which we extract case statistics for the target word.

Step 3: Finally, use the case statistics collected to decide which case form to associate with each token in the input text in a maximum likelihood framework. Variations on employing the search-snippet case statistics are discussed in Section 3.

For the experiments presented in this paper, we employed Microsoft's Bing search engine, which we were granted permission to query automatically. While the snippets generated by Bing typically contain the queried words, they do not necessarily contain the whole query (i.e. the n-gram) as a substring. Thus, we may not be able to find enough snippet instances of the target word in exactly the same n-gram context as that in the input text. Based on the "one sense per discourse" tendency noted by Gale et al. [7] and successfully employed by Mikeev [12] in a truecasing-related task, we assume that the instances of the target word in a snippet are representative of the case form(s) of the target word in the whole Web document that matched the query and from which the snippet was generated by the search engine. Thus, once we retrieve the snippets for an n-gram query containing a target word, we extract all instances of the target word from the snippet regardless of context, except when they follow a period (in such cases, it is likely that the instance represents the first word of a sentence and is capitalized due to orthographic rules). If a snippet contains multiple instances of the target word and those have more than one casing form then we can either discard the snippet or add the frequency of each case form to the total corresponding frequency. These strategies showed no significant difference in performance on an English development set; thus, we employed the latter, as being both simpler in implementation and more similar to accounting for all instances of words in a text corpus (regardless of paragraph and/or article boundaries).

Table 1 provides some intuition about the process we are employing. It shows statistics extracted from the Web by our system for the correct capitalization of the two sentences discussed in Section 1.1 when using the top results returned by the Bing engine. For each input word (e.g., "un") in its context, it shows the fraction of instances that had the correct capitalization (e.g., "UN" - 68/70) in the top *N* snippets obtained by querying the trigram centered in the target word (e.g., "to un sources").

Table 1. Case statistics (number of occurrences with the correct case for the target sentence / total word occurrences) extracted from the snippets of the top 25, 50, and 100 results returned by a Web search engine for the example sentences from Section 1.1.

Word	Top 25	Top 50	Top 100
it	48/53	102/111	195/211
strikes	25/25	50/50	100/100
us	40/40	74/74	149/149
that	31/32	63/64	134/136
the	102/102	190/190	336/337
number	29/42	63/80	104/131
of	56/56	99/99	199/199
children	22/30	48/64	111/134
orphaned	14/14	33/33	81/81
by	19/26	50/60	117/130
AIDS	39/39	80/80	142/146
since	16/16	34/34	74/74
the	68/68	127/127	278/282
beginning	13/14	21/23	54/59
of	67/67	141/141	183/183
epidemic	26/27	44/45	44/45
has	10/10	10/10	10/10
exceeded	23/23	50/50	92/92
11	-	-	-
million	51/51	89/91	192/194

Word	Top 25	Top 50	Top 100
the	74/74	138/138	274/280
annual	25/26	46/54	96/105
aids	4/4	4/4	4/4
for	2/2	2/2	2/2
Somalia	30/30	61/61	117/117
has	33/33	62/62	132/132
not	24/24	52/52	118/118
exceeded	21/21	21/21	21/21
US	33/39	61/74	121/146
$	-	-	-
10	-	-	-
million	29/31	67/69	132/134
according	22/26	43/51	79/100
to	33/35	82/84	192/195
UN	32/32	68/70	139/144
sources	22/22	47/47	95/96
in	77/81	132/138	132/138
Nairobi	28/28	62/62	114/114

3 System Architecture and Development

In this work, we employed the AQUAINT corpus both because previous work has been done on this corpus and because it is not indexed by the search engine used in our experiments. The development set comprises of the first four news stories from the New York Times data for January 1, 2000, as ordered in AQUAINT, which contain a total of 2,348 words. Based on the empirical evidence obtained on these data for various parameter settings, we chose the default parameter values for the final system.

Since we are concerned with measuring accuracy, we force the system to make exactly one capitalization choice for each word in the text. To break ties when necessary, we simply pick one of the case forms at random.

As truth, we used the original capitalization of the text. This strategy has the advantage that no annotation effort is needed. Its disadvantage is that stylistic choices specific to the corpus from which the gold standard was selected are more difficult to predict when using Web statistics than when training on text from the same corpus. In an attempt to quantify the impact of the corpus casing conventions on the performance of truecasing systems, we compared the accuracy obtained by a system trained on data from the same corpus as the gold standard set (Table 4) with the accuracy of a system trained on a different corpus (Table 4).

Liță et al. [10] observed that approximately 88% of the word types in the vocabulary of a large news corpus (AQUAINT) occur with only one capitalization form. Therefore, a unigram model trained on such a corpus that predicts the most frequent

capitalization form seen in the training is likely to obtain a relatively high accuracy on other news text. Liţă et al. reported for this approach accuracy numbers between 93% and 96% on the four test sets they used.

In the training set employed in our study (AQUAINT, NYT July-December 1999), 85.5% of the word types have only one case form, 11% have two case forms, and 3.5% have three case forms. Less than 0.1% of the word types have 4 or more case forms (132 word types have 4 case forms and 5 word types have 5 case forms). However, by-token numbers are radically different from the above, with the dominating class being three capitalization forms, as shown in Figure 1. The main cause for this change is represented by named entities that contain high-frequency words (e.g., "The Spy Who Shagged Me", "A WALK IN THE WOODS", and "The Boston Globe").

The by-token statistics from the training set are in fact very similar to those obtained from the top 200 snippets returned by the Bing search engine for each word in the original text, which are shown in Figure 2. The average number of case forms retrieved from the Web search results per token is three. On the extremes, one word, accounting for one token, is not found at all on the Web (the misspelling "fridayoccurred") and one word appears with no less than seven different case forms in the top 200 snippets ("Latoya", "latoya", "LaToya", "LatoYa", "LaToYa", "LATOYA", and "LaTOYA"). Since we cannot obtain counts for every possible case form by using the top results generated by a Web search engine, we also employed the Google n-gram corpus [1], which was derived from 1 terabyte of publicly accessible Web pages. The by-token statistics derived from these data are shown in Figure 3. Some of the words have close to 30 or more case forms with frequencies greater than the 200 cut-off employed for unigrams in the Google data (shown in Table 2). Note that the percentage of out-of-vocabulary tokens is much larger when employing the Web crawl data (0.47%) than when employing search snippets (0.04%) because our tokenization method for possessives and contractions is different from that employed by Google, and some tokens had fewer than 200 occurrences in the crawled data (e.g., "Edneisha").

The disadvantage of using unigram statistics for truecasing becomes apparent when analyzing the capitalization of words such as "new" and "big", which are most frequently used as adjectives and adverbs, but are also part of popular proper nouns such as New York and Big Apple. Higher order n-gram models, which capture the immediate context in which a word is used, are needed. However, such models are susceptible to data sparseness, which is a significant problem even for the Web-based system (the rightmost column in Table 3(a) shows the number of trigrams in the development set with no search results). To overcome sparsity, we use a back-off model, in which trigram statistics are employed in conjunction with bigram and unigram statistics.

Table 2. The frequencies (in thousands) of different case forms for four words from the development set as observed in the Google crawled data. The last row shows the number of mixed case forms that occurred at least 200 times in the crawled data and their aggregated frequency.

about	791,445	people	382,196	Black	126,506	friends	69,116
About	413,227	People	92,625	black	105,075	Friends	39,369
ABOUT	22,037	PEOPLE	5,483	BLACK	13,079	FRIENDS	2,248
26 forms	54	27 forms	20	27 forms	71	28 forms	35

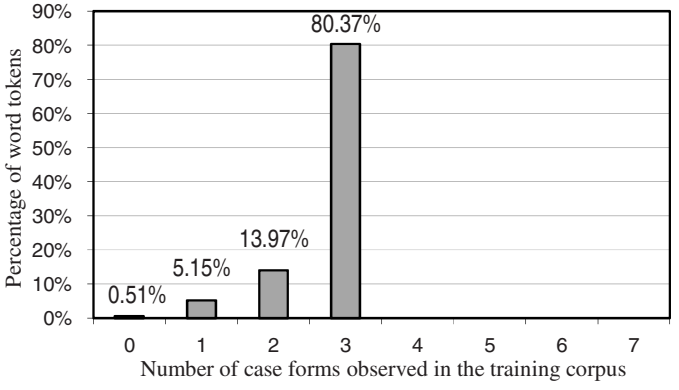


Fig. 1. The distribution of number of case forms observed in the AQUAINT training set for the tokens in the development set

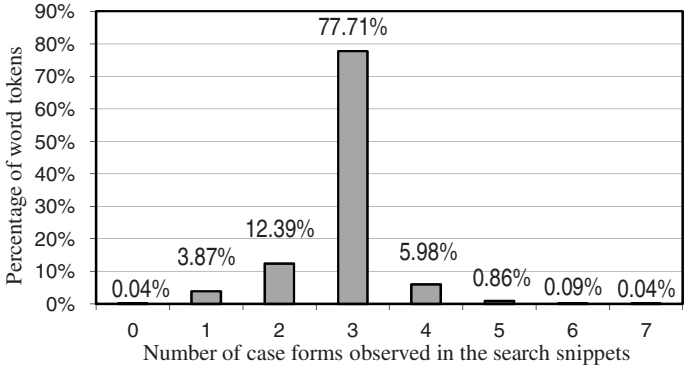


Fig. 2. The distribution of number of case forms observed in the top 200 Web search results for the tokens in the development set

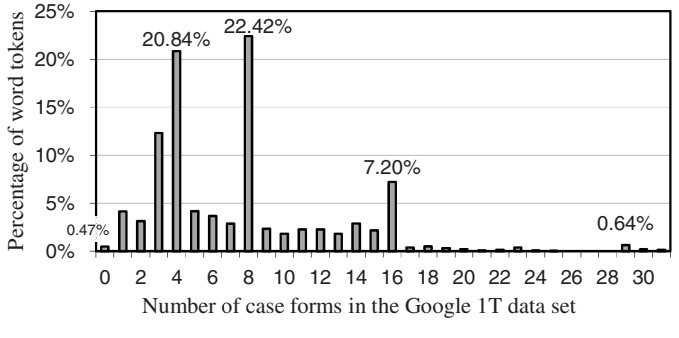


Fig. 3. The distribution of number of case forms observed in Google's Web crawled data for the tokens in the development set (forms with frequencies lower than 200 were discarded)

Note that there are three trigrams that contain a word w_i in a text $w_0 \ldots w_{i-2} w_{i-1} w_i w_{i+1} w_{i+2} \ldots w_n$, with the exception of values $i \in \{0, 1, n\text{-}1, n\}$. We refer to $w_{i-2} w_{i-1} w_i$ as the *left trigram* for w_i, $w_{i-1} w_i w_{i+1}$ as the *middle trigram*, and $w_i w_{i+1} w_{i+2}$ as the *right trigram* for the target word w_i.

We tried three different ways of combining the trigram type statistics to hypothesize the capitalization of each word: *Max* chooses the capitalization provided by the trigram type that has the highest maximum probability estimate, *Vote* chooses the capitalization on which at least two of the trigram types agree, while *Sum* first sums all three trigram statistics and then picks the maximum. Tie-breaking for each method is done by using Sum first and Max second if necessary. As back-off, we use the sum of bigram casing statistics, followed by unigram statistics. When no trigram, bigram, or unigram evidence is retrieved, the system hypothesizes first letter in uppercase.

Table 3 shows the results obtained for each individual trigram type (with back-off to bigrams and unigrams) and the three combination methods on the development set. These results are almost identical (no statistically significant difference except for Sum) to those obtained by employing n-gram statistics extracted from the news articles corresponding to the previous six months in the same corpus (NYT, July 1 to December 31, 1999), which are shown Table 4, and much higher than the accuracies obtained by employing statistics from an MSNBC news corpus of similar size from 2009, which was chosen to match as closely as possible the dates for the Web-search-based experiments (shown in Table 5). The differences between the numbers reported in Tables Table 4 and Table 5 can be interpreted as estimates of error rates induced by the differences in vocabularies and capitalization conventions across corpora.

Table 3. Accuracy on the development set when using quoted trigrams as search queries. (a) Left, Middle, and Right represent the position of the context relative to the target word in the trigram; (b) Max, Vote and Sum represent the class combinations of the three trigram results.

(a)

Trigram Type	Accuracy	Total errors	No-search-result trigrams
Left	94.68%	120	45
Middle	95.25%	106	48
Right	93.81%	144	76

(b)

Combination Type	Accuracy	Total errors	No-search-result trigrams
Max	97.12%	66	18
Vote	96.82%	84	18
Sum	**97.46%**	61	18

Table 4. Accuracy obtained on the dev test set by employing n-gram statistics from the same news corpus (training data: July-December 1999, dev test: January 2000)

Trigram Type			Combination Type		
Left	Middle	Right	Max	Vote	Sum
94.93%	95.40%	93.73%	97.27%	96.85%	96.89%

Table 5. Accuracy obtained on the dev test set by employing n-gram statistics from an MSNBC corpus (training data: 2009, dev test: January 2000)

Trigram Type			Combination Type		
Left	Middle	Right	Max	Vote	Sum
89.95%	90.47%	88.85%	95.88%	92.58%	95.36%

Table 6. Accuracy obtained on the dev test set by employing n-gram statistics from the Google 1T set (training data: crawled in 2006, dev test: January 2000)

Trigram Type			Combination Type		
Left	Middle	Right	Max	Vote	Sum
91.90%	92.33%	90.11%	95.95%	94.16%	94.50%

Table 7. The accuracy of the system when using the casing statistics from the snippets of the top *N* search results, as well as when using 10 random results from the top 200

Top *N*	1	2	5	10	20	50	100	200	Rand 10
Acc	78.89	85.02	90.90	94.12	96.11	96.61	97.21	**97.46**	97.21

We also ran the same system by using the Google 1T trigram, bigram, and unigram statistics. Unfortunately, the different tokenization rules employed made difficult a direct comparison (no less than 251 tokens in the dev set were missing trigram information). We tried to address some of these differences (for possessives and contractions) by employing higher order n-grams, but other tokenization differences may have impaired the results. Even under these circumstances, the accuracies obtained, shown in Table 6, are very promising and warrant further investigation.

Table 7 shows the dependence of accuracy on the number of Web search results employed to extract n-gram statistics from. Note that performance increases steadily with the number of search snippets employed, and that using statistics from the first few search snippets is much worse than using statistics from the snippets of a random set of 10 results from the top 200. In fact, randomly choosing 10 search results achieves performance identical to that of the top 100 search results.

4 Final Evaluation

We performed the final evaluation of the system on a test set comprising a total of 11k words from the top four articles corresponding to three randomly selected days from the NY Times section of the AQUAINT corpus, one day for each of the years covered by this corpus (1998-2000): June 18, 1998, April 23, 1999, and May 24, 2000. We compared the results obtained by using Web statistics with those obtained by a system trained on all articles of the AQUAINT corpus from July 1 to December 31, 1999, from which we automatically removed XML tags and all text lines that did not contain at least one lowercase character (typically, such lines represent titles or editorial information), totaling over 50 million words. While we did not have access to the data sets used by Liță et.al [10], we employed data from the same corpus in an attempt to minimize the discrepancies between the performance numbers reported.

Table 8 shows the results obtained on the test set by trigram type and by combination employed for both systems. The combination for the Web-search-based system that was determined to be the best on the development set (Sum) also obtains the best performance on the test set, and is significantly better than the corresponding system that employs AQUAINT n-gram statistics. Figure 4 shows some insights into the performance of the Web-search-based system for various sizes of the result sets retrieved by the search engine employed. In general, the more results the engine retrieves, the more reliable the statistics extracted from the result snippets are.

Table 8. Accuracies on the test set of the systems that employ n-grams statistics from the same corpus and the one based on Web-search statistics

	Trigram Type			Combination Type		
	Left	Middle	Right	Max	Vote	**Sum**
Same corpus	94.05%	94.90%	92.74%	98.15%	96.29%	97.76%
Web search	94.51%	93.20%	93.04%	98.07%	96.60%	**98.22%**

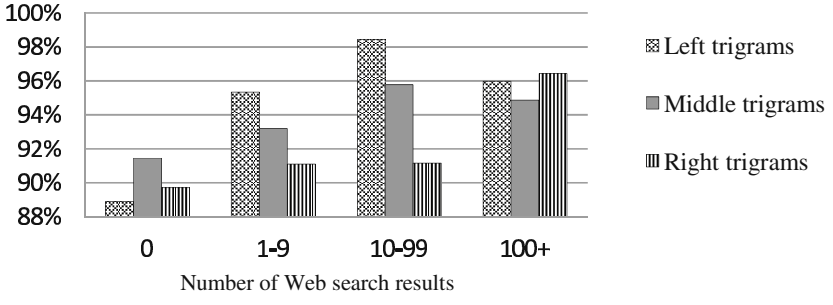


Fig. 4. The dependency of Accuracy of the Web trigram (with bigram and unigram back-off) on number of Web search results for each trigram type

5 Other Findings

While investigating the use of search results for the truecasing task, we were also able to compute the Web coverage of n-grams in texts and compare them with previous results reported by Zhu and Rosenfeld [20]. In their work, Zhu and Rosenfeld used a test set of 24 fresh sentences randomly chosen from 4 Web news sources (i.e. they presumed that the search engines had not indexed the Web pages containing those sentences). Of the 327 word types in those sentences, 8 were not previously seen in a 103-million-word Broadcast News corpus, while all of them were present in the indexes of AltaVista, Lycos, and FAST. 5 out of 462 word bigrams (1.1%) and 46 out of 453 word trigrams (10.2%) were not found in the search engines' indexes. Figure 5 shows the n-gram coverage provided by the Bing search engine for the English test set. All words are found in the index of the Web search engine. Bigram coverage is consistent with the previously reported numbers, while trigram coverage improved by close to 2.5% absolute value. The trigram Web coverage is substantially higher than

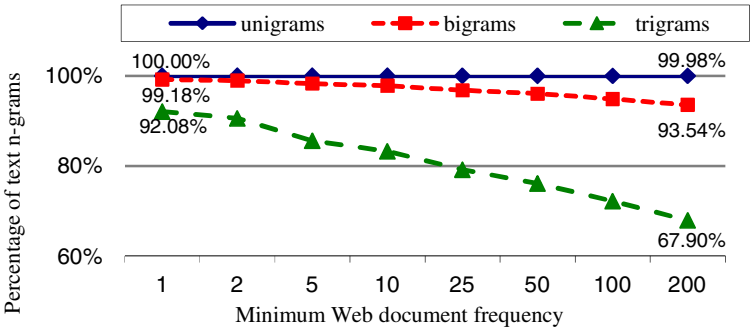


Fig. 5. Percentage of unigrams, bigrams, and trigrams from news text in three languages with different Web-based document frequency levels (i.e., the Web search engine employed retrieved for the quoted n-grams at least the number of documents specified on the Ox axis)

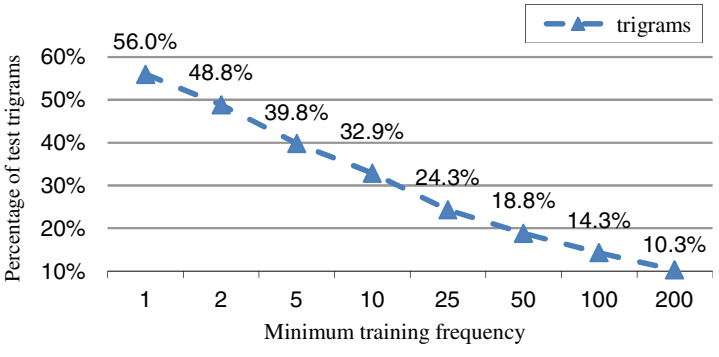


Fig. 6. Percentage of trigrams in the AQUAINT test set that have at least a certain number of occurrences in the training data set from the same corpus (NYT July-December 1999)

that provided by the AQUAINT subset employed for training (plotted in Figure 6). In fact, for 67.9% of the trigrams, the search engine retrieves at least 200 Web results, while only 56% of the trigrams appear at least once in the AQUAINT training set.

6 Conclusion

We investigated the use of statistics gathered from the snippets of a Web search engine for the case restoration task. We analyzed several ways of employing such statistics and showed that they can compete successfully with corpus-based approaches.

In general, Web statistics present the advantage that they allow the normalization of any text collection using the Web as a mediator. Furthermore, out-of-vocabulary words are extremely rare for such Web-based systems regardless of the age of the collection, and there is no practical need to address them in a sophisticated manner.

It was of no surprise that the performance of the system constantly improved with increasing the number of search results used. Nonetheless, a very interesting outcome

of our study is that the performance obtained by using the top 10 search results was much poorer than the performance obtained by using 10 random results out of the top 200. We believe that this bias is due to the dynamic ranking component of a Web search engine's favoring documents in which the query words play special roles (whether they occur more often than expected, they are in the anchor text of links to those documents, they are part of titles, or have other particular positional properties). Such behavior should be accounted for by any NLP system that relies on statistics extracted from the top results provided by a Web search engine.

Acknowledgement

The author wishes to thank Robert Ragno for helping with building the querying and caching infrastructure and the anonymous reviewers for kindly suggesting ways to improve the paper, including reporting experiments on the Google 1T 5-gram corpus.

References

1. Brants, T., Franz, A.: Web 1T 5-gram Version 1. Linguistic Data Consortium, Catalog ID: LDC2006T13 (2006)
2. Brill, E., Lin, J., Banko, M., Dumais, S., Ng, A.: Data-intensive question answering. In: Proceedings of TREC 2001, pp. 393–400 (2001)
3. Church, K.: A stochastic parts program and noun phrase parser for unrestricted text. In: Proceedings of ANLP 1988, pp. 136–143 (1988)
4. Church, K.: One Term or Two? In: Proceedings of SIGIR 1995, pp. 310–318 (1995)
5. Cucerzan, S., Yarowsky, D.: Minimally Supervised Induction of Grammatical Gender. In: Proceedings of HLT/NAACL 2003, pp. 40–47 (2003)
6. Gale, W.A., Church, K., Yarowsky, D.: Discrimination Decisions for 100,000-Dimensional Spaces. In: Current Issues in Computational Linguistics, pp. 429–450 (1987)
7. Gale, W.A., Church, K., Yarowsky, D.: One Sense per Discourse. In: Proceedings of the 4th DARPA Speech and NL Workshop, pp. 233–237 (1992)
8. Grefenstette, G.: The World Wide Web as a Resource for Example-based Machine Translation Tasks. In: Proceedings of the ASLIB Conference on Translating and Computer (1998)
9. Keller, F., Lapata, M., Ouriopina, O.: Using the Web to overcome data sparseness. In: Proceedings of EMNLP 2002, pp. 230–237 (2002)
10. Liţă, L.V., Ittycheriah, A., Roukos, S., Kambhatla, N.: tRuEcasIng. In: Proceedings of ACL 2003, pp. 152–159 (2003)
11. Mihalcea, R.: Bootstrapping Large Sense Tagged Corpora. In: Proceedings of LREC 2002 (2002), http://www.cs.unt.edu/~rada/papers/mihalcea.lrec02.pdf
12. Mikev, A.: A Knowledge-free Method for Capitalized Word Disambiguation. In: Proceedings of ACL 1999, pp. 159–166 (1999)
13. Palmer, D.D., Hearst, M.A.: Adaptive Sentence Boundary Disambiguation. In: Proceedings of ANLP 1994, pp. 78–83 (1994)
14. Radev, D., Qi, H., Zheng, Z., Blair-Goldstein, S., Zhang, Z., Fan, W., Prager, J.: Mining the Web for Answers to Natural Language Questions. In: Proceedings of CIKM 2001, pp. 143–150 (2001)

15. Reynar, J.C., Ratnaparkhi, A.: A Maximum Entropy Approach to Identifying Sentence Boundaries. In: Proceedings of ANLP 1997, pp. 16–19 (1997)
16. Sampson, G.: Writing systems: A Linguistic Introduction. Stanford University Press (1985)
17. Sumita, E., Sugaya, F.: Using the Web to Disambiguate Acronyms. In: Proceedings of HLT/NAACL 2006, pp. 161–164 (2006)
18. Tang, J., Li, H., Cao, Y., Tang, Z.: Email Data Cleaning. In: Proceedings of KDD 2005, pp. 489–498 (2005)
19. Yarowsky, D.: Decision Lists for Lexical Ambiguity Resolution: Application to Accent Restoration in Spanish and French. In: Proceedings of ACL 1994, pp. 88–95 (1994)
20. Zhu, X., Rosenfeld, R.: Improving Trigram Language Modeling with the World Wide Web. In: Proceedings of ICASSP 2001, pp. 592–597 (2001)

A Named Entity Extraction Using Word Information Repeatedly Collected from Unlabeled Data

Tomoya Iwakura

Fujitsu Laboratories Ltd.
1-1, Kamikodanaka 4-chome, Nakahara-ku, Kawasaki 211-8588, Japan
iwakura.tomoya@jp.fujitsu.com

Abstract. This paper proposes a method for Named Entity (NE) extraction using NE-related labels of words repeatedly collected from unlabeled data. NE-related labels of words are candidate NE classes of each word, NE classes of co-occurring words of each word, and so on. To collect NE-related labels of words, we extract NEs from unlabeled data with an NE extractor. Then we collect NE-related labels of words from the extraction results. We create a new NE extractor using the NE-related labels of each word as new features. The new NE extractor is used to collect new NE-related labels of words. The experimental results using IREX data set for Japanese NE extraction show that our method contributes improved accuracy.

1 Introduction

Named Entity (NE) extraction aims to extract proper nouns and numerical expressions in text, such as persons, locations, organizations, dates, times, and so on. NE extraction is one of the basic technologies used in text processing such as information extraction, question answering, and so on.

To implement NE extractors, semi-supervised-based methods have recently been widely applied [1–6]. These methods use several different types of information obtained from unlabeled data, such as word clusters [1, 2, 4], phrase clusters [6], hyponymy relations extracted from Wikipedia [4], word dictionaries [7, 8], and the outputs of classifiers or parsers created from unlabeled data [3]. It is also possible to use unlabeled data in semi-supervised-learning algorithms [5]. These results have shown the utilization of unlabeled data can contribute to improved accuracy.

We propose a method for Named Entity (NE) extraction using NE-related labels of words collected from unlabeled data repeatedly. A set of NE-related labels of words is one type of automatically generated dictionary. NE-related labels of each word include candidate NE classes of each word, NE classes of co-occurring words of each word, and so on. We use the NE-related labels of words as new features for NE extraction.

To collect NE-related labels of words, we propose the following method. First, we extract NEs from unlabeled data by using an NE extractor, and collect NE-related labels of words from the NE extraction results. Then a new NE extractor using the NE-related labels of each word as features is created. The new NE extractor is used to collect new NE-related labels of words. By repeating this process several times, we expect to obtain better NE-related labels of words than those obtained by a one pass process approach.

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 212–223, 2010.

Table 1. NE examples defined by IREX committee

ARTIFACT	LOCATION	ORGANIZATION	PERSON
Nobel Prize in Chemistry	Japan	the Ministry of Foreign Affairs	Tarou Yamada

DATE	MONEY	PERCENT	TIME
May	100 JPY	100%	10:00 a.m.

2 Japanese Named Entity Extraction

This section describes our NE extraction method that combines both word-based and character-based NE extraction. We use the IREX Japanese NE extraction task [9] to evaluate our method. Table 1 lists the eight NE classes defined by the IREX committee.

2.1 Named Entity Extraction by Word-Unit Chunking

One of the problems for extracting NEs is that each NE consists of one or more words. To extract NEs, we have to identify word chunks with their NE classes. To identify word chunks, we use methods to annotate chunk tags to words. IOB1 [10], IOB2, IOE1, IOE2 [11] and Start/End (SE) [12] are proposed for chunk representation. We use SE chunk representation because a SE representation-based NE extractor shows the best performance among previous reports [13]. SE representation uses five tags which are S, B, I, E and O, for representing chunks. S means that the current word is a chunk consisting of only one word. B means the start of a chunk consisting of more than one word. E means the end of a chunk consisting of more than one word. I means the inside of a chunk consisting of more than two words. O means the outside of any chunk. The SE based NE label set for IREX task has $(8 \times 4) + 1 = 33$ labels.

For example, ” ⟨ORG⟩ 田中 (Tanaka) 使節 (mission) 団 (party) ⟨/ORG⟩ は (particle) ⟨LOC⟩ 日 (Japan) ⟨/LOC⟩ ⟨LOC⟩ 米 (U.S.A) ⟨/LOC⟩ 間 (between)...” is represented as follows by SE representation: ”田中 / ORG-B 使節 /ORG-I 団 /ORG-E は /O 日 /LOC-S 米 /LOC-S 間 /O ...”.

2.2 Named Entity Extraction by Word-Unit Chunking

We classify each word into one of the NE labels defined by the SE representation for extracting NEs. Our NE extractor uses the following features extracted from the current word, the preceding two words and the two succeeding words (5-word window). This setting has shown the best performance in Japanese NE experiments [13]. In the following explanations, we assume a sentence consisting of m words $\{w_1, ...,w_m\}$ $(0 < m)$.

– **Words**, **base-forms**, and **readings**: We use words tokenized with a morphological analyzer because Japanese language has no word boundary marker. We use ChaSen [1] as the morphological analyzer. When we classify the i-th word w_i $(1 \leq m)$ to one

[1] http://chasen-legacy.sourceforge.jp/

Table 2. Basic character type

Character type	Hiragana (Japanese syllabary characters), Katakana, Kanji (Chinese letter) Capital alphabet, Lower alphabet, number and Others.

of the NE labels, we use w_{i-2}, w_{i-1}, w_i, w_{i+1} and w_{i+2} as features. In addition, we use the base-forms and the readings of these five words annotated by ChaSen. Base-forms are representative expressions for conjugational words. As for the base-forms, if the base-form of each word is not equivalent to the word, we use the base-form of the word as a feature.

- **Part of speech (POS) tags**: We use the POS tags for each word annotated by ChaSen. Let $POS(w)$ be the POS tag of a word w. We use $POS(w_{i-1})$, $POS(w_{i-1})$, $POS(w_i)$, $POS(w_{i+1})$ and $POS(w_{i+2})$ as features. ChaSen uses the set of POS tags having at most four levels of subcategories. We use the top level of POS tags, and the POS tags consisting of all the levels of subcategories as the POS tag features.
- **Character types**: If a word consists of only one character, the character type is expressed by using the corresponding character types listed in Table 2. If a word consists of more than one character, the character type is expressed by using a combination of the basic character types listed in Table 2, such as Kanji-Hiragana. Let $CT(w)$ be the character type of a word w. We use $\{CT(wi_{i-2}), ... CT(w_{i+2})\}$ as features.
- **Characters in words**: If a word consists of more than one character, we use some of the characters of the word as features. Let $BC(w)$ and $EC(w)$ be the first character and the last character of a word w. We use $\{BC(w_{i_2}),..., BC(w_{i+2})\}$ and $\{EC(w_{i_2}),.., EC(w_{i+2})\}$ as features.
- **NE-related labels of words**: Let $NERL(w)$ be NE-related labels of a word w. We use $\{NERL(w_{i-2}), .., NERL(w_{i+2})\}$ as features. NE-related labels of words are described in section 3.

To distinguish the same features when they appear multiple times within a single 5-word window, features of each word are expressed with the relative position from the current word. For example, features of two preceding word from the current position i are expressed like "-2-th-word=w_{i-2}" and "-2-th-word-of-POS=$POS(w_{i-2})$".

2.3 Named Entity Extraction by Character-Unit Chunking

Japanese NEs sometimes include partial words. Partial words form the beginning or the end of NE chunks. Partial words also form whole NEs. For example, the "訪米 (visit U.S.A)" in "田中 (Tanaka) 使節 (mission) 団 (party) は (particle) 訪米 (visit U.S.A) (Tanaka mission party visit U.S.A.). " does not match with LOCATION "米 (U.S.A)" because this sentence is tokenized as "田中 (Tanaka)/ 使節 (mission) / 団 (party)/ は / 訪米 (visited U.S.A)", where "/" indicates a word boundary.

To extract Japanese NEs including partial words, we use a character-unit-chunking-based NE extraction algorithm [14, 15] following word-based NE extraction as in [7].

Word	POS	POS-fist-level	CharType	BC	EC	NE label
田中	Noun-Surname	Noun	Kanji+	田	中	B-ORG
使節	Noun-General	Noun	Kanji+	使	節	I-ORG
団	Noun-Suffix-General	Noun	Kanji			E-ORG
は	Particle-Case-General	Particle	Hiragana			O
訪米	Noun-Verb-Connection	Noun	Kanji+	訪	米	S-LOC

↓

Char	POS	CharType	NE label of word	Word	Word CharType	NE label
田	B-Noun-Surname	Kanji	B-B-ORG	B-田中	Kanji+	B-ORG
中	E-Noun-Surname	Kanji	E-B-ORG	E-田中	Kanji+	I-ORG
使	B-Noun-General	Kanji	B-I-ORG	B-使節	Kanji+	I-ORG
節	E-Noun-General	Kanji	E-I-ORG	E-使節	Kanji+	I-ORG
団	S-Noun-Suffix-General	Kanji	S-E-ORG	S-団	Kanji	E-ORG
は	S-Particle-Case-General	Hiragana	S-O	S-は	Hiragana	O
訪	B-Noun-Verb-Connection	Kanji	B-S-LOC	B-訪米	Kanji	O
米	E-Noun-Verb-Connection	Kanji	E-S-LOC	E-訪米	Kanji	S-LOC

Fig. 1. Feature expression for training: The top table is an example of a word unit chunking and the bottom table is a character unit chunking.

Figure 1 shows the examples of a word-unit-chunk representation in the top and a character-unit-chunk representation in the bottom.

We use the approach of [7] for selecting features. In addition to the features of the current position character, we use the features of the two preceding and two succeeding characters (5-character window). In the following explanations, we use a sentence consisting of n characters $\{c_1, ..., c_n\}$ $(0 < n)$.

- **Characters**: We use each character as a feature. When we classify the i-th character c_i $(1 \leq i \leq n)$, we use $\{c_{i-2}$,...,$c_{i+2}\}$ as features.
- **Character types**: We assign one of the character types listed in Table 2 to each character and use this as a feature. Let $CT(c)$ be the character type of a character c, then we use $\{CT(c_{i-2})$,...,$CT(c_{i+2})\}$ as features.
- **Words including characters**: We use the word including each character within the window size as a feature. Let $W(c_i)$ be the word including the i-th character c_i and $P(c_i)$ be the identifier that indicates the position where c_i appears in $W(c_i)$. We combine $W(c_i)$ and $P(c_i)$ to create a feature. $P(c_i)$ is one of the followings; B is for the character that is the beginning of a word, I is for each character that is in the inside of a word, E is for the character that is the end of a word, and S is for a character that is a word. For example, if "外務省 *(the Ministry of Foreign Affairs)* は *(particle)*" is segmented as "外務省 / は", then words including characters are follows; "W(外) = 外務省", "W(務) = 外務省, "W(省) = 外務省" and "W(は)=は". The identifiers that indicate positions where characters appear are follows; "P(外) =B", "P(務) = I", "P(省) = E" and "P(は)=S".
- **POS tags of words including characters**: Let $POS(W(c_i))$ be the POS tag of the word $W(c_i)$ including the i-th character c_i. We use the POS tags of words

including characters within window size as features. We express these features with the position identifier $P(c_i)$.

- **Character types of words including characters**: Let $CT(W(c_i))$ be the character type of the word including the i-th character c_i. We use $CT(W(c_{i-2}))$, $CT(W(c_{i-1}))$, $CT(W(c_i))$, $CT(W(c_{i+1}))$ and $CT(W(c_{i+2}))$ as features.
- **NE labels of words assigned by a word-unit NE extractor**: Let $NEL(W(c_i))$ be the NE label of the word including the i-th character c_i. We express this feature with the identifier $P(c_i)$ and NE label $NEL(W(c_i))$. In this experiment, we use "$P(c_{i-2}) - NEL(W(c_{i-2}))$", "$P(c_{i-1}) - NEL(W(c_{i-1}))$", "$P(c_i) - NEL(W(c_i))$", "$P(c_{i+1}) - NEL(W(c_{i+1}))$" and "$P(c_{i+2}) - NEL(W(c_{i+2}))$" as features.
- **NE labels of characters**: The NE labels of two preceding extraction results are used as features in the direction from the end to the beginning of each sentence. This setting has shown good performance in past experiments [14, 15]. Let $NEL(c)$ be the NE label assigned to character c by an NE extractor. In this experiment, we use $NEL(c_{i+1})$ and $NEL(c_{i+2})$ as features.

To distinguish the same features when they appear multiple times within a single 5-character window, features of each character are expressed with the position of character from current character. For example, a character two positions to the left of the current position i is expressed as "-2-th-character=c_{i-2}". Each character is classified into one of the 33 NE labels provided by the SE representation.

3 Collecting NE-Related Labels of Words Repeatedly

3.1 NE-Related Labels of Words

We collect the following information as NE-related labels from unlabeled data. The top of Table 3 shows examples of NE extraction results. We count the following two types of frequencies for words from the extraction results. The first is the number of times each NE label is associated with each word. The other is the number of times each NE label is associated with co-occurring words of each word. We count NE labels associated with co-occurring words within two preceding and two succeeding words for co-occurring NE labels of words.

The bottom of Table 3 shows an example of the frequency of occurrence of the NE-related labels associated with the words. For example, for the word "田中 *(Tanaka)*", the frequency of the "S-PERSON" label is 20000 and the "B-ORG" label is 10000. Furthermore, the candidate NE class labels of the words following "田中*(Tanaka)*", "E-ORG" is collected 1000 times and "O" is collected 2000 times. Finally, for the words preceding "*Tanaka*", "O" is collected 1000 times as the candidate NE class label. We can generate the following NE-related labels of each word from the above information.

- **Candidate NE labels**: We use NE labels associated with each word more than 10 times as candidate NE labels of words. For example, B-ORG and S-PERSON are collected as the candidate NE class labels of "田中 *(Tanaka)*" from the examples listed in Table 3.

Table 3. Examples of extraction results (top) and examples of NE-related labels collected from the extraction results (bottom)

田中 /B-ORG	株式会社 /E-ORG	上場 /O
(Tanaka)	(Co.Ltd.)	(go public)
田中 /S-PERSON	社長 /O	
(Tanaka)	(president)	
田中 /S-PERSON	さん /O	
(Tanaka)	(Mr.)	
	:	

↓

Word	Position from the current word	Candidate NE labels in each position	Freq. of NE labels	Ranking in each position
田中 (Tanaka)	current	S-PERSON	20000	1
		B-ORG	10000	2
	next	O	2000	1
		E-ORG	1000	2
	previous	O	10000	1
さん (Mr.)	current	O	20000	1
	previous	S-PERSON	20000	1
		:		

- **Candidate co-occurring NE labels**: We use NE labels associated with co-occurring words of each word more than 10 times as candidate co-occurring NE labels. As the preceding candidate NE labels of "田中*(Tanaka)*", the followings are collected from the examples in Table 3; "E-ORG" and "O" for the next word position, and "O" for the previous word.
- **Frequency information of candidate NE labels and candidate co-occurring NE labels**: These are the frequencies of the NE candidate labels of each word, which are counted from the parsed results. To express the frequencies of NE-related labels as binary features, we categorize the frequencies of these NE-related labels by the frequency of each word n; $10 < n \leq 100$, $100 < n \leq 500$, $500 < n \leq 1000$, $1000 < n \leq 5000$, $5000 < n \leq 10000$, $10000 < n \leq 50000$, $50000 < n \leq 100000$, and $100000 < n$.
- **Ranking of candidate NE-labels**: This information is the ranking of candidate NE class labels for each word. Each ranking is decided according to the label frequencies counted from the parsed results. As for the ranking in the candidate NE class labels of "田中*(Tanaka)*", "S-PERSON" is ranked as the first, and "B-ORG" is ranked as the second. As for the ranking of the candidate NE class labels of "田中*(Tanaka)*" for the next word , "O" is ranked as the first, and "E-ORG" is ranked as the second.

For example, when we identify the NE label of "山田 " in "山田*(Yamada)* さん*(Mr.)*", we use NE-related labels of words like "candidate-NE-label-of-the-next-

word=O", "candidate-NE-label-for-the-previous-word-of-the-next-word=S-PERSON" as the NE-related labels given by the next word "さん" in addition to features described in section 2.1, such as "current-word=山田 " "next-word=さん ".

3.2 Collecting NE-Related Labels of Words

We use NE-related labels of words for additional features as in [7]. However, our method for collecting NE-related labels is different from the method used in [7]. This method collects NE-related labels of words one time. In contrast, we collect NE-related labels of words again and again. The collection method of NE-related labels is as follows.

- (1) Create an NE extractor from a training data, and use the NE extractor as a current NE extractor.
- (2) Collect NE-related labels of words by parsing unlabeled data with the current NE extractor.
- (3) Create a new NE extractor with the training data and the collected NE-related labels of words. The new NE extractor using the NE-related labels of words as features, and this extractor is used for collecting NE-related labels of words at next iteration.
- (4) Go back to step (2), if the termination criterion is not satisfied. The process (2) to (4) is repeated 4 times in this experiment.

The reason why we collect NE-related labels of words in this way is as follows. The paper [7] reported that employing NE-related labels of words contributed improved accuracy. Furthermore, the paper [7] reported that better accuracy is obtained when using an NE extractor employing NE-related labels of words collected with several NE extractors than a single extractor alone. We hypothesize that the result indicates the performance of an NE extractor used for collecting NE-related labels of words relates to improved accuracy. Thus we expect to obtain NE extractors showing better extraction accuracy by repeating the process (2) to (4).

4 Experiments

4.1 Machine Learning Algorithm

We use a boosting-based learner that learns rules consisting of a features, and rules represented by combined features consisting of more than one feature [8]. The boosting algorithm achieves fast training speed by training a weak-learner that learns several rules from a small portion of candidate rules generated from a subset of features called a bucket. The parameters for the boosting algorithm are as follows. We used the number of rules to be learned as R=100,000, the bucketing size for splitting features into several subsets as $|B|$=1,000, the number of rules learned at each boosting iteration as ν =10, the number of candidate rules used to generate new combined features at each rule size as ω=10, and the maximum number of features in rules as ζ=2.

Table 4. Training and evaluation data for Japanese NE extraction in this experiment

	Training	Evaluation		Development	
NE / Data	CRL data	formal-run GENERAL	formal-run ARREST	domain-specific training	dry-run
ARTIFACT	871	49	13	11	42
DATE	3654	277	72	69	110
LOCATION	5660	416	106	165	192
MONEY	390	15	8	19	33
ORGANIZATION	3813	389	74	80	214
PERCENT	500	21	0	3	6
PERSON	3870	355	97	94	169
TIME	503	59	19	18	24
Total	19261	1581	389	459	790

The boosting algorithm operates on binary classification problems. To extend the boosting to multi-class, we used the one-vs-the-rest method. To identify proper tag sequences, we use the Viterbi search. To apply the Viterbi search, we convert the confidence value of each classifier into the range of 0 to 1 with sigmoid function defined as $s(X) = 1/(1 + exp(-\beta X))$, where X is the output of a classifier to an input. We used β=5 in this experiment. Then we select a tag sequence which maximizes the sum of those log values.

In addition, we apply a technique to control the generation of combined features to obtain a fast processing speed which is required to collect NE-related labels of words in real time [16]. Using this technique, instead of specifying combined features to be used manually, we specify features that are not used in the combined features as atomic features. The combined features are only generated from non-atomic features. The boosting algorithm learns rules consisting of more than one feature from the combined features, and rules consisting of only a feature from the atomic and the non-atomic features. We can obtain faster processing speed by reducing the number of combined features to be examined. We specify NE-related labels of words as the atomic features.

4.2 Experimental Settings

The following data prepared for IREX [9] are used in our experiment. We used the CRL data for the training, and the formal-run GENERAL task and the formal-run ARREST task for the evaluation. We use the domain-specific training data in addition to the dryrun data as development data because the domain-specific training data is prepared for the formal-run ARREST task.

Development data is used for determining how many times we repeat the generation process of NE-related labels of words. We select the NE extractor which shows the highest macro average on these development data. Table 4 lists the statistics of NE types for each data set. We compared performance of NE extractors by using the F-measure defined as follows;

F-measure = 2 $\times$ Recall $\times$ Precision / (Recall + Precision),
where,
Recall = NUM / (the number of correct NEs),

Table 5. Experimental results on the development data and the test data: Each F-measure is obtained by using the word-based chunking and the character-based chunking. The "Av." indicates macro average F-measure values of the development data or the evaluation data. Iteration 1 indicates NE extraction results without NE-related labels of words.

Development

Iteration	1	2	3	4	5
domain-specific	90.37	90.91	91.65	91.55	92.11
dryrun	83.22	84.92	84.34	84.88	84.38
AV.	86.79	87.91	87.99	88.21	**88.24**

Evaluation

ARREST	88.39	90.82	89.47	89.81	**91.95**
GENERAL	85.15	87.08	86.90	87.21	**87.34**
AV.	86.77	88.95	88.18	88.51	**89.64**

Precision = NUM / (the number of NEs extracted by an NE extractor),
and NUM is the number of NEs correctly identified by an NE extractor.

The data from the Mainichi Shinbun between 1991 and 2008 are used as unlabeled data for collecting NE-related labels of words.

4.3 Experimental Results

We list experimental results on the development data in the top half of Table 5. We see that we obtain better accuracy by repeatedly collecting NE-related labels of words. The NE extractor at Iteration 5 shows the best average F-measure (FM). After the 5 times iteration, the NE extractor shows 1.45 point higher average FM than that of NE extractor at Iteration 1 which does not use NE-related labels of words. The NE extractor at Iteration 5 shows 0.33 point higher average FM than that of NE extractor at Iteration 2. These results show our method contributes improved accuracy.

We list the extraction results on GENERAL and ARREST in the bottom half of Table 5. We list all the results obtained with NE extractors at Iteration 1 to Iteration 5 for comparison. The FM for GENERAL and ARREST obtained with the NE extractor at Iteration 5 are 87.34 and 91.95. The FM for GENERAL and ARREST obtained with the NE extractor at Iteration 1 without NE-related labels of words are 85.15 and 88.39. Our method improved accuracy by 2.19 and 3.56 on GENERAL and ARREST respectively. The FM for GENERAL and ARREST obtained with the NE extractor at Iteration 2 are 87.08 and 90.91. These results also show the iterative collection of NE-related labels contributes improved accuracy.

5 Related Work

5.1 Comparison with Previous Work in Japanese NE Extraction

Table 6 lists the results of the previous works using IREX Japanese NE extraction tasks. Our results are F-measure values obtained with the NE extractor at Iteration 5. The NE

Table 6. Comparison with previous works; Uchimoto et al. [12], Takemot et al. [17], Utsuro et al. [18], Yamada et al.[19], Isozaki and Kazawa [20], Asahara and Matsumoto [14], Nakano and Hirai [15], Iwakura and Okamoto [7], Sasano and Kurohashi [13], and Kazama and Torisawa [4]. GE and AR indicate GENERAL and ARREST.

	GE	AR	CRL	Method (Lexical resources)
[12]	80.17	85.75	-	ME-based extraction by word-unit and transformation rules (hand-crafted NE dictionaries)
[17]	83.86	-	-	Hand-crafted Rules and Compound Word Lexicon
[18]	84.07	-	-	Stacking of ME-based NE extractors
[19]	-	-	83.2	SVMs-based extraction by word-unit
[20]	85.77	-	86.77	SVMs-based extraction by word-unit and template rules for word unit problems (NTT Goi Taikei)
[14]	-	-	87.21	SVMs-based extraction by character-unit using n-best results of morphological analysis (NTT Goi Taikei)
[15]	-	-	89.03	SVMs-based extraction by character-unit by using Japanese base phrase information (NTT Goi Taikei)
[7]	87.09	90.20	88.50	SVMs-based extraction by character-unit using outputs of a word-based NE extractor (news articles)
[13]	87.72	-	89.40	SVMs-based extraction by word using structural information (Case frame, NTT Goi Taikei)
[4]	-	-	88.93	CRFs-based extraction by character-unit (Web documents and Wikipedia)
This paper	**87.34**	**91.95**	-	A boosting-based extraction by character-unit using outputs of a word-based NE extractor (news articles)

extractor shows the highest average F-measure on the development data. Our results show higher performance than the hand-crafted-rule based NE extractor [17], the NE extraction based on Maximum Entropy (ME) [12], the NE extraction through combination of the outputs of three NE extractors by stacking [18], the SVMs-based extractor [20], and the SVMs-based extractor using NE-related labels of words [7]. These results showed NE-related labels of words repeatedly collected from unlabeled data are useful. However, our method showed worse performance than the SVMs-based NE extractor [13]. The reason seems to be the difference of features. The SVMs-based NE extractor [13] uses several features that are not used in our method. We hypothesize that we can improve performance of our NE extractors by using features used in the other methods in addition to NE-related labels of words.

5.2 Comparison with the Other Semi-supervised Approaches

To collect NE-related labels of words, a one pass process approach is used in the previous work [7]. Compared with this method, we expect to obtain better NE-related labels of words than those obtained by the one pass process approach.

We expect to obtain more target-task-oriented word information with our method than that of previous works, which primarily used information obtained with clustering [1, 2, 4, 6]. This is because our method acquires NE-related labels of words from the

outputs of an NE extractor repeatedly. The NE-related labels of words are different from the word cluster information used in the previous works. Thus we expect to obtain better performance by using NE-related labels of words in addition to word clusters.

Compared with semi-supervised learning algorithms such as [5], our method is one of the feature augmentation techniques similar to techniques using word clusters [1, 2, 4, 6]. Therefore, our method can be applied to NE extractions based on any supervised-learning algorithms or semi-supervised-learning algorithms.

6 Conclusion

This paper proposed an NE extraction method using word information collected from unlabeled data. Our method collects candidate NE class labels of words, NE class labels of co-occurring words, and so on, from unlabeled data. We train a new extractor using the word information as new features, and we use the new extractor for obtaining word information. The experimental results show that our proposed method contributes improved accuracy.

References

1. Freitag, D.: Trained named entity recognition using distributional clusters. In: Proc. of EMNLP 2004, pp. 262–269 (2004)
2. Miller, S., Guinness, J., Zamanian, A.: Name tagging with word clusters and discriminative training. In: HLT-NAACL, pp. 337–342 (2004)
3. Ando, R., Zhang, T.: A high-performance semi-supervised learning method for text chunking. In: Proc. of ACL 2004, pp. 1–9 (2005)
4. Kazama, J., Torisawa, K.: Inducing gazetteers for named entity recognition by large-scale clustering of dependency relations. In: Proc. of ACL 2008: HLT, pp. 407–415 (2008)
5. Suzuki, J., Isozaki, H.: Semi-supervised sequential labeling and segmentation using giga-word scale unlabeled data. In: Proc. of ACL 2008: HLT, pp. 665–673 (2008)
6. Lin, D., Wu, X.: Phrase clustering for discriminative learning. In: Proceedings of ACL-IJCNLP 2009, pp. 1030–1038 (2009)
7. Iwakura, T., Okamoto, S.: Japanese named entity extraction by augmenting features with unlabeled data. IPSJ Journal 49(10), 3657–3669 (2008) (in Japanese)
8. Iwakura, T., Okamoto, S.: A fast boosting-based learner for feature-rich tagging and chunking. In: Proc. of CoNLL 2008, pp. 17–24 (2008)
9. IREX, C.: Proc. of the IREX workshop (1999)
10. Ramshaw, L., Marcus, M.: Text chunking using transformation-based learning. In: Proc. of the Third Workshop on Very Large Corpora, Association for Computational Linguistics, pp. 82–94 (1995)
11. Tjong Kim Sang, E., Veenstra, J.: Representing text chunks. In: Proc. of EACL 1999, pp. 173–179 (1999)
12. Uchimoto, K., Ma, Q., Murata, M., Ozaku, H., Utiyama, M., Isahara, H.: Named entity extraction based on a maximum entropy model and transformation rules. In: Proc. of the ACL 2000, pp. 326–335 (2000)
13. Sasano, R., Kurohashi, S.: Japanese named entity recognition using structural natural language processing. In: Proc. of IJCNLP 2008, pp. 607–612 (2008)

14. Asahara, M., Matsumoto, Y.: Japanese named entity extraction with redundant morphological analysis. In: Proc. of HLT-NAACL 2003, pp. 8–15 (2003)
15. Nakano, K., Hirai, Y.: Japanese named entity extraction with bunsetsu features. IPSJ Journal 45(3), 934–941 (2004) (in Japanese)
16. Iwakura, T.: Fast boosting-based part-of-speech tagging and text chunking with efficient rule representation for sequential labeling. In: Proc. of RANLP 2009 (2009)
17. Takemoto, Y., Fukushima, T., Yamada, H.: A Japanese named entity extraction system based on building a large-scale and high quality dictionary and pattern-matching rules 42(6), 1580–1591 (2001) (in Japanese)
18. Utsuro, T., Sassano, M., Uchimoto, K.: Combining outputs of multiple Japanese named entity chunkers by stacking. In: Proc. of EMNLP 2002, pp. 281–288 (2002)
19. Yamada, H., Kudoh, T., Matsumoto, Y.: Japanese named entity extraction using Support Vector Machine. IPSJ Journal 43(1), 44–53 (2002) (in Japanese)
20. Isozaki, H., Kazawa, H.: Speeding up named entity recognition based on Support Vector Machines. IPSJ SIG notes NL-149-1, 1–8 (2002) (in Japanese)

A Distributional Semantics Approach to Simultaneous Recognition of Multiple Classes of Named Entities

Siddhartha Jonnalagadda[1], Robert Leaman[1], Trevor Cohen[2], and Graciela Gonzalez[1]

[1] Arizona State University, USA
[2] The University of Texas Health Science Center at Houston, USA
Siddhartha.Jonnalagadda@asu.edu, Bob.Leaman@asu.edu,
Trevor.Cohen@uth.tmc.edu, Graciela.Gonzalez@asu.edu

Abstract. Named Entity Recognition and Classification is being studied for last two decades. Since semantic features take huge amount of training time and are slow in inference, the existing tools apply features and rules mainly at the word level or use lexicons. Recent advances in distributional semantics allow us to efficiently create paradigmatic models that encode word order. We used Sahlgren *et al*'s permutation-based variant of the Random Indexing model to create a scalable and efficient system to simultaneously recognize multiple entity classes mentioned in natural language, which is validated on the GENIA corpus which has annotations for 46 biomedical entity classes and supports nested entities. Using distributional semantics features only, it achieves an overall micro-averaged F-measure of 67.3% based on fragment matching with performance ranging from 7.4% for "DNA substructure" to 80.7% for "Bioentity".

Keywords: Distributional, Semantics, Multiple, Named, Entity, Recognition, Classification, GENIA, Biomedical.

1 Introduction

The problem of Named Entity Recognition and Classification (NERC) has been studied for almost two decades [24] and there has been significant progress in the field. While earlier attempts were almost all dictionary or rule-based systems, most of the modern systems use supervised machine-learning, whereby a system is trained to recognize named entity mentions in text based on specific (and numerous) features associated with the mentions that the system learns from annotated corpora. Thus, machine-learning based methods are very dependent not only on the specific technique or implementation details, but also the features used for it. Most of the contemporary high-performing tools use non-semantic features like parts of speech, lemmata, regular expressions, prefixes, n-grams, etc. The high computational cost associated with using deep syntactic and semantic features largely restricted the NERC systems to orthographic, morphological and shallow syntactic features.

Another common limitation of NERC systems based on machine learning techniques such as conditional random fields is the significant computational needs when training on a large, rich corpus like GENIA. Conditional random fields have time complexity O(t*S^2*k*n) for training and O(S^2*n) for decoding[17, 23], where:

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 224–235, 2010.

t is the number of training instances

S is the number of states, which is linear in the number of entity classes and exponential in the order

k is the number of training iterations performed

n is the training instance length

While such probabilistic graphical models have also been used for multi-class NERC[7,22,29,31], these are typically trained for less than six entities and are not particularly computationally efficient. In contrast, however, our system has time complexity O(t*S^0) for training and O(S^0) for decoding.

Distributional Semantics is an emerging field that concerns the automatic estimation of the quantitative relatedness between words and between passages based on the distribution of words in a corpus. These estimates of relatedness have been shown to correspond well with human judgment in a number of evaluations, and have proved useful in many applications also [2]. Random Indexing[14], a recently emerged scalable method of distributional semantics, enables the processing of larger corpora than were possible with previous methods. In this paper, we present and evaluate an initial application that explores the use of distributional semantics for simultaneously recognizing and classifying all the named entities present in the GENIA corpus, which could represent a more elegant solution to the problem of multi-class NERC.

2 Background

Semantic features of varying degrees of sophistication have been used previously in systems like ABNER [29] and the joint parser and NER tool developed in Stanford by Finkel [7]. However, their use has not resulted in any improvement in precision and recall. ABNER, a pioneering system for Biomedical NERC using conditional random fields, uses list-look up techniques based on 17 dictionaries that map individual tokens to their semantic types. The dictionaries include some entered by hand (Greek letters, amino acids, chemical elements, known viruses, plus abbreviations of all these), and those corresponding to genes, chromosome locations, proteins, and cell lines. These dictionaries were built carefully using sound algorithmic techniques. However adding these semantic features to the existing word-level features actually had a deleterious effect of decreasing the f-measure by 0.3%. Finkel's tool uses the distributional similarity model built by Clark [3] in 2000 to determine the cluster to which a particular token belongs to. The clusters were built apriori from the British National corpus and English Gigaword corpus. The major limitations of this approach are that Clark's model uses only the adjacent tokens to calculate the distributional similarity and that the ambiguity in the semantic type of the token depending upon the larger context is not taken into consideration. It is also reported that because they were able to find only 200 clusters, it resulted in slower inference and there was no improvement in performance. On the other hand, most of the state-of-art NERC systems such as BANNER don't use any semantic features including distributional semantic features for want of evidence for scalability and impact on performance [19]. The main contribution of this paper is to create a framework to readily adapt distributional semantic features for NERC, and evaluate the performance of this approach on a corpus with multiple classes and nested entities.

2.1 Distributional Semantics

Methods of distributional semantics can be classified broadly as either probabilistic or geometric. Probabilistic models view documents as mixtures of topics, allowing terms to be represented according to the probability of their being encountered during the discussion of a particular topic. Geometric models, of which Random Indexing is an exemplar, represent terms as vectors in multi-dimensional space, the dimensions of which are derived from the distribution of terms across defined contexts, which may include entire documents, regions within documents or grammatical relations. For example, Latent Semantic Analysis (LSA) [18] uses the entire document as the context, by generating a term-document matrix in which each cell corresponds to the number of times a term occurs in a document. On the other hand, the Hyperspace Analog to Language (HAL) model [20] uses the words surrounding the target term as the context, by generating a term-term matrix to note the number of times a given term occurs in the neighborhood of every other term. In contrast, Schütze's Wordspace [28] defines a sliding window of around 1000 frequently-occurring four-grams as a context, resulting in a term-by-four-gram matrix. Usually, the magnitude of the term vectors depends on the frequency of occurrence of the terms in the corpus and the direction depends on the terms relationship with the chosen base vectors.

Random Indexing: Most of the distributional semantics models have high computational and storage cost associated with building the model or modifying it because of the large number of dimensions when a large corpus is modeled. While dimensionality reduction techniques such as Singular Value Decomposition (SVD) are able to generate a reduced-dimensional approximation of a term-by-context matrix, this compression comes at considerable computational cost. For example, the time complexity of SVD with standard algorithms is essentially cubic [4]. Recently, Random Indexing [14] emerged as promising alternative to the use of SVD for the dimension reduction step in the generation of term-by-context vectors. Random Indexing and other similar methods are motivated by the Johnson–Lindenstrauss Lemma [12] which states that the distance between points in a vector space will be approximately preserved if they are projected into a reduced-dimensional subspace of sufficient dimensionality. While this procedure requires a fraction of the RAM and processing power of Singular Value Decomposition, it is able to produce term–term associations [14] of similar accuracy to those produced by SVD-based Latent Semantic Analysis.

Random Indexing avoids the need to construct and subsequently reduce the dimensions of a term-by-context matrix by generating a reduced-dimensional matrix directly. This is accomplished by assigning to each context a sparse high-dimensional (on the order of 1000) *elemental vector* of the dimensionality of the reduced dimensional space to be generated. These vectors consist mostly of zeros, but a small number (on the order of 10) +1 and -1 values are randomly distributed across the vector. Given the many possible permutations of a small number +1's and -1's in a high-dimensional space, it is likely that most of the assigned index vectors will be close-to-orthogonal (almost perpendicular) to one another. Consequently, rather than constructing a full term-by-context matrix in which each context is represented as an independent dimension, a reduced-dimensional matrix in which each context is represented as a close-to-independent vector is constructed. Term vectors are then

generated as the linear sum of the sparse elemental context vectors of each context in which they occur, weighted according to frequency. The method scales at a rate that is linear to the size of the corpus, and is consequently much faster than previous methods (processing, for example the entire MEDLINE corpus in around 30 min), allowing for the rapid prototyping of semantic spaces for experimental purposes. In addition, Random Indexing implementations, such as the Semantic Vectors package used in this research [33], tend to support both term-by-document and sliding-window based indexes, allowing for the comparison between these types of indexing procedures in particular tasks. Random Indexing also efficiently integrates new documents into an existing semantic space, allowing for implementation of efficient NERC systems.

Paradigmatic vs. Syntagmatic relations: Recent research in distributional semantics has explored the differences between relations extracted depending on the type of context used to build a model [26]. As defined by de Saussure [27], there are two types of relationship between words – syntagmatic and paradigmatic. If two words co-occur significantly in passages or sentences, they are said to be in syntagmatic relationship. Examples include terms that occur frequently in succession such as p53 and tumor, APOE and AD, and poliomyelitis and leg. If two words can substitute for each other in the sentences while maintaining the integrity of the syntactic structure of a sentence, they are said to be in a paradigmatic relationship. Examples include: p53 and gata1, AD and SDAT, and poliomyelitis and polio. Since words in paradigmatic relationship don't occur together in the same context, extracting such a relationship typically requires 2nd order analysis, while a 1st order analysis is sufficient to extract syntagmatic relationships. Sahlgren argues that using a small sliding-window rather than an entire document as a context is better suited to extracting paradigmatic relations, and supports this argument with empirical results [25]. The NERC task involves finding words that could conceivably replace the token we want to label without disturbing syntactic structure. However, in scientific language, domain semantics also determine which terms could replace one another [11]. We have chosen to model paradigmatic relationship using the vector coordinate permutations model introduced by Sahlgren, Holst and Kanerva [25], as it has been observed that the relations captured by this method tend to emphasize terms of a similar semantic class [34, 2].

Encoding Word Order Using Permutation: In addition to providing a paradigmatic model, Sahlgren's permutation-based method encodes word-order, thus accounting for the sequential structure of language. The order of the word signifies the grammatical role and hence the meaning of the word. This method is an alternative implementation to the convolution and superposition operations used by BEAGLE [13] to encode word-order information in word spaces. Sahlgren's method captures word information by permutation of vector coordinates which is a computationally light alternative to BEAGLE's convolution operation. To achieve this, Sahlgren *et al* use Random Indexing of vectors [14] to generate context vectors for each term and use permutation or shuffling of coordinates (shifting of all of the non-zero values of a sparse elemental vector to the left or right according to the relative position of terms) to replace the convolution operator. In this way, a different close-to-orthogonal elemental vector is generated for each term depending on its position within the sliding window. A

semantic term vector for each term is then generated as the linear sum of the permuted elemental vectors for each term co-occurring with this term in a sliding window. This permutation function is reversible, allowing for construction of order-based queries. Permutation-based indexing is supported in the Semantic Vectors package (see below), and is described in further detail in Sahlgren, Holst and Kanerva 2008 [25].

2.2 Semantic Vectors System

Semantic Vectors (http://semanticvectors.googlecode.com) is a scalable open source package written in Java and depends only on Apache Lucene. Semantic Vectors software package can be used to create distributional semantic vectors from corpora and also perform different mathematical operations on them. The package also supports operations for finding scalar products and cosine similarity, normalization, tensor operations (inner and outer product, sum, normalization), convolution products, and orthogonalization routines for vector negation and disjunction between term vectors and document vectors.

Apache Lucene: Apache Lucene (http://lucene.apache.org/) is a powerful and widely used piece of open source software that is used by us for tokenization and indexing to extract the relative positions of terms from the corpus. The positions of terms within each document are input to Semantic Vectors package to create a reduced-dimensional approximation of a position-dependent term-by-term matrix. Lucene builds an index for all the documents that need to be searched and a count of the tokens in the document are stored in the term-document matrix. We use the tokenization methods provided by Lucene's StandardAnalyzer class to standardize the tokens from sentences in the test set with those from sentences in the training set. The rules for tokenization from Lucene as available in the documentation of http://lucene.apache.org/java/2_3_0/api/org/apache/lucene/analysis/standard/StandardTokenizer.html.

2.3 GENIA Corpus

To the best of our knowledge GENIA corpus [15,16] is the most complex corpus used to evaluate NERC systems, with around 100,000 annotations for 47 biologically relevant categories from 2000 PUBMED abstracts consisting of more than 400,000 words. Roughly 17% of the entities are embedded within another entity. Because of the limitations discussed in the introduction, there is no framework which recognizes and classifies all the entities above at the same time.

3 Methods

The architecture is a 2-stage pipeline as shown in Figure 1. The entire corpus is broken into more than 18000 documents, each of which contains a unique sentence of the GENIA corpus. A Lucene index is built for this set of documents. The term and document vectors are built using the Semantic Vectors package. We used the Sahlgren's Permutation-based model [25] with the dimensionality of the reduced-dimensional

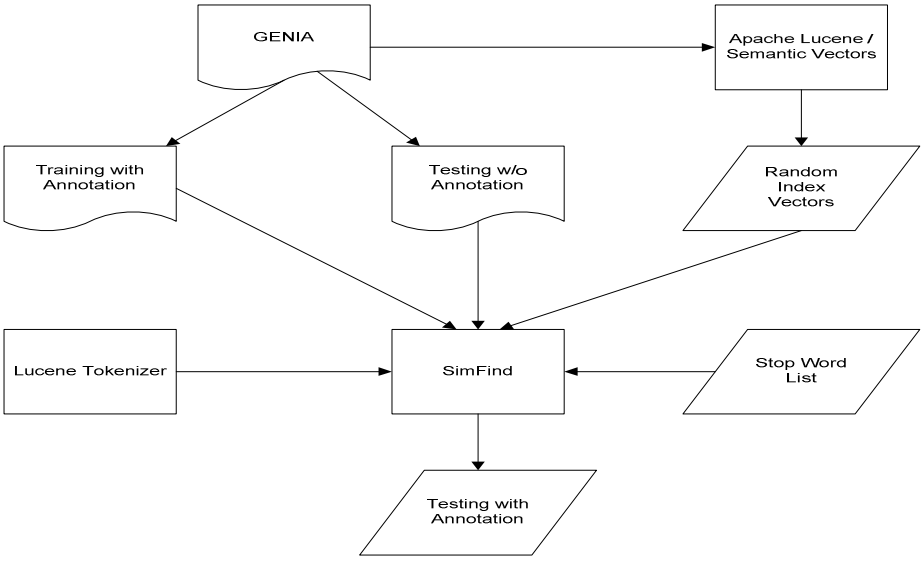


Fig. 1. System Architecture

space as 200 to produce the Random Index Vectors. We selected a sliding window size taking into account the 5-tokens before and after the target token.

The corpus is divided into two halves – one half is the training set and the other half is the test set. The Lucene Tokenizer breaks the sentence into tokens and the SimFind algorithm is used to find the token in the training set that is most similar to the target token. The entity class of the similar token is then assigned to the target token. SimFind therefore takes into consideration the surrounding context when determining the semantic type of each token while previous methods considered the semantic type of the token independent of the context.

In this research, we utilize the estimates of similarity provided by Random Indexing for two purposes. Firstly, as token labels are context-dependent, we find the 100 most similar sentences from the training set that are similar to the vector sum of the terms belonging to the target sentence. Next, we find the first token from the similar sentences that is same as the target token or similar to it. Thus, the SimFind algorithm takes into account all the other tokens present in the sentence and it also doesn't assume that the target token is present in the training set. The pseudo code for the algorithm is explained in Table 1. The complete source code with documentation is publicly available at: http://www.public.asu.edu/~sjonnal3/SV_ NER_src.zip.

We use the list of 421 stop words created by Fox from the Brown corpus [9] to improve the efficiency of SimFind. These stop words were selected to be maximally efficient and effective in filtering the semantically neutral words. There are several options for the labeling model. The simplest is the IO model, which indicates whether the token is **i**nside an entity or **o**utside an entity, which is the model we employ for this work. Another possible model is IOB, where each token is labeled to be either

Table 1. SimFind algorithm

SimFind(targetToken, Line){
List simSentences = getSimilarSentences(Line,100);
List goldenTokenLabel = getTokenLabels(simSentences);
STEP1:
FOREACH (goldenTokenLabel)
IF (goldenTokenLabel has targetToken as token)
return goldenTokenLabel;
STEP2:
IF (token IN STOPLIST)
return <token,NONE>;
terms = 1;
STEP3:
terms *= 10;
<equivTokens,simIndex>=getSimWords(targetToken,terms);
FOREACH (equivToken)
FOREACH (goldenTokenLabel)
IF (goldenTokenLabel has targetToken as token)
return goldenTokenLabel;
IF (simIndex>0.5)
goto STEP3;
return <token, NONE>;
}
	The SimFind function is the core method which retrieves the sentences which share the same context as the target sentence and for each token in the target sentence. The algorithm first checks for the earliest appearance of the target token in the set of similar sentences arranged in the order of similarity. The next step should be to search for the presence of the tokens similar to the target tokens. However, to minimize the total time taken, we eliminate the tokens which appear too frequently in common English and hence are highly unlikely to be part of a biomedical entity.

getSimilarSentences(line, numberOfResults){
break line into tokens using Lucene Tokenizer;
form query vector by computing the sum of tokens;
search for similar documents in Random Index Vectors;
set the number of results to be numberOfResults;
listOfSimilarSentences are the sentences from the training set which correspond to these documents;
return listOfSimilarSentences;
}
	The getSimilarSentences function is responsible for finding the specified number of sentences from the training set that are similar to the vector sum of the terms belonging to a given sentence in the test set.

getSimWords(targetToken, count){
form query vector as targetToken;
search for similar terms in Random Index Vectors;
set the number of results to be count;
return list of similar terms;
}
	The getSimWords function is responsible for fetching the tokens in the corpus similar to a given token.

getTokenLabels(simSentences){
for each token in simSentences
find the label from the xml annotation and add <token, label> to listOfTokenLabels;
return listOfTokenLabels;
}
	The getTokenLabels function is used to get the semantic type of the tokens in an annotated sentence.

beginning of an entity, **i**nside an entity, or **o**utside an entity. There are also systems using IOBEW model which in addition label for the **e**nd of the entity and one-**w**ord entity. In the recent evaluation of BANNER [19], an NERC tool, which used a corpus annotated with biomedical entities for recognizing gene entities, the difference between the performances of these three labeling models was found to be less than 1%. Each token can belong to multiple semantic types as GENIA annotates nested entities. Since there are 36 entity classes at leaf level[15], there are 2^{36} possible types of labels with the IO model.

4 Results

NERC systems are typically evaluated using exact matching, which requires that both the left and right boundary match exactly. For many applications, however, determining the exact boundary is not necessary and it is sufficient to determine whether the sentence contains an entity of the specified type or not and its approximate location. Thus, recently more realistic matching techniques like core-term matching and fragment matching are becoming prominent [32]. In core-term matching, the system's annotated named entity must contain a core term of the named entity in the gold standard. This requires that every annotation in the corpus should also mention which is the core-term. In a corpus like GENIA with around 100,000 entities, this would require an excessive amount of annotation resources. In fragment matching, each token is treated separately. This provides a measure of how much fraction of the entity is matched and is thus more realistic than conventional exact matching and loose partial matching.

Since it is shown [5] that 5x2 validation is statistically more powerful than 10x1 validation, we chose to evaluate using 5x2 validation. We present in Table-2 the precision, recall and f-score measures achieved by our system on all the entities annotated in the GENIA corpus except the biologically irrelevant entities like Protein N/A, DNA N/A, and those with insufficient data. We also provide the count of true positives, false positives, and false negatives in each case. For most of the entities, we are one of the first to use GENIA for evaluation. Hence our results also serve as comparison for all NERC systems that would be evaluated using GENIA corpus. In addition, for each entity we calculate the F-score for a system that randomly assigns a positive or negative in the ratio of the number of actual true or false cases respectively. If a corpus has t tokens belonging to a particular entity class and f tokens not belonging to that entity class, a system which randomly assigns tokens to that class in proportion to the known proportion of positives and negatives would result in both precision and recall approximating $t/(t+f)$. The f-score of the random system would therefore also be approximately $t/(t+f)$, which serves as a quantitative estimate the difficulty of NERC task for a specific entity class. This quantity is labeled Random F-score in Table-2.

The entities in Table 2 are arranged in descending order of their f-scores based on our system. It is encouraging to see that more than half of the entity classes have an f-score greater than 50% just based on distributional semantics features and also the huge differences between f-score and Random F-score. The system also has a considerable good overall micro-averaged f-score of 67.3% which is calculated by adding the respective true positives, false positives and false negatives of each entity class. It took around 5 minutes to build the semantic vectors from the documents belonging to the GENIA corpus and around 3 hours to produce results for the testing set which constitutes more than 9000 sentences. This suggests that this framework is scalable and could have significant impact on the precision and recall of a more complex system.

Table 2. Results for the GENIA entities

Entity	Precision (%)	Recall (%)	F score	TP	FP	FN	Random F-score
Bio-entity	78.9	82.5	80.7	60479	16131	12868	26.22
Substance	77.0	79.6	78.3	46587	13796	11976	20.92
Organic compound	77.0	79.5	78.2	46244	13792	11944	20.82
Compound	77.0	79.5	78.2	46382	13822	11980	20.82
Amino acid	69.4	71.1	70.3	27331	11917	11091	13.69
Protein	69.2	71.0	70.1	26692	11864	10890	13.37
Lipid	66.1	67.0	66.5	1243	637	618	0.66
Virus	65.6	67.3	66.4	1641	862	797	0.86
Source	61.4	66.2	63.7	10434	6554	5326	5.62
Atom	62.0	60.2	61.1	150	92	99	0.10
Nucleotide	57.0	64.4	60.5	114	86	63	0.05
Other organic compound	62.1	58.9	60.5	2105	1285	1470	1.28
Protein molecule	60.9	59.4	60.1	13194	8456	9014	7.87
Organism	59.6	58.8	59.2	2085	1412	1460	1.23
Amino acid monomer	61.2	53.1	56.9	256	162	225	0.15
Mono Cell	69.4	45.9	55.3	100	44	118	0.10
Inorganic	57.1	53.3	55.1	97	73	85	0.66
Natural source	52.0	57.6	54.6	6017	5560	4427	3.76
Carbohydrate	63.2	45.7	53.1	43	25	51	0.05
Nucleic acid	51.0	54.2	52.6	9181	8803	7752	6.05
DNA	48.3	52.6	50.4	7829	8366	7051	5.31
DNA domain or region	44.4	48.5	46.4	5889	7362	6253	4.35
Cell type	42.7	50.7	46.3	3046	4089	2968	2.14
Cell line	44.0	44.9	44.5	2375	3022	2912	1.87
Artificial source	43.9	44.3	44.1	2442	3118	3074	1.98
RNA	47.0	41.2	43.9	707	797	1011	0.61
Body part	39.6	45.0	42.1	148	226	181	0.10
Other name	42.6	40.0	41.3	11591	15645	17367	10.31
Protein domain or region	41.9	38.8	40.2	606	842	958	0.56
Protein complex	40.4	40.1	40.2	1509	2226	2256	1.33
Protein family or group	34.0	39.8	36.7	3761	7289	5697	3.36
Peptide	41.9	32.7	36.7	149	207	307	0.15
RNA molecule	36.5	36.7	36.6	453	783	777	0.40
Multi Cell	36.5	34.7	35.6	315	547	593	0.30
Polynucleotide	44.9	27.0	33.7	62	76	168	0.10
Protein subunit	31.2	31.1	31.2	379	834	838	0.40
DNA molecule	24.6	22.6	23.6	174	533	597	0.30
Tissue	22.8	23.7	23.3	151	510	486	0.20
RNA family or group	28.3	15.7	20.2	67	170	360	0.15
Protein substructure	12.2	16.5	14.0	21	151	106	0.05
DNA family or group	12.8	14.5	13.6	270	1844	1588	0.66
DNA substructure	6.1	9.3	7.4	11	170	107	0.05
Overall Score	**66.3**	**68.4**	**67.3**	**342330**	**174180**	**157909**	

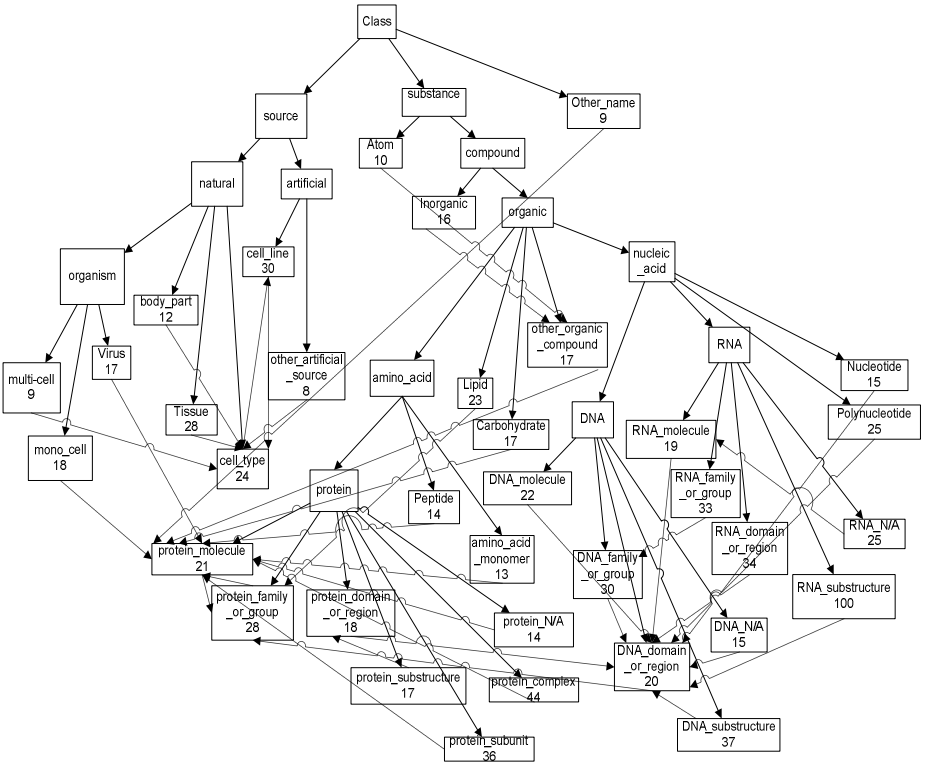


Fig. 2. Depiction of which entities cause confusion for each entity. Each dotted arrow shows which biologically-relevant leaf-level entity class (at head of the arrow) causes most confusion for each leaf-level entity class (at the tail of the arrow) with the corresponding confusion percentage below its name.

There have been several attempts [1,8,10,21,30,35,36] using machine learning to find nested entities in text with many entities like GENIA corpus. As discussed in section 2, these systems limit themselves to work for less than six entities at a time due to computational cost. Since our framework also recognizes nested entities, we believe that it can be used to provide features that can be quickly calculated and can replace the features with slower inference.

We attempted to analyze the errors made by our system by characterizing the confusion between the entitiy classes. An entity class A is said to have confused entity class B, iff either at least one of the false positives of B actually belongs to A or at least one of the false negatives of B was considered by the system to belong to A. The confusion percentage of entity class A relative to entity class B can be defined as the percentage of times A confuses B for a given corpus and a given cross-fold validation. Such knowledge helps us in discovering, refining or validating relationship between entity classes and creating more meaningful ontologies. Information on which entity classes damage the results of the target entity class will be valuable in creating more efficient and powerful rules or features. For example: 34% of the mistakes in classifying “RNA

domain or region" were caused because of "DNA domain or region"; 44% of the mistakes caused in classifying "Protein complex" were caused by "Protein molecule"; and 23% of the mistakes caused in classifying "Lipids" were caused by "Protein molecule". In a significant number of cases, most of the confusions were caused by the immediate siblings as would be expected, but there were many exceptions. For example: "RNA domain or region" with "DNA domain or region"; "Lipids" with "Protein molecule"; and "DNA domain or region" with "Protein family or group". This reflects both the ambiguity inherent in natural language and also the fact that while the GENIA ontology reflects a consideration of the major properties of an entity, the local context of a mention may be more indicative of a single property that may be shared with entities which are otherwise significantly different.

5 Conclusion

We have proposed a scalable, efficient and accurate system using distributional semantic vectors to recognize all the entity classes in natural language using an annotated corpus. Our system is validated on GENIA corpus which has 46 entity classes with annotation that supports nested entities and achieves an overall micro-averaged f-score of 67.3% using fragment matching. In the future, we would present a machine-learning based system that uses distributional semantic features in addition to the available features.

References

1. Byrne, K.: Nested Named Entity Recognition in Historical Archive Text. In: Proceedings of International Conference on Semantic Computing (2007)
2. Cohen, T., Widdows, D.: Empirical distributional semantics: methods and biomedical applications. Journal of Biomedical Informatics 42 (2009)
3. Clark, A.: Inducing Syntactic Categories by Context Distribution Clustering. In: Proceedings of the 2nd Workshop on Learning Language in Logic and the 4th Conference on Computational Natural Language Learning (2000)
4. David, B., Lloyd, T.: Numerical linear algebra. Society for Industrial and Applied Mathematics, Philadelphia (1997)
5. Dietterich, T.G.: Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. Neural Comput. 10 (1998)
6. Eddy, S.R.: Hidden Markov Models. Curr. Opin. Struct. Biol. 6 (1996)
7. Finkel, J.R., Manning, C.D.: Joint Parsing and Named Entity Recognition. In: Proceedings of of NAACL HLT (2009)
8. Finkel, J.R., Manning, C.D.: Nested Named Entity Recognition. In: EMNLP (2009)
9. Fox, C.: A Stop List for General Text. ACM SIGIR Forum 24 (199)
10. Gu, B.: Recognizing Nested Named Entities in GENIA Corpus. In: Proceedings of the Workshop on Linking Natural Language Processing and Biology: Towards Deeper Biological Literature Analysis (2006)
11. Harris, Z.S.: The structure of science information. Journal of Biomedical Informatics 35 (2002)
12. Johnson, W.B., Lindenstrauss, J.: Extensions of Lipschitz Mappings into a Hilbert Space. Contemporary Mathematics 26 (1984)

13. Jones, M.N., Mewhort, D.J.K.: Representing Word Meaning and Order Information in a Composite Holographic Lexicon. Psychol. Rev. 114 (2007)
14. Kanerva, P., Kristofersson, J., Holst, A.: Random Indexing of Text Samples for Latent Semantic Analysis. In: Proceedings of the 22nd Annual Conference of the Cognitive Science Society (2000)
15. Kim, J.D., Ohta, T., Tateisi, Y., et al.: GENIA Corpus-a Semantically Annotated Corpus for Bio-Textmining. Bioinformatics-Oxford 19 (2003)
16. Kim, J.D., Ohta, T., Tsujii, J.: Corpus Annotation for Mining Biomedical Events from Literature. BMC Bioinformatics 9 (2008)
17. Lafferty, J., McCallum, A., Pereira, F.: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In: Proceedings of ICML (2001)
18. Landauer, T.K., Dumais, S.T.: A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition, Induction, and Representation of Knowledge. Psychol. Rev. 104, 211–240 (1997)
19. Leaman, R., Gonzalez, G.: BANNER: An Executable Survey of Advances in Biomedical Named Entity Recognition. In: Proceedings of PSB (2008)
20. Lund, K., Burgess, C.: Hyperspace Analog to Language (HAL): A General Model of Semantic Representation. Language and Cognitive Processes (1996)
21. Màrquez, L., Villarejo, L., Martí, M.A., et al.: Semeval-2007 Task 09: Multilevel Semantic Annotation of Catalan and Spanish. In: Proceedings of the 4th International Workshop on Semantic Evaluations (2007)
22. McCallum, A., Li, W.: Early Results for Named Entity Recognition with Conditional Random Fields, Feature Induction and Web-Enhanced Lexicons. In: Proceedings of CoNLL (2003)
23. McDonald, R., Fernando, P.: Identifying gene and protein mentions in text using conditional random fields. BMC Bioinformatics (2005)
24. Rau, L., Res, G., Center, D., et al.: Extracting Company Names from Text. In: Proceedings of IEEE Conference on Artificial Intelligence Applications (1991)
25. Sahlgren, M., Holst, A., Kanerva, P.: Permutations as a Means to Encode Order in Word Space. In: Proceedings of CogSci. (2008)
26. Sahlgren, M.: The Word-Space Model. Doctoral Dissertation in Computational Linguistics. Stockholm University (2006)
27. Saussure, F., Bally, C., Séchehaye, A., et al.: Cours de linguistique générale. Payot, Paris (1922)
28. Schütze, H.: Automatic Word Sense Discrimination. Computational Linguistics 24, 97–123 (1998)
29. Settles, B.: ABNER: An Open Source Tool for Automatically Tagging Genes, Proteins and Other Entity Names in Text. Bioinformatics 21 (2005)
30. Shen, D., Zhang, J., Zhou, G., et al.: Effective Adaptation of a Hidden Markov Model-Based Named Entity Recognizer for Biomedical Domain. In: Proceedings of ACL (2003)
31. Song, Y., Kim, E., Lee, G.G., et al.: POSBIOTM-NER in the Shared Task of BioNLP/NLPBA 2004. In: Proceedings of IJNLPBA (2004)
32. Tsai, R.T., Wu, S.H., Chou, W.C., et al.: Various Criteria in the Evaluation of Biomedical Named Entity Recognition. BMC Bioinformatics 7 (2006)
33. Widdows, D., Ferraro, K.: Semantic Vectors: A Scalable Open Source Package and Online Technology Management Application. In: Proceedings of LREC (2008)
34. Widdows, D., Cohen, T.: Semantic Vector Combinations and the Synoptic Gospels. In: Proceedings of the Third Quantum Interaction Symposium (2009)
35. Zhou, G., Zhang, J., Su, J., et al.: Recognizing Names in Biomedical Texts: A Machine Learning Approach. Bioinformatics 20 (2004)
36. Zhou, G.D.: Recognizing Names in Biomedical Texts using Mutual Information Independence Model and SVM Plus Sigmoid. Int. J. Med. Inf. 75 (2006)

The Recognition and Interpretation of Motion in Language

James Pustejovsky, Jessica Moszkowicz, and Marc Verhagen

Laboratory for Linguistics and Computation
Department of Computer Science
Brandeis University
Waltham, MA 02454
jamesp@cs.brandeis.edu

Abstract. In this paper, we develop a framework for interpreting linguistic descriptions of places and locations as well as objects in motion as found in natural language texts. We present an overview of existing qualitative spatiotemporal models in order to discuss a more dynamic model of motion called Dynamic Interval Temporal Logic (DITL). The resulting static and dynamic descriptions are represented in a spatiotemporal markup language called STML. The STML output then enables a grounding within a metric representation such as Google Earth, through an automatic conversion to KML. Consistent with the STML output, DITL provides a semantics for STML for subsequent reasoning about the text.

1 Introduction

This paper describes our current research efforts towards developing linguistically and cognitively grounded algorithms for reasoning about spatial relations between regions and objects in motion, as described in natural language text. To illustrate these concerns, consider the following excerpt from a travelblog about biking through Central America[1]:

(1) David left San Cristobal de Las Casas four days ago. He arrived in Ocosingo that day. The next day, David biked to Agua Azul and played in the waterfalls there for 4 hours. He spent the next day at the ruins of Palenque and drove to the border with Guatemala the following day.

In order to understand the spatiotemporal aspects of this text, we must be able to extract several kinds of information, including temporal and spatial expressions as well as predicate-argument structure. Current technologies are now able to flag and disambiguate place names in text in terms of geo-coordinates [1], and also to flag and normalize relative and absolute times in text in terms of a calendar representation [2], [3]. Technologies for extracting named entities from text are now fairly commonplace [4], [5], [6], [7]. In analytic applications, developers may

[1] http://www.rideforclimate.com/journals/?p=68

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 236–256, 2010.

construct a system which weakly associates people names with surface events, places, and times, based on co-occurrence in some context. However, such a coarse-grained representation requires a great deal of further filtering by the analyst, and considerable interpolation and extrapolation by hand by the analyst to figure out who did what at specific times and places.

In this work, we take advantage of recently developed technologies that parse the temporal information from natural language texts. In particular, we build on the IARPA-funded TARSQI project which created an open source natural language *Temporal Processing System* (the TARSQI Toolkit, TTK) [8], [9]. Based on the ISO-TimeML markup language [10,11], this system combines rule-based systems, machine learning, and temporal reasoning to link events in a document to the normalized calendar times when they occur, and to order events in a document with respect to each other.

Just as a temporal reasoning system should be able to infer a temporal ordering over the events, a spatial reasoning system should be able to do at least three things:

(2) a. Identify place and location entities in the text;
b. Perform coreference and non-coreference binding over the locations identified;
c. Create a "spatial narrative" (called a trajectory) for any entity in motion.

For basic spatial information, a markup scheme called SpatialML [12] has recently been developed to map relative and absolute locations (both proper names as well as nominal place descriptions in a document), to geo-coordinates. With SpatialML, we can capture locations such as *San Cristobal de las Casas* and *the waterfalls*, but predicates that involve change of location (and, therefore, a temporal component) are not considered markable. Under a previous SGER NSF grant, we integrated the TimeML and SpatialML annotation schemes with a shallow representation of arguments, i.e., participants involved in events. Another product of the prior NSF research was the creation of an Event Structure Lexicon for motion predicates [13] that encodes the syntax and semantics of motion verbs in English.

Here we describe an extension to previous work in order to create a new markup language that covers both spatial and spatiotemporal information in text. The resulting annotation includes attributes that allow for automatically grounding the annotation on a map. In addition, we adapt the TARSQI Toolkit and extend spatial processing software from the aforementioned grant. Eventually, we will embed these modules in a Spatial Processing Toolkit, which will create representations of object motion as expressed in texts.

Linguistically grounded theories of motion have, until recently, been largely overlooked by the qualitative spatial reasoning community (cf. however, [14], [15], [16]). To help remedy this, we develop a theory of motion based on qualitative spatial dynamics, which addresses the fundamental distinction between path and manner constructions. Path motion predicates introduce reference to a distinguished location such as in the sentence *He arrived in Ocosingo that day.* Pure manner-of-motion predicates do not make use of a distinguished location, as in *David biked*

all day; they can, however, be used in a distinguished location interpretation by embedding the motion verb within a path construction, as seen in *David biked to Agua Azul*. We represent motion with Dynamic Interval Temporal Logic (DITL), which will serve as a partial semantics for STML, a markup language that draws on the annotations provided by SpatialML and TimeML while added the ability to talk about paths and locate events in space. This is especially important since there has been little systematic research done on integrating the interpretation of spatial information in text with other aspects of text understanding. The ability to reason about individual locations and track movements is of broad applicability and obvious interest to previously disparate communities.

The paper proceeds as follows. In the next section, we describe the requirements of encoding spatial and temporal information from text within a markup language, and introduce a new language called ISO-Space defined over this domain. We then illustrate the two major strategies that are employed in natural language to describe the movement of an object through space, and discuss how these strategies can be represented in terms of qualitative models of spatial relations. These qualitative spatiotemporal analyses are then used as the formal and cognitive underpinnings for a dynamic treatment of motion, as described in the section on Dynamic Interval Temporal Logic. This, in turn, forms the semantic basis for a spatiotemporal annotation scheme for automatic markup of static relations and motion in natural language texts, called STML. In the final section, we describe the initial version of the Spatial Processing Toolkit (SPTK) we are developing, which integrates temporal and spatial annotations and identifies the paths traversed by entities in motion.

2 Semantic Annotation of Spatiotemporal Information

2.1 Place Annotation with SpatialML and ISO-Space

There has been considerable research on the linguistic behavior of spatial predicates and prepositions in language [17], [18], [19], [20], [21], [22]. Within qualitative spatial reasoning (QSR), work has recently started to focus on incorporating mereo-topological concepts into the calculus of relations between regions [23], [24], [25].

The focus of SpatialML [12] is to mark up spatial locations mentioned in texts while allowing integration with resources that provide information about a given domain, such as physical feature databases and gazetteers. The core SpatialML tag is the PLACE tag, which has attributes `type` (country, continent, populated place, building, etc.), `country`, `gazref` (a reference to a gazetteer entry) and `latlong`. Complex locations such as *Pacific coast of Australia* and *the hot dog stand behind Macy's* are annotated using the LINK and PATH tags, respectively.[2]

[2] Note that a new version of SpatialML is currently under development that implements several changes suggested for the specification by the SpatialML working group, which includes two of the authors. Most notably, PATH is being replaced with a Relative Location Link that more accurately captures the intended meaning of these relations.

The link types for the LINK tag are adopted from the RCC8 version of the Region Connection Calculus [25], [26], which uses eight basic topological configurations of region pairs as a basis for a spatial calculus. The SpatialML link types include: IN (tangential and non-tangential proper parts), EC (extended connection), DC (discrete connection), PO (partial overlap), EQ (equality), and NR (near).

SpatialML is one of the cornerstones of a new standard being developed within the ISO, named ISO-Space. ISO-Space extends and enriches the spatial expressiveness seen in SpatialML. Specifically, ISO-Space focuses on encoding the following spatial properties: (a) topological relations between objects, (b) orientation and metric relations between objects, (c) shape of an object, (d) size of an object, (e) elevation (LatLong values), (f) geopolitical Entities, (g) granularity, (h) aggregates and distributed objects (spatial integrity), and (i) objects in motion.

2.2 Annotation of Temporal Information with ISO-TimeML

The recognition of spatial entities in natural language is an important component of understanding a text [27]. However, simply identifying fixed geospatial regions and specific "facilities" is not enough to achieve a complete representation of all the spatial phenomena present since it leaves out one of the most crucial aspects of spatial information, motion. To capture motion, we must integrate temporal and spatial information with the lexical semantics of motion predicates and prepositions.

TimeML [10], [11] is an annotation scheme for representing temporal information in text. The basic elements of a TimeML annotation are temporal expressions such as dates, times, and durations, and events that can be anchored or ordered to those expressions or with respect to each other. Once these temporal objects are captured, they are related to each other by way of a temporal link. TimeML's temporal relations are based on Allen's 13 basic relations [28] and include before, simultaneous, includes, begins, ends, as well as their inverses and an identity relation. In addition to temporal links, TimeML includes subordinating links that are used to capture information about irrealis events. This allows temporal links to be created even when the participating events may or may not have happened. For example, in *John planned to leave on Tuesday*, a temporal link can only anchor the *leave* event to *Tuesday* if the annotation also includes the fact that it is not clear whether this event has actually occurred. By adding a subordinating link between the *plan* event and the *leave* event of type Modal, we can safely say that, if the *leave* event happened, then it happened on Tuesday. Subordinating links will also be necessary for spatiotemporal annotation so that subordinated events can safely be located in space, even if we are not sure if the event has really occurred. Consider the sentence *John planned to spend four hours at the park.* Not only do we want to create a temporal link between the *spend* event and four hours, but we also want to spatially anchor this event to the *park* location. However, the text does not tell us whether this event has actually happened. The subordinating link from ISO-TimeML effectively wraps the *spend* event in a modal context so that both of these links can be created.

Associated with the TimeML representation language is a manually annotated corpus named TimeBank [29] and a set of tools for automatically annotating times, events, and relations called the TARSQI Toolkit [9].[3] TimeML has recently been incorporated into the new ISO-TimeML standard.

2.3 Towards a Spatiotemporal Markup Language

Because we are interested in both static and dynamic descriptions of spatial relations, we need a broader definition of what is to be captured by a spatiotemporal specification than is expressed in SpatialML. The following list contains what we believe are the required elements of the Spatiotemporal Markup Language (STML).[4]

1. Places[5]: geographic, geopolitical places, functional locations, arbitrary locations;
2. Entities as Spatial Objects: intrinsic orientation, dimensionality, size, shape;
3. Path Objects: routes, lines, turns, arcs;
4. Links: topological relations, dimension and orientation, metrics;
5. Spatial Functions: *behind the building, twenty miles from Boulder*
6. Movements and Spatial Processes: functions from regions to regions.

Wherever possible, STML leverages existing resources, such as ISO-TimeML, when referencing concepts such as times, durations, and orderings.

Modeling Locations and States with STML. STML uses the REGION tag as a general term for any area of space that is relevant to the annotation. These can be static places such as *Boston* or *the building*, or they can be coerced locations such as *car* as in *John and Mary got into the car.* Any moving object is also a region, including individuals such as *John* or *Mary* in the above sentence. Static locations can be modeled with SpatialML. However, SpatialML's treatment of moving objects is spotty and currently (as of version 3.0) does not extend beyond the VEHICLE type. Special attention will have to be given to these elements in the STML specification.

Another important aspect of the annotation of locations is the use of spatial functions to define new locations implicitly. SpatialML focuses only on explicitly mentioned locations, but, for STML, we also want to be able to locate events in space whenever possible. For example, in the sentence *There was an accident in front of the bank*, it is not enough to simply identify *the bank* as a location, but we also want to say that the *accident* event occurred at an implicit location that is introduced by the spatial function *in front of*. In STML, spatial functions will provide an identification number for these implicit locations.

[3] TimeBank is available for free from the Linguistics Data Consortium, `http://ldc.upenn.edu/`
The tools can be downloaded from `http://www.timeml.org/`

[4] We also draw heavily on [30] for spatially relevant categories.

[5] This tag is named REGION in ISO-Space, but we will continue to use the term PLACE here as this is what is used in SpatialML.

Regions can be related to each other by way of a qualitative spatial link. This tag builds on the SpatialML LINK tag. It includes topological relationships based on RCC8 as well as relative relationships such as above, below, and next to.

Many events do not involve a change of location, but are still directly related to an STML region. This is the case in the sentence *David spent the next day at the ruins of Palenque.* ISO-TimeML marks *spent* as an event and STML marks *the ruins of Palenque* as a region. In addition, STML includes a special link named EVENT_REGION that relates this kind of event to a specific location.

Modeling Motion with STML. When an ISO-TimeML event involves a change of location, it must participate in the STML tag, MOTION_PATH. An STML path is a special kind of region that can have a begin point and an endpoint. Paths can be introduced explicitly in the text as in *John met Mary along the way*, but it is more common for them to be introduced by a motion event and/or a spatial function as in *John walked to the store.* In STML, a motion event is linked to a path while a non-motion event is linked to a region. The semantics of STML with respect to motion is modeled by the Dynamic Interval Temporal Logic, which is discussed in section 3.3 below. Special attention is given to direction and orientation issues that are introduced by certain motion predicates such as *climb* and *take off.*

2.4 Grounding the Annotation

Once spatial information has been identified in a text, the mark-up language includes attributes that can ground locations on a map. For example, the REGION tag includes latitude and longitude values. In order to represent the annotation on a map, we can map to the Keyhole Markup Language (KML)[6], which is Google's file format for displaying geographic data in Google Earth or Google Maps. KML is also based on the XML standard and includes features that allow for placemarks, descriptions of places, ground overlays, paths, and polygons.

Placemarks are used most commonly in KML and can be used for all locations. Take, for example, the location *San Cristobal de Las Casas.* The STML annotation and the corresponding KML representation are shown below:

```
<REGION latLong=‘‘16.73N 92.63W’’> San Cristobal de Las Casas </REGION>

<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Placemark>
    <name>San Cristobal de Las Casas</name>
    <Point>
      <coordinates> -92.633281, 16.738578,0</coordinates>
    </Point>
  </Placemark>
</kml>
```

[6] KML was submitted to the Open Geospatial Consortium (OCG), which adopted KML Version 2.2 as an OGC implementation standard.
See http:/www.opengeospatial.org/standards/kml/.

Fig. 1. Grounded Representation

Opening a file in Google Earth with this code results in a pushpin being located at the San Cristobal coordinates. It is also possible to specify a path in KML, which we would use to represent motion on the map. A path is drawn simply by including a list of coordinates in the KML mapping. A new set of coordinates can be added to the path whenever a distinct location is given in the text. The end result is a grounded representation of the text. Figure 1 shows an example of what such a grounded representation would look like on Google Maps.

3 Modeling Motion in Language

3.1 How Languages Talk about Movement

Understanding motion in language involves more than just identifying the geo-coordinates involved in a motion event, and anchoring an object between these locations. Locations are often implicit and not overtly mentioned; often, non-spatial entities are construed as locations, and hence are not identified by lexicons or gazetteers as location entities. Further, not every predicate used as a motion predicate in the text is tagged as such by a lexical resource. Perhaps most importantly, motion is conveyed in two very different constructions in language, as discussed below, and any interpretive algorithm must recognize the semantic distinction and consequences between these.

[31] is perhaps the first to systematize the observation that languages have distinct strategies for expressing concepts of motion. He noticed that there are essentially two basic constructions associated with the expression of motion: *verb-framed* and *satellite-framed* patterns (subsequent work on this includes [18], [32],

[33]). This is also referred to as the *path* verb vs. *manner-of-motion* verb distinction. The latter strategy (satellite-framing) can be seen in sentences such as:

(3) a. John hopped$_{manner}$ out of the room$_{path}$.
b. Mary crawled$_{manner}$ to the window$_{path}$.

The path (verb-framed) construction is illustrated with the following examples:

(4) a. John arrived$_{path}$ by foot$_{manner}$.
b. John descended$_{path}$ the stairs running$_{manner}$.

We can split languages broadly into the two classes. MANNER CONSTRUCTION LANGUAGES encode *path* information in directional PPs, particles, and other adjuncts, while the main verb encodes the *manner-of-motion*; examples include English, German, Russian, Swedish, and Chinese. PATH CONSTRUCTION LANGUAGES encode *path* information in the matrix verb, while adjuncts optionally specify the *manner-of-motion*; examples include Modern Greek, Spanish, Japanese, Turkish, and Hindi. However, recent work has questioned the earlier claims of languages having uniquely one strategy or another [34], [35], but the observation holds generally as a description for how a language (typically) expresses motion.

As observed in (3) and (4) above, English allows both constructions, and these are common in the travel blog sublanguage. For example, *biking* is a manner verb used in a path PP-construction to indicate direction and path information. The verbs *arrive* and *leave* are both inherently path verbs and give no information regarding the manner-of-motion without further context. This distinction will prove to be very useful for modeling basic motion with qualitative spatial reasoning calculi, as we see in the next section.

3.2 Qualitative Models for Space and Time

Historically, there are two qualitative models that have been used to represent spatial information. The classic Region Connection Calculus [25] models topological relationships using C, the *connected-to* relation. RCC8 consists of eight relations that are jointly exhaustive and pairwise complete: disconnected, externally connected, partial overlap, equal, tangential proper part and its inverse, and non-tangential proper part and its inverse. RCC8 and other systems like it do an adequate job of representing static information about space, but they cannot help us deal with motion, since that task requires a temporal component. Galton [36], [37] discusses a commonsense theory of motion, but this work does not focus on merging temporal and spatial phenomena.

While RCC8 has been used considerably for modeling spatial relations as expressed in language, the 9-Intersection Calculus [38] and the 9^+-Intersection Calculus [39] are more suggestive as a starting point to analyze motion. The 9-Intersection Model for line-region relations [40] is a somewhat more complex system based on the intersections of the interiors, boundaries, and exteriors of two point sets in the following matrix, where R^o represents the region interior, ∂R represents the region boundary, and R^- represents the region exterior:

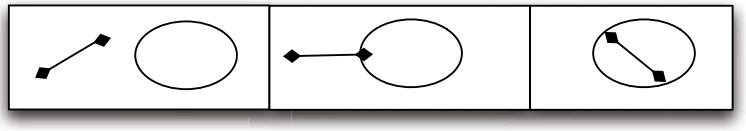


Fig. 2. Three Line-Region Relationships in 9IC

$$(5)\quad I(A,B) = \begin{pmatrix} A^o \cap B^o & A^o \cap \partial B & A^o \cap B^- \\ \partial A \cap B^o & \partial A \cap \partial B & \partial A \cap B^- \\ A^- \cap B^o & A^- \cap \partial B & A^- \cap B^- \end{pmatrix}$$

The chart in Figure 2 shows some of the line-region relationships in 9IC, where the boundary of a line consists of its two endpoints. The middle cell in Figure 2 refers to a line touching the edge of another region, and can be represented with the matrix in (6), corresponding to the RCC8 relation of *external connection* (EC) between two regions, i.e., $EC(L, R)$.

$$(6)\quad \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}^{(LR13)}$$

How can this be applied to the problem of modeling motion as expressed in language? In order to model the motion of an individual, we need at least to identify locations as regions, and the path between two regions as a separate region itself. We also need directionality. The 9-intersection calculus model can be modified to account for motion by distinguishing two boundaries instead of one as follows: $\partial_L L$ (left boundary) and $\partial_R R$ (right boundary). This has, in fact, already been proposed in Kurata and Egenhofer (2007), where the notion of a *directed line* is introduced. Using this model, we can view a line, L, as having two distinct endpoints. When intersected with a region, R, the resulting matrix, I^e, can be defined as the intersection between R and the two-boundaried line L shown below[7]:

$$(7)\quad I^e(L,R) = \begin{pmatrix} L^o \cap R^o & L^o \cap \partial R & L^o \cap R^- \\ \partial_L L \cap R^o & \partial_L L \cap \partial R & \partial_L L \cap R^- \\ \partial_R L \cap R^o & \partial_R L \cap \partial R & \partial_R L \cap R^- \\ L^- \cap R^o & L^- \cap \partial R & L^- \cap R^- \end{pmatrix}$$

We will adopt this extended model of LR-intersections in order to motivate an interpretation of an object in motion, using what we term a *Dynamic Line-Region Intersection* Model. This is, in many respects, in the spirit of [16], where dynamic aspects of spatial change are captured for modeling motion.

[7] The matrix used in the Directed Line-Region Model is slightly different than what we present here. We present the matrix in this way because it is more conducive to modeling motion.

A dynamic LR-Intersection is essentially a directed LR matrix viewed over time. Let us imagine that a specific directed LR-intersection matrix can be viewed as encoding the value of intersective relations from multiple temporal indices (states). These state values are overlays on top of each other. Motion can then be read off the matrix as a *Temporal Trace* (e.g. ordering) of line-region intersection cell values. The "object in motion" is modeled as the topological transformations over the line, indexed through a temporal trace. We assume that the topological relations are interpreted within a temporally indexed world, called a state. States are partially ordered relative to each other, as described in more detail later.

For example, the above relation $LR13$, when viewed as a Directed Line-Region intersection, encodes two path predicates, including *arrive* and *leave*, as shown in (8).

(8) a. $[\![arrive]\!]_{LR13^e}$: $\langle[\partial_L L \cap \partial R = 0]@s_1, [L^o \cap \partial R = 0]@s_2, [\partial_R L \cap \partial R = 1]@s_3\rangle$
b. $[\![leave]\!]_{LR13^e}$: $\langle[\partial_R L \cap \partial R = 1]@s_1, [L^o \cap \partial R = 0]@s_2, [\partial_L L \cap \partial R = 0]@s_3\rangle$

Consider the first two sentences of the text:

(9) a. David left San Cristobal de Las Casas four days ago.
b. He arrived in Ocosingo that day.

We can model both *leave* and *arrive* as path construction predicates denoting Directed Line-Region intersections $LR13^e$, as indicated above. Each introduces a region and a line (path). Ideally, we want the two paths, $P1$ and $P2$, to unify as shown in the figure below. This then denotes that the movement which started with San Cristobal on $P1$, ends at Ocosingo on $P2$, which is identical to $P1$.

Formally, a line-region intersection matrix is a non-dynamic encoding of topological relations. To adequately trace movement, we need to be able to treat the intersective values dynamically, where single values change over time (i.e., a linear state progression). Viewed with time stamps, this corresponds to the motion of the path predicate *arrive*, as illustrated below.

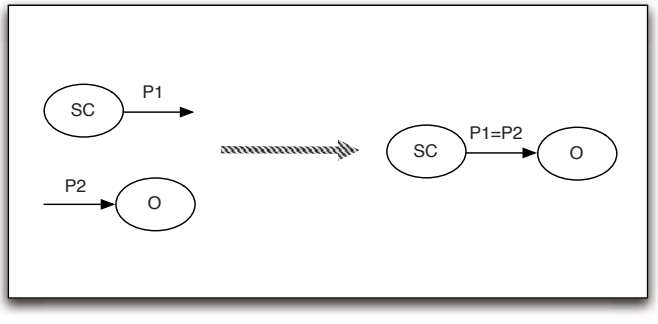


Fig. 3. Two Paths being unified

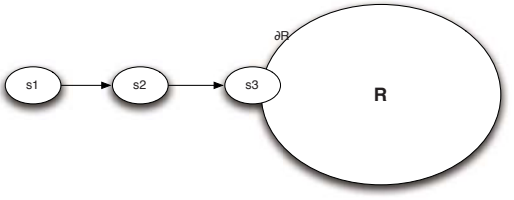


Fig. 4. Dynamic LR-Intersection Model: arrive

The Dynamic LR-Intersection Model can be extended to account for motion involving measurements as well. This allows the model to capture predicates such as *approach* and *pull away* [41], [42]. Similarly, it is easy to see how this model can be extended to include region-region relations involving orientation using Freksa's star calculus or related formalisms [43], [44].

We can use this as the basic model of motion for paths, but it is not the case that all motion verbs will map into this model, since not all motion is expressed through path predicates. Namely, just as frequently, languages employ manner-of-motion verbs with or without path denoting satellite expressions. For example, *walk*, *drive*, and *fly* all denote different modes of transportation, but do not specify begin or endpoints along a path. Another limitation of this model is that maintaining the encoding of dynamic information in the form of state sequences does not scale uniformly to traces of arbitrary length without considerable computational complexity. Finally, it is not obvious how the Dynamic LR Model can be treated compositionally in any obvious sense relative to the linguistic representation from text.

In sum, the present model interprets place descriptions and "moving" objects in a qualitative schematic representation of regions. To account for a natural interpretation of motion, however, we need a natively temporal model that allows for change in the values of relational attributes for an object, such as *location* and *height*. In the next section, we introduce just such a model, *Dynamic Interval Temporal Logic*.

3.3 Dynamic Interval Temporal Logic

As suggested above, in order to adequately model the motion of objects as expressed in language, the representational framework should have two properties: (i) it should be natively temporal; and (ii) it should accommodate change in the assignment of values to the relevant attributes being tracked, e.g., location. These requirements have motivated the development of a hybrid logical framework, which we call *Dynamic Interval Temporal Logic* (DITL). DITL combines a First-order Linear Interval Temporal Logic [45], [28], [46], [47] with an operational semantics that is native to Propositional Dynamic Logic (PDL) [48], Quantified Dynamic Logic (QDL) [49], [50], and Dynamic Predicate Logic (DPL) [51].[8]

[8] Recent work by [52] also works towards providing a more dynamic interpretation for verb meaning within a compositional semantics. The relevance of this work is discussed fully in [42].

For the present discussion, we limit our discussion of the formal mechanisms of the logic to those aspects relevant to modeling the two types of motion constructions introduced in Section 4.1. While we assume the temporal operators normally associated with Linear Temporal Logic (LTL), such as Next ($\bigcirc$), All ($\Box$), Some ($\Diamond$), and Until ($\mathcal{U}$), we avoid their use in the following discussion. It should be pointed out, however, that we interpret our temporal formulae in a discrete, linear model of time. Formally, this structure is represented by $\mathcal{M} = \langle \mathbb{N}, I \rangle$, where $I : \mathbb{N} \mapsto 2^{\Sigma}$ maps each natural number (representing a moment in time) to a set of propositions, where Σ is the set of all atomic propositions.

Returning to the distinction between path and manner constructions, we assume that the underlying semantics for each class of motion predicates is quite distinct. Namely, we assume that predicates making direct reference to a path, such as *arrive* or *exit*, specify explicitly a *distinguished location* along the path which is either reached or departed from, as in the sentence *He arrived in Ocosingo that day.* Manner-of-motion predicates by themselves make no reference to any specific locations at all, as seen in *David biked all day*; they can, however, be used in a distinguished location interpretation by embedding the motion verb within a path construction, as seen in *David biked to Agua Azul.*

Given this distinction, let us flesh out this basic observation about motion predicates in dynamic terms. Dynamic approaches assume that there are two types of expressions: formulae, ϕ, and programs, π. We assume the syntax of PDL [50], where the set of regular programs can be defined as follows:

(10) a. any atomic program is a program;
b. if ϕ is a formula, then $\phi?$ is a test program;
c. if α and β are programs, then $\alpha; \beta$ is a program;
d. if α and β are programs, then $\alpha \cup \beta$ is a program;
e. if α is a program, then α^* is a program;

In addition, for our first-order fragment of dynamic logic, we assume the following:

(11) a. For every program π, we associate a binary relation $[\![\pi]\!] \subseteq S\ X\ S$ called the input/output relation between states (S) from the dynamic line-region model.
b. For every formula ϕ, we associate $[\![\phi]\!] \subseteq S$.
c. The assignment $x := t$ is defined as $[\![x := t]\!] = \{(u, u[x/u(t)]) \mid u \in S\}$.

Recall that we stated above that path verbs designate a distinguished value in the change of location, from one state to another. In the language of first-order dynamic logic, these changes in value are **tested**. That is, the distinguished location is tested against the current location of the object moving, and then retested until the values match. A manner-of-motion verb, on the other hand, involves not a test, but rather a basic *variable assignment* to the attribute value (of location) associated with the moving object. It then iterates this assignment in change of location from state to state, thereby **assigning** and **reassigning** the location value.

The most basic program of motion, a **change-of-location**, is a variable assignment such as $loc(x) := y$. Directed motion, discrete stepwise directed motion, and manner-of-motion are then defined as follows, respectively:

(12) a. $move(x) =_{def}\ loc(x) := y; y := z, scale(p, y), scale(p, z), y < z$
b. $move(x) =_{def}\ loc(x) := y; y := y + 1, scale(p, y)$
c. $move(x, P) =_{def}\ move(x) \wedge manner(P)$

Manner-of-motion predicates always consist of an initial motion and then zero or more iterations of that same motion. So, for example, *walk* is represented as follows[9]:

(13) $[\![walk]\!] = move(x, walk); (move(x, walk)^*$

Some prepositions such as *from* always introduce an assignment at the start of the interpretation of a motion. For example, *John walked* is a simple manner-of-motion predicate, but adding *from* as in *John walked from the store* introduces an initial assignment. These initial assignment prepositions always have the following interpretation:

(14) $prep(x, y) =_{def}\ \lambda y \lambda \pi \lambda x(loc(x) := y; \pi)$

While some prepositions always introduce an assignment, most behave similarly to any motion predicate in that they all consist of a test, some iterated motion, and another test. Example (15a) shows how this is done for motion predicates and (15b) shows how it is generalized to allow for prepositional phrases.

(15) a. $arrive(x, y) =_{def}\ ((\neg loc(x) = y)?; move(x))^+; (loc(x) = y)?$
b. $to(x, y) =_{def}\ \lambda \pi \lambda x((\neg loc(x) = y)?; \pi(x))^+; (loc(x) = y)?$

Figure (5) demonstrates how manner verbs are embedded within a path construction containing a spatial PP. This construction is used for sentences such as those shown in (16). Notice that the manner-of-motion predicates *bike* and *crawl* act to assign and reassign the value of the location of the moving object, while the spatial PPs, *to the ruins* and *to the window*, act as the test, against which the program is checked, until satisfied.

(16) a. John biked$_{manner}$ [to the ruins]$_{path}$.
b. Mary crawled$_{manner}$ [to the window]$_{path}$.

With a path predicate such as *arrive*, the value of the distinguished location is tested inherently by the verb, and the manner is optionally introduced by an adjunct predicate;

(17) a. John arrived$_{path}$ by foot$_{manner}$.
b. John descended$_{path}$ the stairs running$_{manner}$.

[9] We assume a^* denotes the conventional Kleene star operator while a^+ denotes an iteration of one or more.

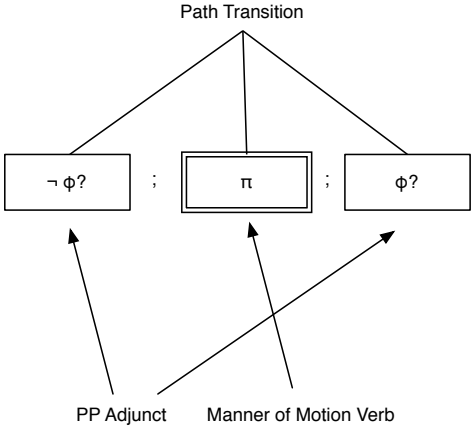


Fig. 5. Manner Verb Construction

We can now represent many different examples of motion in text. Manner-of-motion predicates composed with a path test simply change the manner of the iterated motion that is performed after first testing that the desired goal of the motion has not already been achieved. So, in *John walked to Stanford*, the interpretation is $((\neg loc(j) = s)?; walk(j))^+; (loc(j) = s)?$. That is, first there is a test to make sure that the location of *John* is not already *Stanford*. Then, there is at least one iteration of a *walk*-motion. After each iteration, we test if *John* has reached his goal. If not, we continue to run the *walk* program until he has.

When a manner-of-motion predicate is composed with both an initial assignment preposition and a path test as in *John walked from Menlo Park to Stanford*, the interpretation is the same as the one above except that the location of *John* is first set to *Menlo Park* with the following assignment: $loc(j) = mp$. The interpretation then continues with the usual test if the goal location has been reached.

Initial work on DITL for spatial information is included in [41]. We will continue to develop the logic while focusing on using DITL as a partial semantics for STML. Specifically, DITL must be developed for different kinds of path and manner-of-motion verbs. In addition, it will be important to understand how DITL formulae from a larger discourse compose with each other. Interpreting motion with DITL will also give us the opportunity to reason about the spatiotemporal information present in text for applications such as question answering. Throughout the implementation of the spatial processing algorithms, described in the next section, we examine DITL in particular to see what impact the model has on their success.

4 Implementing Spatial Processing Algorithms

In order to automatically annotate text with STML tags along with an interpretation of the STML, several processing modules are needed: temporal processing,

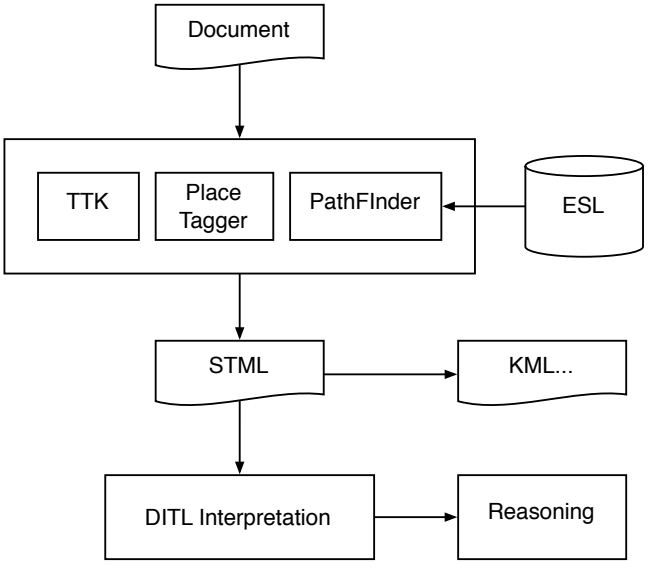


Fig. 6. Representing and Interpreting Spatiotemporal Relations

place tagging, and an algorithm that finds the temporal and spatial paths in a document. The overall layout of the system is given in Figure 6.

4.1 Temporal Processing

The TARSQI Toolkit (TTK, see Figure 7), developed at Brandeis University, is an extensive set of integrated tools for temporal processing of a text [8], [9]. Given our travel blog example in (1), TTK will establish that *left*, *arrived*, *biked*, *played*, *spent*, and *drove* are all events and that they occur in a particular order:

$$left < arrived < biked < played < spent < drove$$

TTK includes modules for document metadata parsing, preprocessing, recognition of events and time expressions, temporal relation parsing, and consistency checking. GUTime is a temporal expression tagger that recognizes the extents and normalized values of time expressions. Evita [53] is a domain-independent tool that performs robust event identification and adds grammatical features, such as tense, aspect, modality, and polarity. Slinket is an application developed to automatically introduce subordinating relations between pairs of events, and classify them into factive, counterfactive, evidential, negative evidential, and modal contexts, based on the modal force of the subordinating event [54].

The temporal parsing stage is comprised of three components that identify temporal relations between events and times known as TLINKs: Blinker is a rule-based component that applies to multiple configurations of events and temporal expressions and creates temporal orderings between them. S2T takes the output

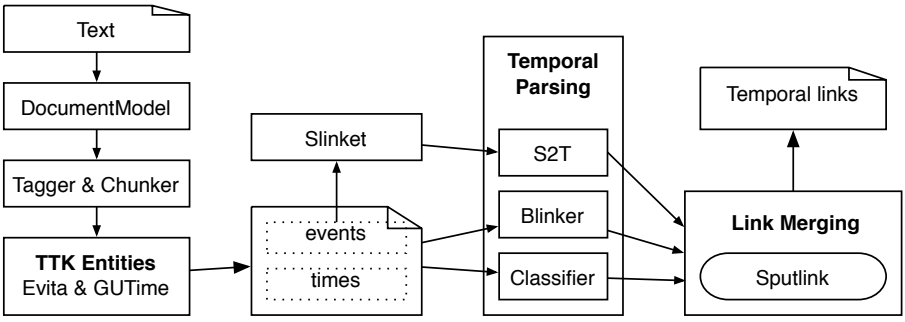


Fig. 7. TARSQI Toolkit

of Slinket and uses around a dozen syntactic rules to map subordinating relations onto TLINKs. The TLINK Classifier [55] is a Maximum Entropy based classifier that generates temporal relations between identified events in text. The classifier accepts its input for each pair of events under consideration as a set of features.

Temporal links generated by the three different components are not necessarily consistent with each other. The link merging component [56], [57] uses a greedy algorithm and a standard constraint propagation algorithm to merge TLinks into a consistent whole, using confidence scores derived from observed precision of rules as well as internally generated scores from the classifier. The result is a consistent annotation where high precision links are preferred over lower precision links.

4.2 Place Finder

To add PLACE tags to a document employ both an existing off-the-shelf place tagger (i.e. [1]) as well as a trained machine learning model on annotated text using, e.g., Maximum Entropy classifiers[10]. We employ a hybrid approach where gazetteer lookup proposes the candidates for PLACE-hood and a statistical model trained on the SpatialML corpus[11] disambiguates between true places and other types. An additional task will be to add functionality for detecting and tagging Spatial Functions and Implicit Locations from text.

4.3 Event Structure Lexicon Resource

We take advantage of a lexical resource called the Event Structure Lexicon (ESL) [13], which encodes subevent predicate information for verbs of motion and thereby effectively acts as an additional markup on top of TimeML. The ESL is a library of context-dependent event structures for verbs consisting of

[10] http://sourceforge.net/projects/carafe/
[11] http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2008T03

an event type, a list of subevents, a verb class specification, a subcategorization frame, and specification of semantic roles for arguments. Predicative content is decomposed into subevents and their temporal ordering, along with headedness. The basic event types are process, state, and transition (achievement and accomplishment). A transition consists of pre-state, process, and result state (post-state).

The verb classes and subclasses are based on the Brandeis Semantic Ontology (BSO) [58]. Each of the subclasses has its own event structure frame assigned. The upper level verb class consists of process, state, change_of_location, change_of_possession, and change_of_state. The change_of_location class can be divided into from_source_to_goal, from_source, to_goal, etc., which are being developed in the broader context of modeling motion in language [59]. The event structure frame for the change_of_location class is shown below, where `pred` is the verb assigned to the class:

(18) pred: change_of_location(x,y)
se1: pre-state: not_be_in(x,y)
se2: process: pred-ing(x,y)
se3: post-state: be_in(x,y)
temporal ordering:
[se1 PRECEDES and OVERLAPS se2] & [se3 ENDS se2]

Associating this frame with the verb *bike* in *David biked to Agua Azul* allows us to derive that the process of biking ended the state of not being in Agua Azul.

4.4 PathFinder Algorithm

PathFinder is a Python framework, developed in the context of our previous NGA-funded research, that provides abstractions for writing concise, declarative rules for processing motion events identified from TimeML together with spatial descriptions from SpatialML. The rules consist of *patterns*, which are based on grammatical relations provided by a dependency parse, a lexicon that specifies the motion classes of the verbs of interest, and the types of the various objects in the document tree, and associated *actions*, which are arbitrary function-like expressions to be run when the pattern is matched.

Specialized readers were developed for TimeML, SpatialML, dependency parses, and an entity reference schema with anaphora. These are then combined into the STM document reader (for spatio-temporal markup, an early version of STML), over which the motion classes and rules discussed above operate. The rules themselves are stored in class slots of classes representing the motion classes derived from [59]. The objects over which the rules apply are of the appropriate motion classes, but the primary purpose of the actions is to create and initialize *paths* and *traversals* that represent the specific motion of an object. Paths have beginnings and ends; traversals also have beginnings and endings (which will always be times), and are

of a specific path and *by* a set of traversers, and may optionally have a duration. For example, take the sentence (markup heavily abbreviated for clarity):

```
<PERSON>John</PERSON> <EVENT>traveled</EVENT> from <PLACE>Chicago</PLACE>
to <PLACE>Boston</PLACE> on <TIMEX3>Monday</TIMEX3>
```

The motion class to which 'travel' is assigned has a pattern defined on it that would create a *path* whose beginning is the *place* 'Chicago', whose end is the *place* 'Boston', and would also create a traversal of that path with a single traverser, the *person* 'John', whose beginning was 'Monday'.

The initial prototypes and ideas for both ESL and PathFinder were courtesy of the NSF project *Inferring Spatio-Temporal Trajectories of Entities from Natural Language Documents*. There, the immediate concern was to (i) integrate TimeML and SpatialML tags, (ii) create a small lexicon of motion verbs, and (iii) design a component to create paths using the TimeML/SpatialML tags and lexical information. For this project, we propose to take a subset of ESL and embed it in PathFinder as a new lexicon. In addition, PathFinder will need to be adapted to create the richer STML annotations, while incorporating the semantics provided by DITL.

4.5 Corpus Annotation and Algorithm Evaluation

We have collected a small corpus of travel blogs, called TravelBank. We are currently semi-automatically annotating about 25K tokens of TravelBank with the TARSQI Toolkit and the Place Tagger, using manual correction where needed. We are then manually annotating the corpus with STML tags. This work is still in progress, and is expected to be a cyclical affair where annotation is driven by the process of creating the STML representation and where annotation provides feedback due to the confrontation with real data.

The main use of the annotated corpus is as a gold standard against which we will evaluate the performance of the PathFinder component. TravelBank will also be used to evaluate the Place Tagger and any changes we made to the TARSQI Toolkit. TARSQI Toolkit changes will also be evaluated against TimeBank.

Acknowledgments

Portions of this work are presented in a current submission to the Journal of *Spatial Cognition and Computation*. We would like to thank Anna Rumshisky, for her input and help in the preparation of this work, as well as Inderjeet Mani for useful discussion. Portions of this work were presented at the 2009 Stanford Workshop on Spatial Relations, the 2009 AAAI Spring Symposium on Benchmarking Qualitative Spatiotemporal Systems, as well as the 2009 COSIT Conference in Aber W'rach, France. All remaining errors are, of course, the responsibility of the authors. This work was supported in part by a grant to J. Pustejovsky by NGA HM1582-07-1-2037.

References

1. Mani, I., Hitzeman, J., Richer, J., Harris, D., Quimby, R., Wellner, B.: Spatialml: Annotation scheme, corpora, and tools. In: Chair, N.C.C., Choukri, K., Maegaard, B., Mariani, J., Odjik, J., Piperidis, S., Tapias, D. (eds.) Proceedings of the Sixth International Language Resources and Evaluation (LREC 2008), Marrakech, Morocco, European Language Resources Association, ELRA (2008), http://www.lrec-conf.org/proceedings/lrec2008/
2. Mani, I., Wilson, G.: Robust temporal processing of news. In: Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL 2000), New Brunswick, New Jersey, pp. 69–76 (2000)
3. Mani, I., Wilson, G., Sundheim, B., Ferro, L.: Guidelines for annotating temporal information. In: Proceedings of HLT 2001, First International Conference on Human Language Technology Research (2001)
4. Grishman, R., Sundheim, B.: Message understanding conference-6: A brief history. In: Proc. International conference on computational linguistics (1996)
5. Bikel, D., Miller, S., Schwartz, R., Weischedel, R.: Nymble: a high-performance learning name-finder. In: Proc. Conference on applied natural language processing (1997)
6. Etzioni, O., Cafarella, M., Downey, D., Popescu, A.M., Shaked, T., Soderland, S., Weld, D., Yates, A.: Unsupervised named-entity extraction from the web: an experimental study. Artificial Intelligence 165, 91–134 (2005)
7. Sekine, S., Nobata, C.: Definition, dictionaries and tagger for extended named entity hierarchy. In: Proc. Conference language resources and evaluation (2004)
8. Verhagen, M., Mani, I., Sauri, R., Knippen, R., Jang, S., Littman, J., Rumshisky, A., Phillips, J., Pustejovsky, J.: Automating Temporal Annotation with TARSQI. In: Interactive Poster and Demo Session. Proceedings of the ACL 2005 (2005)
9. Verhagen, M., Pustejovsky, J.: Temporal processing with the TARSQI toolkit. In: Coling 2008: Companion volume: Demonstrations, Manchester, UK, Coling 2008 Organizing Committee, pp. 189–192 (2008)
10. Pustejovsky, J., Castano, 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), www.timeml.org
11. Pustejovsky, J., Knippen, R., Littman, J., Saurí, R.: Temporal and event information in natural language text. Language Resources and Evaluation 39, 123–164 (2005)
12. MITRE: Spatialml: Annotation scheme for marking spatial expressions in natural language (2007), http://sourceforge.net/projects/spatialml/
13. Im, S., Pustejovsky, J.: Annotating event implicatures for textual inference tasks. In: GL 2009: 5th International Conference on Generative Approaches to the Lexicon (2009)
14. Muller, P.: A qualitative theory of motion based on spatio-temporal primitives. In: Cohn, A.G., Schubert, L., Shapiro, S.C. (eds.) KR 1998: Principles of Knowledge Representation and Reasoning, pp. 131–141. Morgan Kaufmann, San Francisco (1998)
15. Galton, A.P.: Qualitative Spatial Change. Oxford University Press, Oxford (2000)
16. Hornsby, K.S., Cole, S.J.: Modeling moving geospatial objects from an event-based perspective. T. GIS 11, 555–573 (2007)
17. Talmy, L.: How language structures space. In: Pick, H., Acredolo, L. (eds.) Spatial Orientation: Theory, Research, and Application. Plenum Press, New York (1983)
18. Jackendoff, R.: Semantics and Cognition. MIT Press, Cambridge (1983)
19. Herskovits, A.: Language and Spatial Cognition: an Interdisciplinary Study of the Prepositions in English. Cambridge University Press, Cambridge (1986)

20. Asher, N., Sablayrolles, P.: A typology and discourse for motion verbs and spatial pps in french. Journal of Semantics 12, 163–209 (1995)
21. Boas, H.C.: Frame semantics as a framework for describing polysemy and syntactic structures of english and german motion verbs in contrastive computational lexicography. In: Rayson, P., Wilson, A., McEnery, T., Hardie, A., Khoja, S. (eds.) Proceedings of the corpus linguistics 2001 conference, Lancaster, UK, vol. 13. University Center for Computer corpus research on language (2001)
22. Cappelle, B., Declerck, R.: Spatial and temporal boundedness in english motion events. Journal of Pragmatics 37, 889–917 (2005)
23. Asher, N., Vieu, L.: Towards a geometry of common sense: a semantics and a complete axiomatisation of merotopology. In: Proceedings of IJCAI 1995 (1995)
24. Cohn, A.G., Renz, J.: Qualitative spatial representation and reasoning. An Overview. Fundamenta Informaticae 46, 1–2 (2001)
25. Randell, D., Cui, Z., Cohn, A.: A spatial logic based on regions and connections. In: Kaufmann, M. (ed.) Proceedings of the 3rd International Conference on Knowledge Representation and Reasoning, San Mateo, pp. 165–176 (1992)
26. Wolter, F., Zakharyaschev, M.: Spatio-Temporal Representation and Reasoning based on RCC-8. In: KR 2000: Principles of Knowledge Representation and Reasoning, pp. 3–14 (2000)
27. Mani, I., Hitzeman, J., Clark, C.: Annotating natural language geographic references. In: Workshop on Methodologies and Resources for Processing Spatial Language, LREC 2008 Conference, Marrakech, Morocco (2008)
28. Allen, J.: Towards a general theory of action and time. Arificial Intelligence 23, 123–154 (1984)
29. Boguraev, B., Pustejovsky, J., Ando, R., Verhagen, M.: Timebank evolution as a community resource for timeml parsing. Language Resources and Evaluation 41, 91–115 (2007)
30. Bateman, J.A., Hois, J., Ross, R., Tenbrink, T.: A linguistic ontology of space for natural language processing. Artificial Intelligence (2010)
31. Talmy, L.: Lexicalization patterns: Semantic structure in lexical forms. In: Shopen, T. (ed.) Language typology and semantic description. Grammatical categories and the lexicon, vol. 3, pp. 36–149. Cambridge University Press, Cambridge (1985)
32. Talmy, L.: Towards a cognitive semantics. MIT Press, Cambridge (2000)
33. Choi, S., Bowerman, M.: Learning to express motion events in english and korean: The influence of language-specific lexicalization patterns. Cognition 41, 83–121 (1991)
34. Narasimhan, B.: Motion events and the lexicon: a case study of hindi. Lingua 113, 123–160 (2003)
35. Hovav, M.R., Levin, B.: Reflections on manner/result complementarity. In: Doron, E., Hovav, M.R., Sichel, I. (eds.) Syntax, Lexical Semantics, and Event Structure, pp. 21–38. Oxford University Press, Oxford (2010)
36. Galton, A.P.: Towards an integrated logic of space, time, and motion. In: Bajcsy, R. (ed.) Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (IJCAI 1993), San Mateo, CA, pp. 1550–1555. Morgan Kaufmann, San Francisco (1993)
37. Galton, A.P.: Space, time and movement. In: Stock, O. (ed.) Spatial and Temporal Reasoning, pp. 321–352. Kluwer, Dordrecht (1997)
38. Egenhofer, M., Franzosa, R.: Point-set topological spatial relations. International Journal of Geographical Information Systems 5, 161–174 (1991)

39. Kurata, Y., Egenhofer, M.: The 9+ intersection for topological relations between a directed line segment and a region. In: Gottfried, B. (ed.) Workshop on Behaviour and Monitoring Interpretation, Germany, pp. 62–76 (2007)
40. Egenhofer, M., Mark, D.: Modeling conceptual neighborhoods of topological line-region relations. International Journal of Geographical Information Systems 9, 555–565 (1995)
41. Pustejovsky, J., Moszkowicz, J.: The qualitative spatial dynamics of motion. The Journal of Spatial Cognition and Computation (submitted)
42. Pustejovsky, J.: Measuring change in language. Synthese (submitted)
43. Freksa, C.: Using orientation representation for qualitative spatial reasoning. In: Frank, A., Campari, I., Formentini, U. (eds.) Theories and Methods of Spatio- temporal Reasoning in Geographic Space: Proceedings of the International Conference GIS - From Space to Territory, Pisa, Italy, pp. 162–178 (1992)
44. Mitra, D.: Modeling and reasoning with star calculus: An extended abstract. In: Eighth International Symposium on AI and Mathematics (2004)
45. Kröger, F., Merz, S.: Temporal Logic and State Systems. Springer, Heidelberg (2008)
46. Moszkowski, B.: Executing Temporal Logic Programs. Cambridge University Press, Cambridge (1986)
47. Manna, Z., Pnueli, A.: Temporal Verification of Reactive Systems: Safty. Springer, Heidelberg (1995)
48. Harel, D.: Dynamic logic. In: Gabbay, M., Gunthner, F. (eds.) Handbook of Philosophical Logic (1984)
49. Goldblatt, R.: Logics of Time and Computation, 2nd edn. CSLI Lecture Notes, vol. 7 (1992)
50. Harel, D., Kozen, D., Tiuyn, J.: Dynamic Logic, 1st edn. The MIT Press, Cambridge (2000)
51. Groenendijk, J., Stokhof, M.: Dynamic predicate logic. Linguistics and Philosophy 14, 39–100 (1990)
52. Naumann, R.: Aspects of changes: a dynamic event semantics. Journal of semantics 18, 27–81 (2001)
53. Saurí, R., Verhagen, M., Pustejovsky, J.: Annotating and recognizing event modality in text. In: Proceedings of of the 19th International FLAIRS Conference, FLAIRS 2006, Melbourne Beach, Florida, USA (2006)
54. Saurí, R., Verhagen, M., Pustejovsky, J.: SlinkET: A partial modal parser for events. In: Proceedings of LREC 2006, Genoa, Italy (2006)
55. Mani, I., Verhagen, M., Wellner, B., Lee, C.M., Pustejovsky, J.: Machine learning of temporal relations. In: Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Sydney, Australia, pp. 753–760. Association for Computational Linguistics (2006)
56. Verhagen, M.: Temporal Closure in an Annotation Environment. Language Resources and Evaluation 39, 211–241 (2005)
57. Mani, I., Wellner, B., Verhagen, M., Pustejovsky, J.: Three approaches to learning TLINKs in timeml. Technical Report CS-07-268, Brandeis University, Waltham, United States (2007)
58. Pustejovsky, J., Havasi, C., Saurí, R., Hanks, P., Rumshisky, A., Litman, J., Castaño, J., Verhagen, M.: Towards a Generative Lexical resource: The Brandeis Semantic Ontology. In: Language Resources and Evaluation Conference, LREC 2006, Genoa, Italy (2006)
59. Pustejovsky, J., Moszkowicz, J.L.: Integrating motion predicate classes with spatial and temporal annotations. In: Proceedings of COLING 2008, Manchester, UK (2008)

Flexible Disambiguation in DTS

Livio Robaldo and Jurij Di Carlo

Department of Computer Science, University of Turin, Italy
robaldo@di.unito.it, jurij.dicarlo@educ.di.unito.it

1 Introduction

Quantifier scope ambiguities may engender several logical readings of a NL sentence. For instance, sentence (1) yields six possible readings, depending on the scoping of its three quantifiers: (∀5∃), (∀∃5), (∃∀5), (∃5∀), (5∀∃), and (5∃∀).

(1) Every$_x$ manager showed five$_y$ representatives a$_z$ sample. [5]

In (∀5∃), for every manager there are five different representatives to whom he showed a different sample each. Also in (∀∃5) there are five different representatives for every manager, but he showed the same sample to each of them, etc.

In many real cases the knowledge needed to disambiguate is not fully available during the processing of the sentence. In such cases, all readings should be stored; afterwards, when new world knowledge becomes available, they may be sequentially checked in order to remove those that became inconsistent with it.

In order to provide a flexible treatment of semantic ambiguities, as quantifier-scope ambiguities, *Underspecified* semantic formalisms have been proposed. They allow to encapsulate scope ambiguities into a single compact structure that may be specified afterwards into one of the readings it refers to. A popular approach to Underspecification is grounded on dominance constraints between certain scope-bearers and certain scope-arguments. Underspecified Discourse Representation Theory [14] [15], Hole Semantics [2], Minimal Recursion Semantics [4], and Dependency Tree Semantics (DTS, henceforth) [16] belong to this approach.

As argued by [5], constraint-based formalisms fail to be *Expressively Complete*, i.e. able to produce all possible refiniments of the initial ambiguous expression. If an NL sentence yields n readings, an expressively complete underspecified logic is able to provide a formula for each of the 2^n subsets of those readings.

For instance, Ebert showed that it is not possible to assert, in any of the formalisms mentioned above, an underspecified formula for (1) that excludes (∃5∀) only, i.e. that refers to the following subset of five readings only:

(2) $$\{(5\forall\exists), (5\exists\forall), (\exists\forall 5), (\forall\exists 5), (\forall 5\exists)\}$$

Of course, Ebert does not claim that it is impossible to build a semantic representation referring to any subset of readings. It is always possible to do so; in the worst case, we simply built the disjunction of those readings. In the example under examination, we could set up the "underspecified" representation as:

(3) $$(5\forall\exists) \vee (5\exists\forall) \vee (\exists\forall 5) \vee (\forall\exists 5) \vee (\forall 5\exists)$$

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 257–268, 2010.

But, as pointed out above, underspecified logics have been designed precisely to avoid such a solution. We would like to work with *compact* formulae, in order to control spatial and temporal Complexity. Ebert's point of view is that Expressive Completeness and Compactness are in trade-off of one another.

[17] empirically showed that upon consideration of the trade-off between Expressive Completeness and Compactness/Complexity, we must definitely choose Expressive Completeness. Accordingly, they proposed an extension of DTS that allows to specify any subset of readings. The spatial complexity, of course, becomes exponential, but the authors showed that in real cases exponential combinatorial explosion occurs very rarely. In this paper, we present a modified version of [17] that preserves Expressive Completeness and allows for the definition of flexible procedures to achieve incremental disambiguation.

2 Dependency Tree Semantics (DTS)

DTS [16] is an underspecified formalism for dealing with quantifier scope ambiguity. Fully-underspecified DTS structures are based on a graph G representing the predicate-argument relations. The nodes of G are either predicates or variables called discourse referents. Predicates connect discourse referents via arcs labeled with the argument position. Each variable is also associated with a quantifier, via a function quant, and with a restriction, via a function restr from discourse referents to subgraphs of G. The fully-underspecified representation of (1) is:

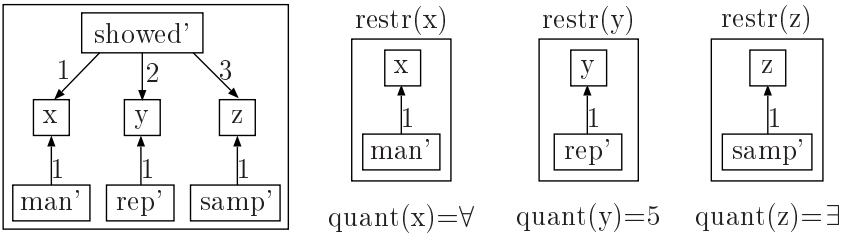


Fig. 1. DTS fully-underspecified representation of sentence (1)

In order to make the dependencies among sets of entities explicit, another kind of arcs is introduced, termed SemDep arcs, and resemble Skolem dependencies. A discourse referent is taken to depend on all discourse referents it is connected via a SemDep arc. Moreover, G includes a special element called Ctx. Ctx refers to the context, i.e. the domain wrt which the final structure will be evaluated. All discourse referents are linked to Ctx via a SemDep arc; however, the ones linked to Ctx *only* are assumed to denote fixed sets of entities, i.e. to correspond to Skolem constants. The several readings of a sentence differ in the set of SemDep

arcs only. For instance, the *seven* readings DTS accepts for (1) correspond to the following configurations of SemDep arcs[1]:

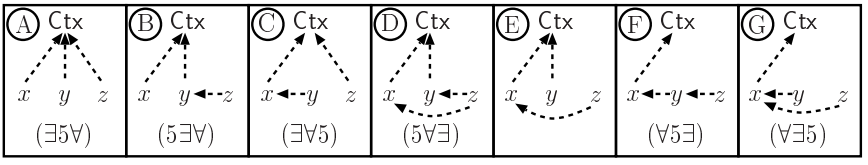


Fig. 2. The seven readings of sentence (1)

Reading (E) is not accepted by Ebert nor by the other proposals cited above as it does not correspond to any linear scope-order. However, (E) seems to be acceptable in NL; consider: *(During the final exam,) every$_x$ student will show five$_y$ professors a$_z$ project.* In this sentence, isomorphic to (1), it is likely that the five professors form a committee, and so do not vary on the students, and that the project is different from student to student, i.e. that z depends on x.

2.1 Positive and Negative Arcs

We illustrate here how the disambiguation process may be carried out, e.g. how to obtain the fully specified readings in fig.2 from the structure in fig.1. In [17], we proposed to introduce in DTS other two kinds of arc, termed Positive and Negative arcs, that respectively mark allowed and disallowed dependencies. They may be formalized into a set PN of constraints in the form:

(4) $$\{n_{11}{\rightarrow}n_{12},\ldots,n_{i1}{\rightarrow}n_{i2}\,|\,p_{11}{\rightarrow}p_{12},\ldots,p_{j1}{\rightarrow}p_{j2}\}$$

(4) is termed a 'Positive|Negative set'. $n_{11}{\rightarrow}n_{12},\ldots,n_{i1}{\rightarrow}n_{i2}$ are said 'Negative arcs' and $p_{11}{\rightarrow}p_{12},\ldots,p_{j1}{\rightarrow}p_{j2}$ as 'Positive arcs'. (4) specifies that it is not possible to generate a DTS structure that includes the SemDep arcs $n_{11}{\rightarrow}n_{12},\ldots,n_{i1}{\rightarrow}n_{i2}$, unless it also includes the arcs $p_{11}{\rightarrow}p_{12},\ldots,p_{j1}{\rightarrow}p_{j2}$. Either one of the two subsets of arcs may be empty, leading to the two limit cases. They are respectively called 'Positive set' (5.a) and 'Negative set' (5.b).

(5)
a. $\{\,|\,p_{11}{\rightarrow}p_{12},\ldots,p_{j1}{\rightarrow}p_{j2}\}$
b. $\{n_{11}{\rightarrow}n_{12},\ldots,n_{i1}{\rightarrow}n_{i2}\,|\,\}$

At the beginning of the disambiguation process, all dependencies are asserted in PN as Positive sets. For example fig.1 is augmented with the following PN:

(6) PN=$\{\{\,|\,A{\rightarrow}\mathsf{Ctx}\}, \{\,|\,x{\rightarrow}y\}, \{\,|\,y{\rightarrow}x\}, \{\,|\,x{\rightarrow}z\}, \{\,|\,z{\rightarrow}x\}, \{\,|\,y{\rightarrow}z\}, \{\,|\,z{\rightarrow}y\}\}$

[1] Transitive dotted arcs are not shown in the graphical representations, but they *do* occur (cf. [16], §2.3). For instance in reading (F), z depends on both x, y and Ctx.

$A{\rightarrow}\mathsf{Ctx}$, where A stands for 'All', is a special positive arc that links all discourse referents to Ctx. At each step, it is possible to either select a non-Negative set from PN and assert all its arcs as SemDep arcs (such an operation is called 'conversion of a non-Negative set'), or adding new constraints, i.e. asserting a new non-Positive set in PN. Since DTS structures are model-theoretically interpretable iff all discourse referents are connected to Ctx, we cannot interpret them until $\{|A{\rightarrow}\mathsf{Ctx}\}$ is not converted. After that, each DTS structure that may be generated via PN corresponds to a fully-disambiguated reading.

Let's see an example of how a new constraint is added in PN. Sentence (1) includes a universal and an existential quantifier. As discussed in [3], [10], and [19], those two quantifiers (as well as definites, possessives, etc. cf. [1]), engender redundant structures. Since universals range on the whole domain of individuals, they cannot exhibit any dependency on another quantifier[2]. The truth conditions of reading (A) in fig.2 would be the same if x would depend on y and/or z: x must denote the set of all representatives in any case. Similarly, no SemDep arc can enter an existential quantifier, as it cannot induce variation on its dependents. For those reasons, in fig.2 no arc exits x (but the one to Ctx) and no arc enters z. Accordingly, we must insert all the corresponding Negative sets in the initial PN in (6). Obviously, the insertion of those sets must consistently trigger the deletion of the Positive sets that lead to them. PN turns out to be:

(7) $PN{=}\{\{|A{\rightarrow}\mathsf{Ctx}\}, \{|y{\rightarrow}x\}, \{|z{\rightarrow}x\}, \{|z{\rightarrow}y\}\}, \{x{\rightarrow}y|\}, \{x{\rightarrow}z|\}, \{y{\rightarrow}z|\}\}$

The four Positive sets left in PN (on the left) generate all and only the readings in fig.2. This may be easily seen by ordering them from the stronger, i.e. (A), the one satisfied by less models, to the weaker(s), as shown in fig.3.

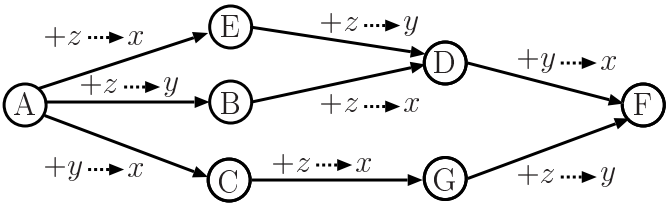


Fig. 3. Arrangement of the seven readings of sentence (1)

Starting from the strongest reading (A), obtained by converting $\{|A{\rightarrow}\mathsf{Ctx}\}$, we may add either one of the Positive sets in (7), obtainig (E), (B), and (C). Those are weaker than (A) in the sense that they allow a set of entities to vary on the entities in another set, and so the set of models satisfying (A) is included in the sets of models satisfying them. From each of those readings, by converting one or both the remaining Positive sets we get (D), (G), and (F) (which is the weakest).

The method makes DTS expressively complete. For instance, in order to exclude (A), we prevent $A{\rightarrow}\mathsf{Ctx}$ unless at least another dependency is established.

[2] With some exceptions concerning Inverse Linking constructions [12].

This is achieved by modifying (7) as in (8.a). On the other hand, in order to exclude (A) *and* (D), we assert PN as in (8.b). (8.b) is obtained from (8.a) by adding the Positive|Negative set $\{z{\rightarrow}x,\ z{\rightarrow}y\,|\,y{\rightarrow}x\}$, that prevents (D) but allows for (F).

(8) a. $PN{=}\{\{\,|\,A{\rightarrow}\mathsf{Ctx},\ y{\rightarrow}x\},\ \{\,|\,A{\rightarrow}\mathsf{Ctx},\ z{\rightarrow}x\},\ \{\,|\,A{\rightarrow}\mathsf{Ctx},\ z{\rightarrow}y\}\},$
$\{x{\rightarrow}y\,|\,\},\ \{x{\rightarrow}z\,|\,\},\ \{y{\rightarrow}z\,|\,\}\}$

b. $PN{=}\{\{\,|\,A{\rightarrow}\mathsf{Ctx},\ y{\rightarrow}x\},\ \{\,|\,A{\rightarrow}\mathsf{Ctx},\ z{\rightarrow}x\},\ \{\,|\,A{\rightarrow}\mathsf{Ctx},\ z{\rightarrow}y\}\},$
$\{z{\rightarrow}x,\ z{\rightarrow}y\,|\,y{\rightarrow}x\},\ \{x{\rightarrow}y\,|\,\},\ \{x{\rightarrow}z\,|\,\},\ \{y{\rightarrow}z\,|\,\}\}$

As discussed above, Expressively Completeness prevents Compactness. In order to identify one of the 2^n subsets of readings, in many cases it is necessary to build a Positive|Negative set for each of them, thus creating an underspecified representation that correspond to their disjunction, as in (3). However, in [17] we empirically shown that, by implementing the constraints on logical redundancy as in (7), the ones on Nested Quantifiers, and standard Island Constraints (see below), in the (few) worst cases PN contains at most forty Positive or Negative arcs, a number of arcs which is clearly affordable by a real system.

3 Procedures for the Proper Management of *PN*

The previous section illustrated the mechanism proposed to make DTS Expressively Complete. It has been pointed out that, after the convertion of a set in PN or after the assertion of a new one, PN needs to be updated. However, it has not been explained how. The updating has to obey the following criteria:

(9) a. **Consistency**: Positive and Negative sets have to be globally consistent. Hence, we must disallow all non-Negative sets that lead to a disallowed pattern of dependencies. Furthermore, cycles on SemDep have to be prevented, as they would describe a set of entities that varies on the entities in another set and vice-versa, which is clearly paradoxal.

b. **Allowed/Disallowed dependencies that are already converted**: it is useless to keep Positive arcs that are already asserted in SemDep. Similarly, preventing a set N_s of Negative arcs, such that some arcs $\{n_1,\ldots,n_m\}\subseteq N_s$ already occur in SemDep is equivalent to preventing $N\setminus\{n_1,\ldots,n_m\}$. Therefore, after a convertion, we must remove all such arcs from PN.

c. **Disallowed dependencies that are already not allowed**: Disallowing a dependency is obviously equal to not allowing it. For instance, the Negative sets $\{x{\rightarrow}y\,|\,\}$, $\{x{\rightarrow}z\,|\,\}$, $\{y{\rightarrow}z\,|\,\}$ in (7) are useless because no combination of sets in PN may cover them. Accordingly, they may be actually removed. However, a Positive|Negative set whose Negative part cannot be covered anymore, cannot be simply removed. We must also add the Positive set corresponding to the union of all its arcs. E.g., suppose that it is no longer possible to cover the Negative part of $\{z{\rightarrow}x,\ z{\rightarrow}y\,|\,y{\rightarrow}x\}$ in (8.b). Then we can remove it from PN, *but* we must also add the Positive set $\{\,|\,z{\rightarrow}x,\ z{\rightarrow}y,\ y{\rightarrow}x\}$, otherwise reading (F) is blocked.

Consistency criterion seems easy to understand, but it actually needs some further explainations. For instance, suppose that a DTS structure involves three discourse referents x, y, and z, and its PN includes the following sets:

(10) $\{x{\rightarrow}y\,|\,\}, \{\,|\,x{\rightarrow}y,\ x{\rightarrow}z\}, \{x{\rightarrow}y\,|\,y{\rightarrow}z\}$

The first Negative set refers to reading (H) in fig.4, which is the strongest therein: (H) and all the readings weaker than it, i.e. those in the dotted area, must be blocked. Consequently, the second Positive set and the third Positive|Negative set in (10), which enables reading (J) and (I), must be removed from PN.

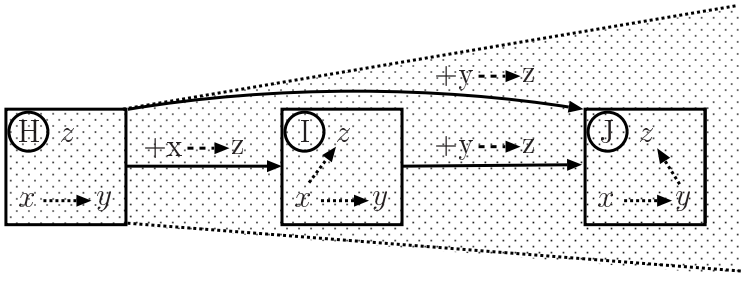


Fig. 4. Consistency checking: $\{\,|\,x{\rightarrow}y,\ x{\rightarrow}z\}$ and $\{x{\rightarrow}y\,|\,y{\rightarrow}z\}$ must be removed

Instead, whenever the strongest reading is denoted by a Positive|Negative set, we must disallow all readings including all its Negative arcs, *unless* they do not also contain all its Positive arcs. For instance, suppose PN includes the sets in (11), the first of which prevents reading (H) but allows for (I), as shown in fig.5.

(11) $\{x{\rightarrow}y\,|\,x{\rightarrow}z\}, \{\,|\,x{\rightarrow}z,\ z{\rightarrow}y\}, \{x{\rightarrow}y\,|\,z{\rightarrow}y\}, \{\,|\,x{\rightarrow}z,\ z{\rightarrow}y\}$

In this case, $\{\,|\,x{\rightarrow}z,\ z{\rightarrow}y\}$ enables reading (K) and must be allowed in that it satisfies $\{x{\rightarrow}y\,|\,x{\rightarrow}z\}$. Instead the other two Positive(|Negative) sets in (11) must be disallowed as they respectively refer to (L) and (J).

3.1 The Procedures *Convert* and *Constrain*

This section presents the procedures *Convert* and *Constrain* that respectively convert a non-Negative set in PN into SemDep arcs and add a new set in PN. Firstly, we define three convenient subprocedures that respectively compute the transitive closure of SemDep wrt a new added arc, add each arc of a non-Negative set to SemDep, and remove from a Positive|Negative set all arcs occurring in SemDep (cf. (9.b)). N_D is the set of discourse referents occurring in the structure. The three subprocedures are shown below in (12).

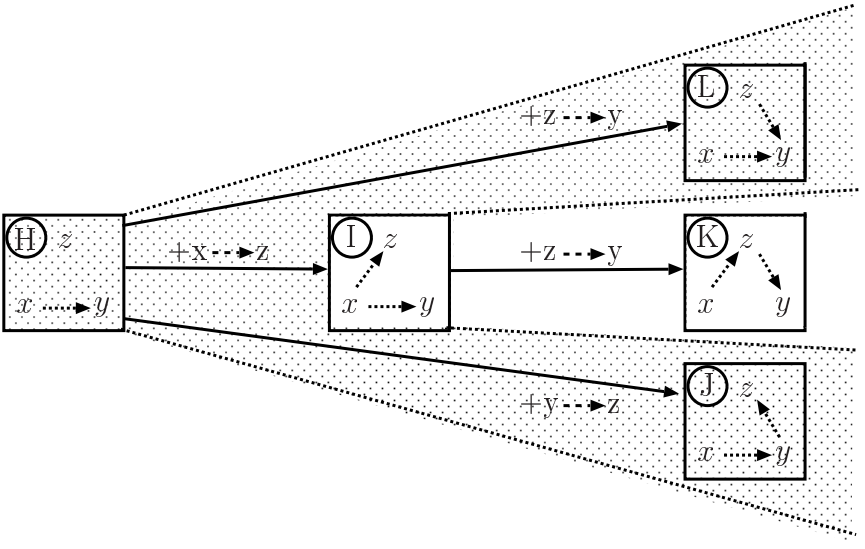


Fig. 5. Consistency checking: $\{x{\rightarrow}y\,|\,z{\rightarrow}y\}$ and $\{\,|\,x{\rightarrow}z,\ z{\rightarrow}y\}$ must be removed

(12) *TransitiveClosure*(SemDep, $x{\rightarrow}y$)

1. for each $d \in N_D$ do
2. if ($(d == x)$ or $(d{\rightarrow}x \in \mathsf{SemDep})$) then
3. set SemDep as $\mathsf{SemDep} \cup \{d{\rightarrow}y\}$
4. for each $y{\rightarrow}d' \in \mathsf{SemDep}$ do
5. set SemDep as $\mathsf{SemDep} \cup \{d{\rightarrow}d'\}$
6. return SemDep

Add(SemDep, $\{n_{11}{\rightarrow}n_{12}, \ldots|p_{11}{\rightarrow}p_{12}, \ldots\}$)

1. for each $a{\rightarrow}b \in \{n_{11}{\rightarrow}n_{12}, \ldots|p_{11}{\rightarrow}p_{12}, \ldots\}$ do
2. set SemDep as *TransitiveClosure*(SemDep, $a{\rightarrow}b$)
3. return SemDep

Compact(SemDep, $\{n_{11}{\rightarrow}n_{12}, \ldots|p_{11}{\rightarrow}p_{12}, \ldots\}$)

1. return $\{\{n_{11}{\rightarrow}n_{12}, \ldots\} \cap \mathsf{SemDep}\}|\{p_{11}{\rightarrow}p_{12}, \ldots\} \cap \mathsf{SemDep}\}\}$

In order to convert a non-Negative set occurring in PN, we first add all its arcs to SemDep, then we compact and update PN with respect the new value of SemDep. The Update procedure is defined below in (15).

(13) *Convert*(SemDep, PN, $\{n_{11}{\rightarrow}n_{12}, \ldots | p_{11}{\rightarrow}p_{12}, \ldots\}$)

1. set SemDep as *Add*(SemDep, $\{n_{11}{\rightarrow}n_{12}, \ldots | p_{11}{\rightarrow}p_{12}, \ldots\}$)
2. set PN as $PN \setminus \{n_{11}{\rightarrow}n_{12}, \ldots | p_{11}{\rightarrow}p_{12}, \ldots\}$
3. for each $pn \in PN$ do set PN as $((PN \setminus pn) \cup$ *Compact*(SemDep, pn))
4. set PN as *Update*(SemDep, PN)
5. return $\langle$SemDep, $PN\rangle$

The insertion of a new non-Positive set in PN simply compacts the former, adds it to the latter, and updates the latter.

(14) *Constrain*(SemDep, PN, pn)

1. set pn as *Compact*(SemDep, pn)
2. set PN as $PN \cup pn$
3. set PN as *Update*(SemDep, PN)
4. return PN

Finally, *Update* maintains PN consistent according to the criteria in (9)

(15) *Update*(SemDep, PN)

1. set SemDepTotal as SemDep
2. for each $pn{=}\{n_{11}{\rightarrow}n_{12}, \ldots | p_{11}{\rightarrow}p_{12}, \ldots\} \in PN$ do
3. if ($\{p_{11}{\rightarrow}p_{12}, \ldots\}$ is empty) then go to 2.
4. set SemDep′ as *Add*(SemDep, pn)
5. for each $d \in N_D$ do
6. if ($d{\rightarrow}d \in$ SemDep′) then
7. set PN as $PN \setminus pn$
8. go to 2.
9. for each $\{n'_{11}{\rightarrow}n'_{12}, \ldots | p'_{11}{\rightarrow}p'_{12}, \ldots\} \in \{PN \setminus pn\}$ do
10. if (($\{n'_{11}{\rightarrow}n'_{12}, \ldots\}$ is not empty) and
($\{n'_{11}{\rightarrow}n'_{12}, \ldots\} \subseteq$ SemDep′) and
($\{p'_{11}{\rightarrow}p'_{12}, \ldots\} \not\subset$ SemDep′)) then
11. set PN as $PN \setminus pn$
12. go to 2.
13. set SemDepTotal as *Add*(SemDepTotal, pn)
14. for each $\{n_{11}{\rightarrow}n_{12}, \ldots | p_{11}{\rightarrow}p_{12}, \ldots\} \in PN$ do
15. if ($\{n_{11}{\rightarrow}n_{12}, \ldots\}$ is empty) then go to 14.
16. if ($\{n_{11}{\rightarrow}n_{12}, \ldots\} \not\subseteq$ SemDepTotal) then
17. set PN as $PN \setminus \{n_{11}{\rightarrow}n_{12}, \ldots | p_{11}{\rightarrow}p_{12}, \ldots\}$
18. if ($\{p_{11}{\rightarrow}p_{12}, \ldots\}$ is not empty) then
19. set PN as $PN \cup \{|n_{11}{\rightarrow}n_{12}, \ldots, p_{11}{\rightarrow}p_{12}, \ldots\}$
20. return PN

Update executes two scans of PN. In the first (rows 2-13), it considers each non-Negative set pn; if pn's Positive part is empty, pn is ignored (row 3). Then

it builds a temporary variable SemDep′ by adding pn to SemDep. If SemDep′ contains a cycle (rows 5-8), or if there is another non-Negative set whose Negative part is included, and whose Positive part is not *properly* included, in SemDep′, cf. figg.4-5, (rows 9-12), then pn is removed from PN. The second scan (rows 14-19) ranges over each non-Positive set pn'. If pn''s Negative part cannot be covered by the sets left in PN, it is removed (rows 16-17). However, as explained in (9.c), if pn''s Positive part is not empty, we add the Positive set obtained by the union of all pn''s arcs (rows 18-19). pn''s coverage by PN is checked by testing if pn' is not included in SemDepTotal, a temporary variable that is incrementally built during the first scan by adding to SemDep all valid sets in PN (row 1, row 13).

4 Flexible Incremental Disambiguation in DTS

This section shows how NL constraints may be incrementally added by calling *Constrain*. Besides the constraints on logical Redundancy discussed above in (7), [17] implement in their experiment other two NL constraints: Nested Quantifiers and Island constraints. The former prevent certain dependencies on sentences involving quantifiers occurring in the restriction of other quantifiers. An example is (16), taken from [11], where $Every_z$ occurs in the restriction of $Some_y$.

(16) [Two$_x$ politicians] spy on [someone$_y$ from [every$_z$ city]]

It is awkward to interpret (16) via, for instance, SemDep=$\{x \rightarrow$Ctx, $y \rightarrow$Ctx, $z \rightarrow$Ctx, $x \rightarrow z\}$, i.e. a reading where there is a single person coming from every city which is spied by two different politicians for each city he comes from.

A popular solution proposed for Nested Quantifiers, e.g. [13], [21], and [9], states that no other quantifier can 'intercalate' between two nested quantifiers. With respect to (16), the constraints forbid both ($\forall 2 \exists$) and ($\exists 2 \forall$). Conversely, [16] states constraints on Nested Quantifiers in terms of semantic dependencies, and claimed that a discourse referent d_1, occurring in the restriction $R(d)$ of another discourse referent d, may depends on a third discourse referent d_2 outside $R(d)$ (and vice-versa), just in case d does. This is exemplified in fig.6.a-b.

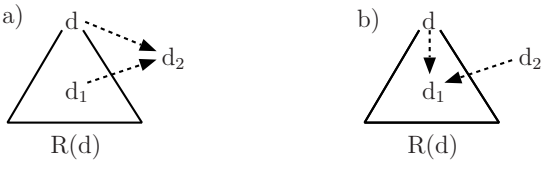


Fig. 6. DTS constraints for nested quantification: acceptable configurations

In order to account for Nested Quantifiers in DTS, it is clear that we must disallow any kind of dependency $d_1 \rightarrow d_2$ (or $d_2 \rightarrow d_1$) unless $d \rightarrow d_2$ (or $d_2 \rightarrow d$) also occurs in SemDep. With respect to (16), and its initial PN that contains all

possible dependencies[3], we then invoke *Constrain*(SemDep, PN, $\{z{\rightarrow}x\,|\,y{\rightarrow}x\}$) and *Constrain*(SemDep, PN, $\{x{\rightarrow}z\,|\,y{\rightarrow}z\}$). During the *Update*, the Positive sets $\{\,|\,x{\rightarrow}z\}$ and $\{\,|\,z{\rightarrow}y\}$ are removed on rows 9-12. Then the two Positive|Negative sets as parameter become useless and are removed from PN. However, since their Positive part is not empty, the Positives sets corresponding to the union of all their arcs are added in PN (rows 14-19). The latter turns out to be:

(17) $PN = \{\{\,|\,A{\rightarrow}\mathsf{Ctx}\},\ \{\,|\,x{\rightarrow}y\},\ \{\,|\,y{\rightarrow}x\},\ \{\,|\,y{\rightarrow}z\},\ \{\,|\,z{\rightarrow}y\},\ \{\,|\,z{\rightarrow}x,\ y{\rightarrow}x\},\ \{\,|\,x{\rightarrow}z,\ y{\rightarrow}z\}\}$

Moreover, since y and z are respectively associated with an existential and a universal quantifier, we apply constraints on logical redundancy, i.e. we prevent any arc entering y and any arc exiting z. Accordingly, we call *Constrain* on the Negative sets $\{x{\rightarrow}y\,|\,\}$, $\{z{\rightarrow}y\,|\,\}$, and $\{z{\rightarrow}x\,|\,\}$. PN becomes:

(18) $PN = \{\{\,|\,A{\rightarrow}\mathsf{Ctx}\},\ \{\,|\,y{\rightarrow}x\},\ \{\,|\,y{\rightarrow}z\},\ \{\,|\,x{\rightarrow}z,\ y{\rightarrow}z\}\}$

The sets in (18) generates the readings in fig.7, in line with [16]'s predictions.

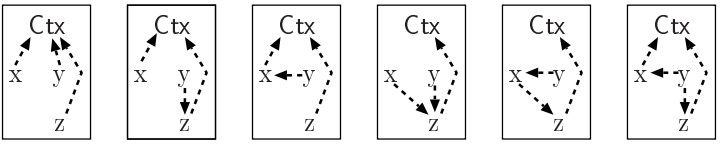


Fig. 7. The six disambiguated readings in DTS for sentence (16)

Finally, two kinds of Islands Constraints have been traditionally identified in the literature: finite clauses and coordinated structures [18]. An example of the former is shown in (19.a): $\forall$ cannot outscope $\exists$ in that *a student* is outside the finite clause where *every* occurs. In contrast, the scope of NPs can freely rise over non-finite clauses, as shown in (19.b), where $\forall\exists$ is available. An example of coordinated structures acting as islands is shown in (19.c). (19.c) yields two scopings only, $(\forall_x\exists_y\exists_z)$ and $(\exists_y\exists_z\forall_x)$, corresponding respectively to a reading where every student reads a different book and a different paper, and a reading where a single book and a single paper have been read by every student. In contrast, readings $(\exists_y\forall_x\exists_z)$ and $(\exists_z\forall_x\exists_y)$, where only one of the existentials depends on the universal, are impossible.

(19)
a. A_y student said you met every$_x$ professor.
b. A_y student wants you to meet every$_x$ professor.
c. Every$_x$ student reads a_y book and a_z paper.

[8] and [6], among others, proposed solutions to handle Island constraints of the first kind in Underspecification. Island constraints arising from coordinate structures have received less attention, an exception being [20].

[3] $PN = \{\{\,|\,A{\rightarrow}\mathsf{Ctx}\},\ \{\,|\,x{\rightarrow}y\},\ \{\,|\,y{\rightarrow}x\},\ \{\,|\,x{\rightarrow}z\},\ \{\,|\,z{\rightarrow}x\},\ \{\,|\,y{\rightarrow}z\},\ \{\,|\,z{\rightarrow}y\}\}$.

In DTS, in order to account for Island constraints of the first kind, we forbit any dependency $d_1 \rightarrow d_2$ where d_1 is a discourse referent outside an island, and d_2 a discourse referent occurring therein. With respect to (19a), we then invoke *Constrain*(SemDep, PN, $\{y \rightarrow x \mid\}$). On the other hand, in (19c) we do not leave y to depend on x unless z also does (and vice-versa). This is handled by calling *Constrain* on $\{y \rightarrow x \mid z \rightarrow x\}$ and $\{z \rightarrow x \mid y \rightarrow x\}$. Note that the first call removes $y \rightarrow x$ from the initial PN, then it finds out that $\{y \rightarrow x \mid z \rightarrow x\}$ is useless (*Update*, row 16) and substitutes it with the Positive set $\{\mid y \rightarrow x,\ z \rightarrow x\}$ (*Update*, row 19). The second call removes both $z \rightarrow x$ and the Positive set just inserted (*Update*, row 10), and re-insert it as before. After the insertion of the constraints preventing the Redundancy triggered by the universal/existential quantifiers, we get:

(20) $PN = \{\{\mid A \rightarrow \mathsf{Ctx}\},\ \{\mid y \rightarrow x,\ z \rightarrow x\}\}$

It is easy to see that (20) enables only the two readings corresponding to $(\forall_x \exists_y \exists_z)$ and $(\exists_y \exists_z \forall_x)$. Note that (20) corresponds to the disjunction of those readings.

5 Conclusion and Future Work

In this paper, we presented procedures to carry out incremental disambiguation in DTS. The present paper evolves the research done in [17], where we proposed an Expressively Complete version of DTS, but did not show how disambiguation might be computationally achieved, i.e. how the several readings might be obtained/blocked starting from the fully-underspecified one.

We claim that the disambiguation process proposed here is *flexible*, in the sense that it is able to account for any kind of NL constraints on available readings. In this paper, we show the behaviour of the procedure *Constrain* wrt the three NL constraints on dependencies among quantifiers studied by [17], but it is clear that it may be used as well for other constraints, even extra-linguistic ones. For instance, as observed in [7], §1.2, (21) cannot be interpreted via a DTS structure where x and y are linked to Ctx only. That would describe a reading in which a child has several fathers, which makes no sense in real contexts.

(21) I've met a_x child of every$_y$ man in this room.

Obviously, in the framework presented here, the unavailable reading may be blocked by simply calling *Constrain*(SemDep, PN, $\{A \rightarrow \mathsf{Ctx} \mid x \rightarrow y\}$).

Most current underspecified formalisms implement constraints on Nested Quantifiers and Island constraints as static compositional rules in the syntax-semantic interface. However, it is not clear how to extend such an interface in order to encompass other constraints, as those on logical redundancy or those arising from world-knowledge as the one in (21). Conversely, in the present framework, since the procedure *Constrain* may be used to deal with all of them, each class of constraints may be studied in isolation in order to identify the parameters that must be given to the procedures in order to block the corresponding unavailable readings. Obviously a complete study and implementation of the several sources of constraints deserves much further work.

References

1. Beghelli, F., Ben-Shalom, D., Szabolski, A.: Variation, Distributivity, and the Illusion of Branching. In: Szabolcsi, A. (ed.) Ways of Scope Taking, pp. 29–69. Kluwer, Dordrecht (2001)
2. Bos, J.: Computational Semantics in Discourse: Underspecification, Resolution, and Inference. Journal of Logic, Language and Information 13, 139–157 (2004)
3. Chaves, R.P.: Non-Redundant Scope Disambiguation in Underspecified Semantics. In: Proc. of the 8th ESSLLI Student Session, Vienna, pp. 47–58 (2003)
4. Copestake, A., Flickinger, D., Sag, I.A.: Minimal Recursion Semantics. An introduction. Research on Language and Computation 3(2) (2005)
5. Ebert, C.: Formal Investigations of Underspecified Representations. PhD thesis, Department of Computer Science, King's College London (2005)
6. Egg, M., Koller, A., Niehren, J.: The Constraint Language for Lambda Structures. Journal of Logic, Language and Information 10, 457–485 (2001)
7. Hobbs, J.R., Shieber, S.: An Algorithm for Generating Quantifier Scoping. Computational Linguistics 13, 47–63 (1987)
8. Joshi, A.K., Kallmeyer, L.: Factoring Predicate Argument and Scope Semantics. Research on Language and Computation 1, 3–58 (2003)
9. Joshi, A.K., Kallmeyer, L., Romero, M.: Flexible Composition in LTAG: Quantifier Scope and Inverse Linking. In: Musken, R., Bunt, H. (eds.) Computing Meaning, vol. 3. Kluwer, Dordrecht (2003)
10. Koller, A., Thater, S.: Towards a redundancy elimination algorithm for underspecified descriptions. In: Proc. of the 5th Int. Workshop on Inference in Computational Semantics (ICoS-5), Buxton, England (2006)
11. Larson, R.K.: Quantifying into NP. University of Wisconsin (1985) (manuscript)
12. May, R., Bale, A.: Inverse Linking. In: Everaert, M., van Riemsdijk, H. (eds.) The Blackwell Companion to Syntax, vol. II. Malden (2005)
13. Park, J.: Quantifier Scope and Constituency. In: Proceedings of the 33rd Annual Meeting of the ACL, pp. 205–212 (1996)
14. Reyle, U.: Dealing with ambiguities by Underspecification: Construction, Representation and Deduction. Journal of Semantics 13, 123–179 (1993)
15. Reyle, U.: Co-Indexing Labelled DRSs to Represent and Reason with Ambiguities. In: Peters, S., van Deemter, K. (eds.) Semantic Ambiguity and Underspecification, Stanford, pp. 239–268 (1996)
16. Robaldo, L.: Dependency Tree Semantics. PhD thesis, Department of Computer Science, Turin University, Italy (2007)
17. Robaldo, L., Di Carlo, J.: Disambiguating quantifier scope in DTS. In: Proc. of 8th International Workshop on Computational Semantics (IWCS-8), Tilburg, The Netherlands 2009 (2006)
18. Ross, J.R.: Constraints on Variables in Syntax. Ph.D thesis, Massachusetts Institute of Technology (1967)
19. Vestre, E.J.: An algorithm for generating non-redundant quantifier scopings. In: Proc. of the 5th conference on European chapter of the Association for Computational Linguistics, Berlin, Germany, pp. 251–256 (1991)
20. Willis, A.: NP Coordination in Underspecified Scope Representations. In: Proc. of the 7th Workshop on Computational Semantics (IWCS-7), Tilburg, pp. 235–246 (2007)
21. Willis, A.: An Efficient Treatment of Quantification in Underspecified Semantic Representations. Ph.D thesis, University of York (2000)

A Syntactic Textual Entailment System Based on Dependency Parser

Partha Pakray[1], Alexander Gelbukh[2], and Sivaji Bandyopadhyay[1]

[1] Computer Science and Engineering Department,
Jadavpur University, Kolkata, India
parthapakray@gmail.com, sbandyopadhyay@cse.jdvu.ac.in
[2] Center for Computing Research, National Polytechnic Institute,
Mexico City, Mexico
gelbukh@gelbukh.com

Abstract. The development of a syntactic textual entailment system that compares the dependency relations in both the text and the hypothesis has been reported. The Stanford Dependency Parser has been run on the 2-way RTE-3 development set and the dependency relations obtained for a text and hypothesis pair has been compared. Some of the important comparisons are: subject-subject comparison, subject-verb comparison, object-verb comparison and cross subject-verb comparison. Corresponding verbs are further compared using the WordNet. Each of the matches is assigned some weight learnt from the development corpus. A threshold has been set on the fraction of matching hypothesis relations based on the development set. The threshold score has been applied on the RTE-4 gold standard test set using the same methods of dependency parsing followed by comparisons. Evaluation scores obtained on the test set show 54.75% precision and 53% recall for YES decisions and 54.45% precision and 56.2% recall for NO decisions.

Keywords: Textual Entailment, Dependency parsing, Dependency Relations, RTE-3 development set, RTE-4 gold standard test set.

1 Introduction

Recognizing Textual Entailments (RTE) is one of the recent challenges of Natural Language Processing (NLP). Textual Entailment is defined as a directional relationship between pairs of text expressions, denoted by T – the entailing "Text", and H- the entailed "Hypothesis". T entails H if the meaning of H can be inferred from the meaning of T, as would typically be interpreted by people. For instance, the following is a correct entailment pair:

T: US Secretary of State Condoleezza Rice has been defending President Bush's Iraq strategy at a Senate hearing.
H: Rice defends Bush.

There were three Recognizing Textual Entailment competitions RTE-1 in 2005, RTE-2 in 2006 and RTE-3 in 2007 which were organized by PASCAL (Pattern Analysis, Statistical Modeling and Computational Learning) - the European Commission's

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 269–278, 2010.

IST-funded Network of Excellence for Multimodal Interfaces. In 2008, the fourth edition (RTE-4) of the challenge was organized by NIST (National Institute of Standards and Technology) in Text Analysis Conference (TAC). In every new competition several new features of RTE were introduced. The RTE-5 challenge in 2009 includes a separate search pilot along with the main task.

The first PASCAL Recognizing Textual Entailment Challenge (RTE-1) [1], introduced the first benchmark for the entailment recognition task. The RTE-1 dataset consists of manually collected text fragment pairs, termed text (t) (1-2 sentences) and hypothesis (h) (one sentence). The systems were required to judge for each pair whether t entails h. The pairs represented success and failure settings of inferences in various application types (termed "tasks").

In RTE-1 the various techniques used by the participating systems were word overlap, WordNet, statistical lexical relation, world knowledge, syntactic matching and logical inference.

After the success of RTE-1, the main goal of the RTE-2, held in 2006 [2], was to support the continuity of research on textual entailment. The RTE-2 data set was created with the main focus of providing more "realistic" text-hypothesis pair. As in the RTE-1, the main task was to judge whether a hypothesis H is entailed by a text T. The texts in the datasets were of 1-2 sentences, while the hypotheses were one sentence long. Again, the examples were drawn to represent different levels of entailment reasoning, such as lexical, syntactic, morphological and logical.

The main task in the RTE-2 challenge was classification – entailment judgment for each pair in the test set that represented either entailment or no entailment. The evaluation criterion for this task was accuracy – the percentage of pairs correctly judged. A secondary task was created to rank the pairs based on their entailment confidence. A perfect ranking would place all the positive pairs (for which the entailment holds) before all the negative pairs. This task was evaluated using the average precision measure [3], which is a common evaluation measure for ranking in information retrieval.

In RTE-2 the techniques used by the various participating systems are Lexical Relation/ database, n-gram/ subsequence overlap, syntactic matching/ Alignment, Semantic Role labelling/ Framenet/ PropBank, Logical Inference, Corpus/web-based statistics, machine learning (ML) Classification, Paraphase and Templates, Background Knowledge and acquisition of entailment corpus.

The RTE-3 data set consisted of 1600 text-hypothesis pairs, equally divided into a development set and a test set. The same four applications from RTE-2 – namely IE, IR, QA and SUM – were considered as settings or contexts for the pair's generation. 200 pairs were selected for each application in each data set. Each pair was annotated with its related task (IE/IR/QA/SUM) and entailment judgment (YES/NO).

In addition, an optional pilot task, called "Extending the Evaluation of Inferences from Texts" was set up by the NIST, in order to explore two other sub-tasks closely related to textual entailment: differentiating unknown entailment from identified contradictions and providing justifications for system decisions. In the first sub-task, the idea was to drive systems to make more precise informational distinctions, taking a three-way decision between "YES", "NO" and "UNKNOWN", so that a hypothesis being unknown on the basis of a text would be distinguished from a hypothesis being shown false/contradicted by a text.

In RTE-4, no development set was provided, as the pairs proposed were very similar to the ones contained in RTE-3 development and test sets, which could therefore be used to train the systems. Four applications – namely IE, IR, QA and SUM – were considered as settings or contexts for the pair generation. The length of the H's was the same as in the past data sets (RTE-3); however, the T's were generally longer. A major difference with respect to RTE-3 was that the RTE-4 data set consisted of 1000 T-H pairs, instead of 800.

In RTE-4, the challenges were classified as two-way task and three-way task. The two-way RTE task was to decide whether:

- T entails H - in which case the pair will be marked as ENTAILMENT;
- T does not entail H - in which case the pair will be marked as NO ENTAILMENT.

The three-way RTE task was to decide whether:

- T entails H - in which case the pair was marked as ENTAILMENT
- T contradicts H - in which case the pair was marked as CONTRADICTION
- The truth of H could not be determined on the basis of T - in which case the pair was marked as UNKNOWN

In RTE-4 competition [4], 45 runs were submitted by 26 participants, half of whom chose the 3-way task. In the 3-way task, the best accuracy was 0.685. The 3-way task appeared to be altogether quite challenging, as the average 3-way score was 0.51, quite low compared to the results achieved in previous campaigns. The systems performed better in the 2-way task, achieving accuracy scores which ranged between 0.459 and 0.746. These results are lower than those achieved in RTE-3 challenge, where the accuracy scores ranged from 0.49 to 0.80, even though a comparison is not really possible as the data sets were actually different.

In the present paper, a 2-way syntactic textual entailment recognition system has been described that has been trained on the 2-way RTE-3 development set and then tested on the RTE-4 test set. Related works are described in Section 2. Section 3 describes syntactic based RTE system architecture. The experiment carried out on the development and test data sets are described in Section 4 along with the results. The conclusions are drawn in Section 5.

2 Related Works

In the various RTE Challenge, several methods are applied on the textual entailment task. Most of these systems use some sort of lexical matching (e.g. n-gram, word similarity), be it simple word overlap. A number of systems represent the texts as parse trees (e.g. syntactic, dependency) before the actual task. Some of the systems use semantic relation (e.g. logical inference, Semantic Role Labeling) for solving the text and hypothesis entailment problem.

The work presented in [5] suggests that sentence structure plays an important role in recognizing textual entailment and paraphrasing accurately. The Recognizing Textual Entailment System in [6] was based on the use of a broad-coverage parser to extract dependency relations and a module which obtains lexical entailment relations

from WordNet. The use of syntactic tree editing distance to detect entailment relations is proposed in [7]. They calculate the similarity between the two dependency trees of T and H directly. Lexical relation, WordNet and Syntactic Matching for solving the textual entailment problem are used in [8].

The system presented in [9] proposed a novel approach to RTE that exploits a structure-oriented sentence representation followed by a similarity function. The structural features are automatically acquired from tree skeletons that are extracted and generalized from dependency trees.

A syntactic dependency tree approach for the task of textual entailment is used in [10]. This system approach is to construct the syntactic dependency trees for both text and hypothesis sentences and then compare the nodes of the dependency trees by using the semantic similarity between the two nodes. Their approach is closest to method used in the present work. But, a different scoring mechanism and a different set of syntactic relations have been used in the present work. The scoring technique is quite simple and thus easy to compute and interpret.

3 System Description

In this section, we describe our syntactic textual entailment system. The system extracts syntactic structures from the text-hypothesis pairs using Stanford Parser and compares the corresponding structures to determine if the entailment relation is established. The system accepts pairs of text snippets (text and hypothesis) at the input and gives a value at the output: YES if the text entails the hypothesis and NO otherwise. The architecture of the proposed system is described in Figure 1.

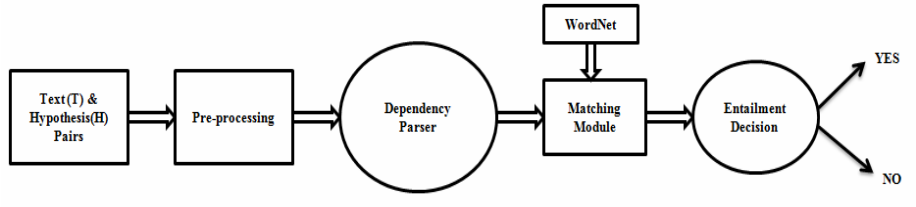


Fig. 1. Syntactic Textual Entailment Recognition System

The various components of the textual entailment recognition system are Pre-processing module, Dependency Parser module, Matching module and Entailment Decision module. Each of these modules is now being described in subsequent subsections.

3.1 Pre-processing Module

The system accepts pairs of text snippets (text and hypothesis) at the input and gives the output: YES if the text entails the hypothesis and NO otherwise. An example text-hypothesis pair from the RTE-3 development set is shown in Figure 2.

<pair id="1" **entailment**="YES" **task**="IE" **length**="short" >
<t>The sale was made to pay Yukos' US$ 27.5 billion tax bill, Yuganskneftegaz was originally sold for US$ 9.4 billion to a little known company Baikalfinansgroup which was later bought by the Russian state-owned oil company Rosneft .**</t>**
<h>Baikalfinansgroup was sold to Rosneft.**</h>**
</pair>

Fig. 2. RTE-3 development set text-hypothesis pair

We replace in all development data the expressions “aren’t” with “are not”, “didn’t” with “did not”, “doesn’t” with “does not”, “won’t” with “will not”, “don’t” with “do not”, “hasn’t” with “has not”, “isn’t” with “is not”, “couldn’t” with “could not”, “ă” with “a”, "á" with “a”, "š" with “s”, "ž" with “z”, "ó" with “o”. These expressions are either abbreviations or include special characters for which the dependency parser gives erroneous results. It has also been observed that escape characters like ", …, ‘ and & are present in the text and in the hypothesis parts and these were removed. All the above pre-processing methods were applied on the development set and the test set.

3.2 Dependency Parser Module

This module is based on the Stanford Parser [11], which normalizes data from the corpus of text and hypothesis pairs, accomplishes the dependency analysis and creates appropriate structures Our Entailment system uses the following features,

a. Subject: The dependency parser generates *nsubj* (nominal subject) and *nsubjpass* (passive nominal subject) tags for the subject feature. Our entailment system uses these tags.
b. Object: The dependency parser generates *dobj* (direct object) as object tags.
c. Verb: Verbs are wrapped with either the subject or the object.
d. Noun: The dependency parser generates *nn* (noun compound modifier) as noun tags.
d. Preposition: Different type of prepositional tags are *prep_in*, *prep_to*, *prep_with* etc. For example, in the sentence “A plane crashes in Italy.”, the prepositional tag identified is *prep_in*(in, Italy).
e. Determiner: Determiner denotes a relation with a noun phase. The dependency parser generates *det* as determiner tags. For example, the parsing of the sentence “A journalist reports on his own murders.” generates the determiner relation as *det*(journalist,A).
f. Number: The numeric modifier of a noun phrase is any number phrase. The dependency parser generates *num* (numeric modifier). For example, the parsing of the sentence “Nigeria seizes 80 tonnes of drugs.” generates the relation *num* (tonnes, 80).

Here is an example from RTE-4 data set. For the sentence, “Nigeria seizes 80 tonnes of drugs”, the Stanford Dependency Parser generates the following set of dependency relations:

[
nsubj(seizes-2, Nigeria-1),
num(tonnes-4, 80-3),
dobj(seizes-2, tonnes-4),
prep_of(tonnes-4, drugs-6)
]

3.3 Matching Module

After dependency relations are identified for both the text and the hypothesis in each pair, the hypothesis relations are compared with the text relations. The different features that are compared are noted below. In all the comparisons, a matching score of 1 is considered when the complete dependency relation along with all of its arguments matches in both the text and the hypothesis. In case of a partial match for a dependency relation, a matching score of 0.5 is assumed.

a. Subject-Verb Comparison: The system compares hypothesis subject and verb with text subject and verb that are identified through the *nsubj* and *nsubjpass* dependency relations. A matching score of 1 is assigned in case of a complete match. Otherwise, the system considers the following matching process.
b. WordNet Based Subject-Verb Comparison: If the corresponding hypothesis and text subjects do match in the subject-verb comparison, but the verbs do not match, then the WordNet distance between the hypothesis and the text is compared. If the value of the WordNet distance is less than 0.5, indicating a closeness of the corresponding verbs, then a match is considered and a matching score of 0.5 is assigned. Otherwise, the subject-subject comparison process is applied.
c. Subject-Subject Comparison: The system compares hypothesis subject with text subject. If a match is found, a score of 0.5 is assigned to the match.
d. Object-Verb Comparison: The system compares hypothesis object and verb with text object and verb that are identified through *dobj* dependency relation. In case of a match, a matching score of 0.5 is assigned.
e. WordNet Based Object-Verb Comparison: The system compares hypothesis object with text object. If a match is found then the verb corresponding to the hypothesis object with text object's verb is compared. If the two verbs do not match then the WordNet distance between the two verbs is calculated. If the value of WordNet distance is below 0.50 then a matching score of 0.5 is assigned.
f. Cross Subject-Object Comparison: The system compares hypothesis subject and verb with text object and verb or hypothesis object and verb with text subject and verb. In case of a match, a matching score of 0.5 is assigned.
g. Number Comparison: The system compares numbers along with units in the hypothesis with similar numbers along with units in the text. Units are first compared and if they match then the corresponding numbers are compared. In case of a match, a matching score of 1 is assigned.
h. Noun Comparison: The system compares hypothesis noun words with text noun words that are identified through *nn* dependency relation. In case of a match, a matching score of 1 is assigned.

i. Prepositional Phrase Comparison: The system compares the prepositional dependency relations in the hypothesis with the corresponding relations in the text and then checks for the noun words that are arguments of the relation. In case of a match, a matching score of 1 is assigned.
j. Determiner Comparison: The system compares the determiner in the hypothesis and in the text that are identified through *det* relation. In case of a match, a matching score of 1 is assigned.
k. Other relation Comparison: Besides the above relations that are compared, all other remaining relations are compared verbatim in the hypothesis and in the text. In case of a match, a matching score of 1 is assigned.

WordNet [12] is one of most important resource. The WordNet 2.0 has been used for WordNet based subject-verb comparison and WordNet based Object-verb comparison. API for WordNet Searching RiWordnet [13] provides Java applications with the ability to retrieve data from the WordNet database.

3.4 Entailment Decision

Each of the matches through the above comparisons is assigned some weight learnt from the development corpus. A threshold of 0.30 has been set on the fraction of matching hypothesis relations based on the development set results that gives optimal precision and recall values for both YES and NO entailment. The threshold score has been applied on the RTE-4 gold standard test set using the same methods of dependency parsing followed by comparisons.

4 Experiments on the Development and the Test Data and the Results

In RTE-4 there was no development set provided, as the pairs proposed were very similar to the ones contained in RTE-3 development and test sets, which could therefore be used to train the systems. Four applications – namely IE, IR, QA and SUM – were considered as settings or contexts for the pair generation. The length of the H's was the same as in the past data sets (RTE-3); however, the T's were generally longer. The RTE-3 development set was used to train our entailment system to identify the threshold values for the various measures towards entailment decision. The 2-way RTE-3 development set consisted of 800 text-hypothesis pairs. The RTE-4 test set consisted of 1000 text-hypothesis pair.

In our textual entailment system, the method was run separately on the RTE-3 development set and two-way entailment (YES or NO) decisions were obtained for each text-hypothesis pair. Experiments were carried out to measure the performance of the final RTE system. It is observed that the precision and recall measures of the final RTE system are best when final entailment decision is based on positive results with threshold value 0.30. The results on the RTE-3 development data set for each task (IE/IR/QA/SUM) are shown in Table 1. It is observed that the system performs best on the development set for the QA task and worst on the development set for the IE task. This points to the requirement of system tuning with respect to the associated

task but this point has not been studied further. Two baseline systems have been developed in the present task. The Baseline–1 system assigns YES tag to all the text-hypothesis pairs and the Baseline–2 system assigns NO tag to all the text0hypothesis pairs. The results obtained on Baseline–1 and Baseline–2 systems on the RTE-3 development data set and the RTE-4 test data set have been shown in Table 2 and Table 3 respectively. The results on the RTE-3 development set for YES and NO entailment decisions are shown in Table 4. The results on RTE-4 test set are shown in Table 5. The system performance on the RTE-3 development set and RTE-4 test set are clearly above the baseline.

Table 1. RTE 3 development set task when threshold value 0.30

RTE 3 Development Set		IE		IR		QA		SUM	
		YES	NO	YES	NO	YES	NO	YES	NO
Cut Off 0.30	Precision	0.55	0.47	0.66	0.65	0.76	0.65	0.68	0.58
	Recall	0.65	0.38	0.48	0.80	0.64	0.77	0.57	0.69

Table 2. Baseline-1 system for RTE-3 Development Set and RTE-4 Test Set

	Entailment Decision	No. of Entailment in Gold standard	Baseline-1	Precision
RTE-3 Development Set	YES	412	800	51.50%
	NO	388	0	0%
RTE-4 Test Set	YES	500	1000	50.00%
	NO	500	0	0%

Table 3. Baseline-2 system for RTE-3 Development Set and RTE-4 Test Set

	Entailment Decision	No. of Entailment in Gold standard	Baseline-2	Precision
RTE-3 Development Set	YES	412	0	0%
	NO	388	800	48.50%
RTE-4 Test Set	YES	500	0	0%
	NO	500	1000	50.00%

Table 4. RTE 3 development set when threshold value 0.30

Entailment Decision	No. of Entailment in Gold standard	No. of correct Entailment in our system	Total No. of Entailment given by our system	Precision	Recall
YES	412	244	371	65.76%	59.22%
NO	388	261	429	60.83%	67.26%
Overall	800	505	800	63.12%	63.12%

Table 5. RTE 4 test set when threshold value 0.30

Entailment Decision	No. of Entailment in Gold standard	No. of correct Entailment in our system	Total No. of Entailment given by our system	Precision	Recall
YES	500	265	484	54.75%	53.00%
NO	500	281	516	54.45%	56.20%
Overall	1000	546	1000	54.60%	54.60%

5 Conclusions

Results show that a syntactic-based approach is not enough to tackle appropriately the textual entailment problem. Experiments have been started for a semantic based RTE task. In the present task, the final RTE system has been optimized for the entailment YES/NO decision using the development set. The role of the application setting for the RTE task has also not been looked into. This needs to be experimented in future. Finally, the two way task has to be upgraded to the three way task.

References

[1] Dagan, I., Glickman, O., Magnini, B.: The PASCAL Recognising Textual Entailment Challenge. In: Proceedings of the First PASCAL Recognizing Textual Entailment Workshop (2005)

[2] Bar-Haim, R., Dagan, I., Dolan, B., Ferro, L., Giampiccolo, D., Magnini, B., Szpektor, I.: The Second PASCAL Recognising Textual Entailment Challenge. In: Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment, Venice, Italy (2006)

[3] Voorhees, E.M., Harman, D.: Overview of the seventh text retrieval conference. In: Proceedings of the Seventh Text REtrieval Conference (TREC-7). NIST Special Publication (1999)

[4] Giampiccolo, D., Dang, H.T., Magnini, B., Dagan, I., Cabrio, E.: The Fourth PASCAL Recognizing Textual Entailment Challenge. In: TAC 2008 Proceedings (2008), http://www.nist.gov/tac/publications/2008/papers.html
[5] Vanderwende, L., Coughlin, D., Dolan, B.: What syntax can contribute in entailment task. In: Proceedings of the First PASCAL Recognizing Textual Entailment Workshop, pp. 13–16 (2005)
[6] Herrera, J., Peñas, A., Verdejo, F.: Textual Entailment Recognition Based on Dependency Analysis and WordNet. In: Proceedings of the First PASCAL Recognizing Textual Entailment Workshop, pp. 21–24 (2005)
[7] Kouylekov, M., Magnini, B.: Tree Edit Distance for Recognizing Textual Entailment: Estimating the Cost of Insertion. In: Proc. of the PASCAL RTE-2 Challenge, pp. 68–73 (2006)
[8] Blake, C.: The Role of Sentence Structure in Recognizing Textual Entailment. In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, pp. 101–106 (2007)
[9] Wang, R., Neumann, G.: Recognizing Textual Entailment Using Sentence Similarity based on Dependency Tree Skeletons. In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, pp. 36–41 (2007)
[10] Varma, V., Krishna, S., Garapati, H., Reddy, K., Pingali, P., Ganesh, S., Gopisetty, H., Bysani, P., Katragadda, R., Sarvabhotla, K., Reddy, V.B., Bharadwaj, R.: Recognizing Textual Entailment (RTE) Track. In: Text analysis conference 2008 Proceedings (2008)
[11] Klein, D., Manning, C.D.: Accurate unlexicalized parsing. In: ACL 2003, pp. 423–430 (2003)
[12] Fellbaum, C.: WordNet: An Electronic Lexical Database. MIT Press, Cambridge (1998)
[13] RiWordnet API Tool, http://www.rednoise.org/rita/wordnet/documentation/index.htm

Semantic Annotation of Transcribed Audio Broadcast News Using Contextual Features in Graphical Discriminative Models

Azeddine Zidouni[1] and Hervé Glotin[2]

[1] Laboratoire LSIS UMR CNRS 6168, Université Aix-Marseille 2, France
azeddine.zidouni@lsis.fr
[2] Laboratoire LSIS UMR CNRS 6168, Université du sud Toulon-Var, France
glotin@univ-tln.fr

Abstract. In this paper we propose an efficient approach to perform named entities retrieval (NER) using their hierarchical structure in transcribed speech documents. The NER task consists of identifying and classifying every word in a document into some predefined categories such as person name, locations, organizations, and dates. Usually the classical NER systems use generative approaches to learn models considering only the words characteristics (word context). In this work we show that NER is also sensitive to syntactic and semantic contexts. For this reason, we introduce an extension of conditional random fields (CRFs) approach to consider multiple contexts. We present an adaptation of the text-approach to the automatic speech recognition (ASR) outputs. Experimental results show that the proposed approach outperformed a CRFs simple application. Our experiments are done using ESTER 2 campaign data. The proposed approach is ranked in 4th position in ESTER 2 participating sites, it achieves a significant relative improvement of 18% in slot rate error (SER) measure over HMMs method.

Keywords: Information extraction, Named entity retrieval, Hierarchical named entities, Automatic speech recognition.

1 Introduction

Named entity retrieval (NER) is an important first step for many information extraction systems in almost natural language processing (NLP) applications. The NER task consists of identifying and classifying every word in a document into some predefined categories such as person name, locations, organizations, and dates. For example, in the sentence "*Albert Einstein was born on March 14, 1879*" the compound word *Albert Einstein* is identified as PERSON and *March 14, 1879* as DATE. The goal is then to construct a system capable automatically annotate text documents as human annotation. Several approaches try to reduce the gap between manual and automatic annotation by reducing ambiguities, they arise to improve robustness and portability. Mostly machines learning (ML) approaches are used in NER task like Support Vector Machine

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 279–290, 2010.

(SVM) [2], graphical models [9]. ML approaches are used in NER task in several applications such as Information Extraction (IE), Question Answering (Q&R), Biomedical Applications (BA), Summarization and Information Retrieval (IR). This diversity implies the use of domain specific NEs and leads consequently to many different corresponding annotations. Specific domains applications usually use small sets of NE types ($\sim$ 6 NE types). However in more general area, like indexing newspapers, applications use larger sets of NE types. Thus, NEs are organized on types and sub-types, each type's domain can be considered as conceptualization level. There are several conceptualization levels and the number of NE types is exponential in comparison to the number of levels. In this paper, we consider that each NE is defined at several levels of conceptualization. Our work consists of exploiting this hierarchical structure of NEs using a combination of models to improve NER performances. The system makes use of the different contextual information of the words along with the NE hierarchical features that are helpful in the prediction process.

Information retrieval in speech documents uses automatic speech recognition (ASR) systems to give a textual transcription. The simplest way to construct NER system for ASR outputs is to adapt the existing NER systems for textual approaches. But spoken language is different from written text in the way these methods are produced. Thus, the context is quite different between the transcribed speech and the training data. Katsuhito et al. [12] construct a NER model which is trained using both text and ASR-based training data. In our work, for speech documents, we propose an adaptation of the text-based approach mentioned above.

The rest of the paper is structured as follows. Section 2 describes the NER task using graphical models and describes the framework of Conditional Random Fields (CRFs). In section 3, we define a context-based approach for NER task using the NEs hierarchical structure and POS tagging as an a priori information. We use a combination of linear-chain CRFs to improve the performances of classical approaches. In section 4, we present the adaptation of the text-based NER approach to ASR output transcription; we give experimental results in both textual and transcribed documents. We conclude in Section 5.

2 NER with Graphical Models

We use a graphical system that relates a word sequence $W = \langle w_1, w_2, \ldots, w_n \rangle$ to a sequence of information states $L = \langle l_1, l_2, \ldots, l_n \rangle$ that maximize the conditional probability $P(L|W)$:

$$\widehat{L} = arg\ max_{L \in \varphi(W)} p(L|W). \tag{1}$$

In the NER task, the states l_i correspond to different types of NEs (labels). $\varphi(W)$ is the set of possible labels sequences for the input sequence W. The power of graphical models lies in their ability to model many variables that are independent by a product of local functions that each depends on only a small number of variables. Generative models, such as hidden Markov models (HMMs)

[10], are widely used on several data annotation tasks. HMMs are used to represent the joint probability distribution $p(w, l)$, where the variable l represents the label that we wish to predict, and the input variables w represent the observed knowledge about the entities (Figure 1). Modeling the joint distribution requires modeling the distribution $p(w)$, however using non-independent features with complex dependencies of the inputs makes the modeling intractable (they have been usually ignored). In contrast, discriminative models, such as conditional random fields (CRFs) [7], can deal with diverse features. The greatest advantage of CRFs is their flexibility to include a variety of features. In what follows, we will present the CRFs approach, which seems to be the current state-of-the-art approaches in information extraction and sequence labeling.

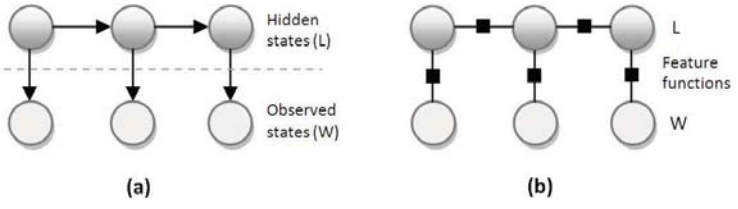


Fig. 1. (a) HMMs structure, as a generative model, and (b) CRFs structure as a descriminative model

2.1 Conditional Random Fields

CRFs are discriminative undirected graphical models. CRFs are conditionally trained to discriminate the correct sequence from all other candidate sequences without making independence assumption for features. The CRFs approach allows the use of arbitrary, dependent features and joint inference over the entire sequence (it incorporates many features of the words in a single unified model, Figure 1). CRFs was successfully used in numerous tasks of the natural language processing as the part-of-speech tagging [7], shallow parsing, named entity recognition [9], and Chinese word segmentation. In what follows, we will define the linear-chain CRFs theoretical framework that allows to resolve the problem formalized in the equation 1.

Definition 1 (Linear-chain CRFs). *Let $W = \langle w_1, w_2, \ldots, w_n \rangle$ be an sequence of observed words of length n (input variables). Let φ be a set of states that we wish to predict (output variables). Each variable $l \in \varphi$ takes outcomes from a set φ, which is discrete. Let $L = \langle l_1, l_2, \ldots, l_n \rangle$ be the sequence of states in φ that correspond to the labels assigned to words in the input sequence W. Linear-chain CRFs define the conditional probability of a state sequence L given an input sequence W to be:*

$$P(L|W) = \frac{1}{Z(W)} \prod_{i=1}^{n} \Psi_K(l_{i-1}, l_i, W, i), \tag{2}$$

where the normalization factor $Z(W)$ is the sum of the scores of all possible state sequences. In arbitrarily-structured CRFs, calculating the normalization factor in closed form is intractable, but in linear-chain-structured CRFs, the probability that a particular transition was taken between two CRFs states at a particular position in the input sequence can be calculated efficiently by dynamic programming. The Viterbi algorithm for finding the most likely state sequence given the observation sequence can be correspondingly modified from its HMM form. The local function $\Psi_K : \varphi^n \rightarrow \Re^+$ is the sum of K feature functions. Ψ_K can be represented in exponential form:

$$\Psi_K(l_{i-1}, l_i, W, i) = \exp\{\sum_{k=1}^{K} \lambda_k f_k(l_{i-1}, l_i, W, i)\}, \tag{3}$$

where $F_K = \{f_k(l_{i-1}, l_i, W, i)\}_{k=1}^K$ is a set of arbitrary binary functions that describes features (These functions are also called *feature functions*). For example, in the NER task, a boolean feature function $f_i(l_{i-1}, l_i, W, i)$ might be true if the word w_i is upper case and the label l_i is the named entity *LOCATION*. Here we write the feature function as potentially depending on the entire input sequence W. $\Lambda = \{\lambda_k\} \in \Re^k$ is a learned parameters vector (one learned parameter for each feature function). The number of state sequences is exponential in the input sequence length n. The CRFs parameters estimation problem is to determine the parameters $\Lambda = \{\lambda_k\} \in \Re^k$ from the training data $\mathcal{D} = \{\langle w_i, l_i \rangle\}_{i=1}^N$ with empirical distribution $\widetilde{p}(w, l)$. These weights are set to maximize the conditional log likelihood objective function $\Im(\Lambda)$ in the training data:

$$\Im(\Lambda) = \sum_{i=1}^{N} logP_\Lambda(w_i|l_i) - \sum_k \frac{\lambda_k^2}{2\sigma^2}, \tag{4}$$

where the second sum is a Gaussian prior over parameters (with variance σ) that provides smoothing to help cope with sparsity in the training data. To find these feature weights efficiently, we can use a quasi-Newton method called L-BFGS. This method approximates the second-derivative of the likelihood by keeping a running, finite-sized window of previous first-derivatives.

In the NER task CRFs are used to learn recognizer models based only on the words characteristics (word context). We claim that this approach reduce the ability of CRFs to learn NEs. In fact, NEs are sensitive to the word context but also to syntactic and semantic contexts. For this reason, we introduce a new extension of CRFs to consider multiple contexts.

3 Context-Based Approach for NER

CRFs model can be enriched by several features. In this section we show how we can use a rich annotated corpus to deduce semantic features. Thus, the hierarchical structure NE types in the training data are used. We define syntactic features as the POS tagger outputs.

3.1 Hierarchical NEs Space

Indexing newspapers applications use larger sets of NE types [13]. In that case it was found that the specific entity types were quite difficult to detect, and there were many sub-types for these categories. Thus, NEs are organized on types and sub-types (hierarchical structure). Each type domain can be considered as a conceptualization level. There are several levels of conceptualization and the number of NE types is exponential in comparison to the number of levels.

Each named entity l is represented as one concept l^1 or several concepts $l^1, l^2 \ldots, l^s$. Consequently, we have $l = l^1.l^2 \ldots l^s$ with the following semantic that each concept l^j is subsumed by the concept l^{j-1} for $2 \leq j \leq s$ and the concept l^1 is subsumed by the most general concept in our representation called *ENTITY*. Therefore, each concept is a node in the concept hierarchy and the NE is represented by a path in the tree structure. In the corpus annotation [3], each NE is defined at two or three levels of conceptualization. Thus, a label has the form $l = l^1.l^2$ or $l = l^1.l^2.l^3$ where $l^1 \in level(L_1)$, $l^2 \in level(L_2)$, and $l^3 \in level(L_3)$. For example, we associate the label *LOC.GEO* with *Paris*, where *LOC* corresponds to the most general concept which is *LOCATION*, *GEO* is the most specific concept that is *GEOGRAPHIC*. Using this hierarchy structure we can define the NEs at different level of conceptualization. For example, we can consider only the first level (*level(L1)*) and then consider only the general concepts of NEs (only the 6 NE types). But on the other hand, we can consider all levels and define a NE at several levels. This approach makes the NER task more complex by increasing the number of NE types. One solution is to consider each level independently of others and apply learning process at each level (construct one CRF model M_j for each conceptualization level j). These models can be enriched by additional information such as syntactic and semantic features. The following sections present the enrichment by a syntactic annotation and semantic enrichment using the hierarchical structure of NEs.

3.2 Syntactic Annotation as an a Priori Information

The CRFs approach constructs models which characterize input data. Each entity of the input data can be defined with several features (also called attributes). These attributes can be used in the learning phase to improve prediction models. Part of Speech (POS) tagging is the task of labeling each word in a sentence with its appropriate syntactic category called part of speech. In the NER process we can include the POS and lemma annotations as word attributes. To perform the syntactic parsing we use the *TreeTagger* [5] (precision of 96% claimed by authors). Thus for each level of conceptualization j the simple model M_j is constructed using the word and syntactic contexts. We can rewrite the local function in the Equation 3 including syntactic features:

$$\Psi_K(l^j_{i-1}, l^j_i, W, i) = \underbrace{\Psi_{K1}(l^j_{i-1}, l^j_i, W, i)}_{\text{word context}} \times \underbrace{\Psi_{K2}(l^j_i, S(w_i), i)}_{\text{syntactic context}}, \quad (5)$$

where Ψ_{K1} is a family of local functions depending only on word context $\{(f_k^w, \lambda_k^w)\}_{k=1}^{K1}$ and Ψ_{K2} is a family of local functions depending only on syntactic context $\{(f_k^s, \lambda_k^s)\}_{k=1}^{K2}$. We associate for each word w_i its syntactic category $S(w_i)$ where $S(w_i) \in \{$ *NAM, VERB, DET, PRE, ADV, ADJ, KON, NUM, PUN* $\}$. For example, the syntactic representation of the sentence $\langle$ *Albert Einstein was born on March 14, 1879*$\rangle$ is $\langle$ *NAM NAM VERB VER PRE NAM NUM PUN NUM* $\rangle$. We assume that $K = K1 + K2$.

3.3 Efficient Features Induction

To perform NE extraction, each data input w_i can be represented by several families of features. Traditionally, atomic observations of the word are used (such as the word itself, capitalization, prefixes and suffixes, neighboring words, etc.). Additional information like syntactic and semantic features can be used to enrich the model. The $M_{j/1 \leq j \leq s}$ models built separately (with word and syntactic features) give a prediction for all levels on the test data. Using these predictions, we can apply a combined learning by levels. Indeed, these predictions can be used as input novel features (family of hierarchical features) for the first level. In our experimentations we try to improve only the 6 general NE types (the NEs of the first level). Thus, the predictions of the second and third levels are used as input features for the first level in the combined process (Figure 2).

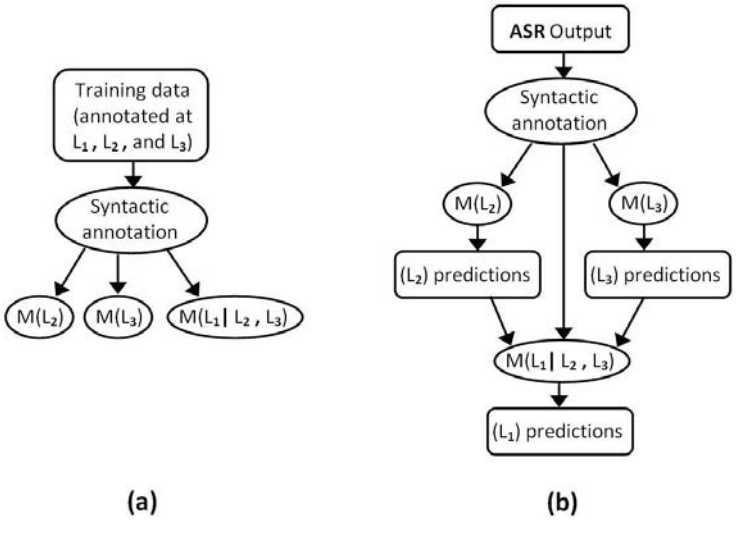


Fig. 2. (a) Models generation from training data, (b) models application on testing data: First we generate the POS tags of the words, after that we generate the predictions of the 2nd and 3rd levels using the models M_2 and M_3. Finaly, we use these tags and predictions as an a priori information to predict the level 1 annotations.

Using the equation 5, we calculate the simple predictions of the other levels $h/(h \neq 1)$. We use these predictions for the level 1 as the hierarchical context in the combined models M_1^{comb}:

$$\Psi_{K^{comb}}(l_{i-1}^1, l_i^1, W, i) = \underbrace{\Psi_{K1}(l_{i-1}^1, l_i^1, W, i)}_{\text{word context}} \times \underbrace{\Psi_{K2}(l_i^1, S(w_i), i)}_{\text{syntactic context}} \times \underbrace{\Psi_{K3}(l_i^1, Pred(i,1), i)}_{\text{hierarchical context}}, \quad (6)$$

where the function $Pred(i, 1)$ gives the predictions of all other levels of conceptualization $h/(h \neq 1)$ for the word w_i using simple models (equation 5). We assume that $K^{comb} = K1 + K2 + K3$. We built separately, for each level j, the simple model M_j with word and syntactic contexts (we use Ψ_{K1} and Ψ_{K2}). These models provide the predictions for each level $2 \leq j \leq s$. We also built for the first level the combined model M_1^{comb} with word, syntactic, and hierarchical contexts (we use Ψ_{K1}, Ψ_{K2}, and Ψ_{K3}). M_1^{comb} is a model for the first level knowing predictions of other levels (generated using simple models). This approach allows us to refine the models learned for each level, because we have a priori knowledge.

4 Experiments

We conducted the following experiments from transcribed speech data to investigate the performance of the proposed approach compared to the simple application of the CRFs to text data.

4.1 Data and Evaluation Metrics

Our experimentations has been done within the framework of the French Rich Transcription Program of Broadcast News ESTER [3]. The ESTER corpus consists of newspapers and radio broadcasts news, are segmented into sections. Each section is dedicated to a thematic set, which implies speakers and guests. We considered each sentence to be a training instance, with single words as tokens. The provided corpus consists of 100 hours made from four French speaking radio channels manually transcribed and annotated with tagset of about 40 NE types. Each radio program is represented by a transcription file. The annotation is carried by an expert of the domain and represented using an XML structure. This corpus is divided into three parts: the first one is used to NER learning models (*train* 84%) which will be used for the automatic recognition whereas the second part, development set (*dev* 8%), is used for the adjustment of inference parameters. The third part, test sets (*test* 8%), is used for the evaluation and the assessment of the learned models performances. *test* corpus is provided with manual transcriptions ($test_{REF}$) and ASR output streams ($test_{ASR}$). *dev* matches the training data, *test* is a four year gap difference. There are also two new radio channels in the test corpus which were not in the training and development data. The chosen NE tagset at the first level is made of 6 main categories: *PERSON*, *FUNCTION*, *ORGANISATION*, *LOCATION*, *DATE*, *AMOUNT*. The NEs are defined on three levels and each NE is defined on two or three levels. The total

number of NE types on all levels is 40. One of the characteristics of the corpus is the high ambiguity rate amount the NE types. For example, administrative region ($LOC.ADMI$) and geographic location ($LOC.GEO$). To evaluate our systems we have considered only the first level (6 NE types).

To measure the performance of each model we use four valuation measures: the recall (R), precision (P), F-measure (F), and the slot rate error (SER) [8]. To help in the analysis, we define the following symbols:

Re = total number of slots in the reference
H = total number of slots in the hypothesis
C = number of the correct slots
D = number of deletitions
I = number of insertions
S = number of substitutions

The number of substitutions S is composed by:

T = number of incorrect type and correct extent
E = number of incorrect extent and correct type
TE = number of incorrect type and incorrect extent

Thus the recall $R = \frac{C}{Re}$ is the percentage of reference slots for which the hypothesis is correct. The precision $P = \frac{C}{H}$ is the percentage of slots in the hypothesis that are correct. The F-measure $F = \frac{(1+\beta^2)RP}{\beta^2 P+R}$ is defined as the weighted harmonic mean of recall and precision. Finally, the $SER = \frac{\alpha_1 I+\alpha_2 D+(\alpha_3 T+\alpha_4 E+\alpha_5 TE)}{Re}$ is more accurate and penalizing than F-measure. α_i are the weighting parameters. In our evaluations we use $\alpha_1 = \alpha_2 = 1, \alpha_3 = \alpha_4 = 0.5, \alpha_5 = 1.5, \beta = 1$.

4.2 NER Evaluation on Manually Transcribed Speech

To design the NEs hierarchical structure we use corpus annotation. In opposition to [11], there is not ambiguous NE types like GPE (Geographical and Political Entity). We consider each sentence to be a training instance, with single words as tokens. For each word w_i, the word context (classical CRFs) consists of a window of 4 neighboring words $[w_{i-2}, w_{i-1}, w_i, w_{i+1}, w_{i+2}]$ and $|prefix| \leq 3$, $|suffix| \leq 3$. The syntactical context is defined by the POS label of the the word w_i and the POS labels of the neighboring words (CRFs+POS). We construct the combined model CCRFs which uses the predictions of the Level 2 and Level 3 as an input information. This model uses also the POS tags as an input features (CCRFs+POS). We compare these models to the classical CRFs. The resullts are given in Table 1.

In what follows, all the comparative values are expressed as relative gain compared to the classical CRFs approach. The results show that using syntactic features gives a significant improvement in performance (12% in SER). In addition, there is a small but consistent gain using hierarchical features. It is also interesting to note that improvement in NER results in a decrease in performance, since the test corpus has a four year gap difference. In the next subsection, we

Table 1. SER estimation for simple and enriched CRFs models with manual transcription. The last column gives the results obtained by the combined model (syntactic + semantic) and the relative gain compared to the classical model. The relative gain is obtained with this formula: $G = (NewVal - OldVal) * 100/OldVal$.

NEs (% occurrence)	**CRFs**	**CRFs+POS**	**CCRFs+POS**
PERS (26.3%)	47.29	25.04	24.4 (-48%)
LOC (24.5%)	34.64	21.34	20.81 (-39%)
ORG (20.9%)	55.18	47.23	43.96 (-20%)
TIME (17.2%)	27.65	25.65	25.11 (-9%)
AMNT (5.8%)	39.73	35.73	35.6 (-10%)
FUNC (5.2%)	69.23	66.6	64.8 (-6%)
ALL (100%)	43.11	31.63	**30.45** (-29%)

show how we adapt the text-based approach to the automatic speech recognition output.

4.3 Model Adaptation for ASR Systems

Because the spoken language differs with the written text, we need to preprocess the ASR transcription before applying our NER system. Thus, automatic speech recognition output must be segmented into its individual sentences. The NER task operates at the sentence level and assume the presence of standard punctuation. Sentence segmentation [1,14] task takes word sequences transcribed from an audio document and defines sequence boundaries. In our work we use the word-context features to annotate the words sequences. We take in consideration only full stop and comma punctuation marks. In addition to this, we use speaker diarization segmentation. Speaker diarization is the task of automatically partitioning an input audio stream into homogenous segments [6]. We consider only the segmentation error and not the true identity of speakers. We use the training corpus to construct sequences segmentation model (SSM) and speaker diarization model (SDM). This models are based on CRFs approach using word and syntactic features (see subsection 3.2). Evaluation results of the segmentation models are presented in Table 2:

Table 2. Recall, precision and F-measure of the ASR output segmentation using manual transcriptions as reference

Segmentation type	Recall (%)	Precision (%)	F-measure (%)
SSM	44.8	76.9	56.6
SSM using POS tag	53.0	77.3	62.9
SDM	56.2	69.5	62.1
SDM using POS tag	58.3	69.2	63.3
SSM+SDM	42.4	80.2	55.5
SSM+SDM POS tag	50.8	80.6	62.4

Table 3. F-measure and Slot Rate Error estimation on NER in manual and automatic speech transcriptions

	Manual annotation		ASR output	
	F-measure (%)	*SER (%)*	*F-measure (%)*	*SER (%)*
HMMs	64	48.3	34	65.78
CRFs	67	43.11	37	62.12
CRFs+POS tags	76	31.63	45	55.43
CCRF + POS tags	**78**	**30.45**	**64**	**53.92**

We note that the segmentation system has an acceptable precision (80%) with sentences and diarization model (*SSM+SDM POS tag*). We find that the use of POS labels improves significantly the recall with 18 % in the sentences segmentation model and only 2 % in the diarization segmentation. The reference punctuation are carried by an expert of the domain, these punctuation are not unique and can include ambiguities. Indeed, in some cases it has the same semantics using the point or comma. In the next NER evaluations, we will use the segmentation provided by the *SSM+SDM POS tag* model.

To evaluate the proposed approach with ASR output, segmentation is applied on the ASR output stream. This is a fundamental task, since the data are not presented in the same form.

Table 3 shows that the entity sensitivity is different between the manual transcriptions and ASR output, this is due to noise present in the automatic transcriptions. The automatic transcriptions have been produced using a vocabulary of 100K words and HMM-GMM automatic multispeaker speech recognition (ASR). It resulted in 11% of Word Error Rates (WER). We notice that the CCRFs+POS approach provides a gain of $37,5\%$ in SER measure and $21,8\%$ in F-measure compared to HMMs approach in manual transcriptions. POS tagging improves NER predictions with $26,6\%$ in SER measure in manual transcriptions and $11,2\%$ in ASR output. As shown, the proposed method achieved the best F-measure, 78% in manual annotation and 64% in ASR output, among the compared methods. It was $16,4\%$ in F-measure better than the baseline result. Figure 3 shows the evolution of SER and F-measure between manual and automatic transcriptions. We notice that the SER improvement follows the same curve in both types of transcription. On the other hand, the curve of the F-measure is better for the ASR output. This means that the proposed approach is robust to noise in automatic transcriptions.

For technical reasons, the encoding of evaluation files are used for non-accented languages. This implies that all the words with accents are not taken into account. By using upgraded file format, We improve of 5% the value of the SER measure of CCRF+POS approach (up from $30,45\%$ to 25.92% in the results of tables 2 and 3).

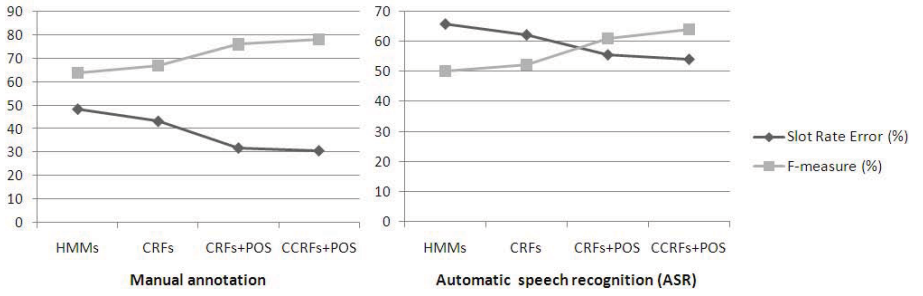


Fig. 3. Comparison of the evolution of SER and F-measure between manual and automatic transcriptons

5 Conclusion

In this paper, we presented a NER system in audio broadcast news transcriptions based on CRFs approach, which does not take into account only the language features, but also a set of semantic features. We showed that the discriminative approaches are more adapted than generative approaches when we have non-independent features with complex dependencies. The use of structures in the learning process provides a gain in quality labeling, because it provides an a priori knowledge of the data description. We proposed an adaptation of the text-based approach to a ASR output stream using a supervised segmentation and diarization model. In experiments using combined CRFs models, the proposed approach showed a better performance, in terms of ASR measure, than a simple application of CRFs approach.

Therefore, we demonstrated that in the field of semantic indexation, the conceptual representation surpasses its language representation. Graphical approaches allow the use of arbitrary features. Further research will consist in using audio signal characteristics as a new family features in the CRFs model. Harmonic to noise ratios (HNR) [4] of the speech source can be used as an ASR confidence in NER process. Thus, further work will consist in the integration of HNR values in our flexible CCRF model as criterion of selection for NER predictive models, since it provides a speech quality measure.

References

1. Doss, M.M., Hakkani-Tur, D., Cetin, O., Shriberg, E., Fung, J., Mirghafoli, N.: Entropy based classifier combination for sentence segmentation. In: Society, I.C. (ed.) Proceedings of international Conference on Acoustics, Speech, and Signal Processing, Honolulu, Hi, vol. 4, pp. 189–192 (2007)
2. Ekbal, A., Bandyopadhyay, S.: Named entity recognition using support vector machine: A langage independent approach. International Journal of Computer Science and Engineering, 155–170 (2008)

3. Galliano, S., Gravier, G., Chaubard, L.: The ester2 evaluation compaign for the rich transcription of french radio broadcasts. In: 10th annual Conference of the International Speech Communication Association, InterSpeech 2009 (2009)
4. Glotin, H., Vergyri, D., Neti, C., Potamianos, G., Luettin, J.: Weighting schemes for audio-visual fusion in speech recongnition. In: IEEE international conference on Acoustics Speech and Signal Process (ICASSP), Salt Lake City - USA (2001)
5. Helmut, S.: Part-of-speech tagging using decision trees. In: Proceedings of International Conference on New Methods in Language Processing (1994)
6. Jurafsky, D., Martin, J.H.: Speech and language processing: An introduction to natural language processing. In: Computational Linguistics, and Speech Recognition. Prentice Hall PTR, Englewood Cliffs (2000)
7. Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: ICML 2001: Proceedings of the Eighteenth International Conference on Machine Learning, pp. 282–289. Morgan Kaufmann Publishers Inc., San Francisco (2001)
8. Makhoul, J., Kubala, F., Schwartz, R., Weischedel, R.: Performance measures for information extraction. In: Proceedings of DARPA Broadcast News Workshop, pp. 249–252 (1999)
9. McCallum, A., Li, W.: Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In: Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003, Morristown, NJ, USA, pp. 188–191. Association for Computational Linguistics (2003)
10. Rabiner, L.R.: A tutorial on hidden markov models and selected applications in speech recognition. In: Proc. IEEE 77, vol. 2, pp. 257–286 (1989)
11. Sekine, S., Sudo, K., No-bata, C.: Extended named entity hierarchy. In: Proceedings of LREC 2002 (2002)
12. Sudoh, K., Tsukada, H., Isozaki, H.: Named entity recognition from speech using discriminative models and speech recognition confidence. Journal of Information Processing 17, 72–81 (2009)
13. Zidouni, A., Quafafou, M., Glotin, H.: Structured named entity retrieval in audio broadcast news. In: International Workshop on Content-Based Multimedia Indexing, pp. 126–131. IEEE Computer Society, Los Alamitos (2009)
14. Zimmermann, M., Tur, D., Fung, J., Mirghafori, N., Gottlieb, L., Shriberg, E., Liu, Y.: The icsi+ multi-lingual sentence segmentation system. In: Proceedings of the ICSLP 2006, Pittsburgh, pp. 117–120 (2006)

Lexical Chains Using Distributional Measures of Concept Distance

Meghana Marathe and Graeme Hirst

University of Toronto, Toronto, ON, M5S3G4
mm@cs.toronto.edu, gh@cs.toronto.edu

Abstract. In practice, lexical chains are typically built using term reiteration or resource-based measures of semantic distance. The former approach misses out on a significant portion of the inherent semantic information in a text, while the latter suffers from the limitations of the linguistic resource it depends upon.

In this paper, chains are constructed using the framework of distributional measures of concept distance, which combines the advantages of resource-based and distributional measures of semantic distance. These chains were evaluated by applying them to the task of text segmentation, where they performed as well as or better than state-of-the-art methods.

1 Introduction

Lexical chains are sequences of semantically related words in a text. A word is added to an existing chain only if it is related to one or more of the words already in the chain by a *cohesive* relation. In practice, the cohesion between two words is approximated either by term reiteration or by the *semantic distance* between them. Methods that restrict lexical cohesion to reiteration consider two terms to be related only if they are instances of the same word. Hence, these methods miss out on a significant portion of the semantic information inherent to a text.

Semantic distance is typically computed using linguistic resource-based measures or measures of distributional similarity, both of which have inherent disadvantages. This motivates the need for a hybrid that incorporates the advantages of both these methods. Mohammad and Hirst (2006) proposed *distributional measures of concept distance (DMCDs)* that combine distributional co-occurrence information with semantic information from a lexicographic resource, such as a thesaurus. These measures were shown to outperform traditional distributional measures on the tasks of correcting real-word spelling errors, and ranking word pairs in order of semantic distance. In this work, we build lexical chains using Mohammad and Hirst's framework of distributional measures of concept distance. The chains are evaluated by applying them to the task of text segmentation.

Text segmentation is the task of dividing a text document into cohesive units or segments by topic (Hollingsworth 2008). In particular, we focus upon *linear* segmentation, in which segments are not further subdivided; as opposed to *hierarchical* segmentation, where each unit may in turn be divided into sub-units.

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 291–302, 2010.

Morris and Hirst (1991) were the first to suggest using lexical chains for text segmentation, which has since become a standard application of lexical chains. Since lexical chains consist of semantically related words, each chain corresponds to a theme or topic (or a set thereof) in the text. As a result, lexical chains provide three useful cues, namely:

- A significant number of chains beginning at a point in text probably indicates the emergence of some new topic(s).
- A significant number of chains ending at a point in text probably means that certain topics are not discussed henceforth in the text.
- Points where the number of chains beginning or ending is not significant probably represent a continuation in the discussion of some topic(s).

Our hypothesis is that these cues help detect positions at which there are changes or shifts in topic, representing segment boundaries.

2 Background

This section provides a review of previous work in lexical chaining and text segmentation, and provides the motivation for the proposed method.

2.1 Lexical Chains

Halliday and Hasan (1976) laid the foundation for lexical chains, when they suggested relating words of a text back to the first word to which they are cohesively "tied". They also specified five types of lexical cohesion based on the dependency relationship between the words. However, they did not consider exploiting the transitivity of these relationships, nor did they discuss computational methods for finding lexical chains.

Morris and Hirst (1991) were the first to suggest computational means of building lexical chains. They used the hierarchical structure of *Roget's International Thesaurus, 4th Edition (1977)* to find lexical relationships between words. Based on their analysis of five texts, Morris and Hirst concluded that lexical chains computed by their algorithm correspond closely to the intentional structure[1] of that text produced from the structural analysis method of Grosz and Sidner (1986). Unfortunately, no online copy of the thesaurus was available to Morris and Hirst, so the algorithm was worked out by hand, preventing extensive tests.

There have since been several attempts at constructing lexical chains using WordNet (Fellbaum 1998), a large lexical database for English. The structure of WordNet being quite different from that of *Roget's*, researchers proposed new notions of semantic relatedness. Hirst and St-Onge (1998), for instance, classifed WordNet synset relations into upward, downward, and horizontal directions. For a given pair of words, the connections between some synset of one word and some synset of the other and the directions of these connections determine how related the words are.

[1] Intentional structure is based on the idea that every discourse has an overall purpose; and that every discourse segment has a purpose, specifying how it contributes to the overall purpose.

Stokes et al. (2004) proposed the use of lexical chaining as a means of segmenting news stories. They experimented with synonymy, specialization, and part-whole relationships from WordNet; and statistical word association as indicators of lexical cohesion for building chains. Even so, they concluded that optimal performance was achieved when only noun repetition patterns were examined during boundary detection.

Yang and Powers (2006) employed WordNet together with the Edinburgh Associative Thesaurus (EAT)[2] to build "improved" lexical chains called lexical hubs, for word sense disambiguation (WSD). The EAT consists of an associative network of words, constructed by asking subjects to state the first word they thought of in response to a stimulus word (Kiss et al. 1973). Since WordNet usually restricts itself to paradigmatic relations between words (Fellbaum 1998), the EAT was used to add associative information. This significantly improved results on the WSD task. However it limits the method's scope to resource-rich languages, requiring not only WordNet but also an associative thesaurus.

These methods suffer from WordNet's fine-grainedness, which has been a typical and frequent criticism of WordNet in the literature. Moreover, it is mainly the noun hierarchy of WordNet that has been extensively developed. Hence these methods cannot exploit the information contained in other parts of speech, such as verbs and adjectives.

Strength of a Chain. Lexical chaining algorithms often produce a much larger number of chains than desired for a particular task (Hollingsworth 2008). *Chain strength* is used to select the "best" or most relevant chains out of a given set of chains. Morris and Hirst (1991) first proposed the concept of chain strength, naming three factors that contribute to it: reiteration, density, and length. Reiteration is computed by counting the number of word-tokens of each word-type present in the chain. Chain density is the ratio of the number of words in a chain to the number of content words in the text (Hollingsworth 2008). The length or size of a chain is the number of word-types it contains. Morris and Hirst advocate using a combination of these three factors to compute chain strength.

In practice, chain strength has often been calculated as a weighted sum of the number of occurrences of each word-type in a chain (Barzilay and Elhadad 1997; Hirst and St-Onge 1998; Hollingsworth 2008). The value of a weighting coefficient depends on the kind of lexical relation used to add that term to the chain. It should be noted that this implicitly assumes that the same relation is used to add every occurrence of a word-type to a specific chain.

2.2 Text Segmentation

TextTiling (Hearst 1994, 1997) is widely considered a foundational work in paragraph-level text segmentation. It is an algorithm for partitioning expository texts into coherent multi-paragraph discourse units that reflect the underlying subtopic structure.

Instead of identifying individual subtopics, TextTiling focuses on detecting *subtopic shifts*. It assumes that a significant change in the vocabulary being employed is indicative of a shift from one subtopic to another. It uses term reiteration to detect these shifts. Thus, TextTiling does not depend on any lexical resource or inference mechanisms and can be applied to a variety of natural languages. Unfortunately, the algorithm requires setting several interdependent parameters, with no fixed way of determining the ideal values.

[2] http://www.eat.rl.ac.uk

Okumura and Honda (1994) used a Morris and Hirst style lexical chainer to determine segment boundaries. They hypothesized that when a lexical chain ends, there is a tendency for a segment to end; and when a new chain begins, it might indicate that a new segment has begun. Thus, sentence-gaps with the highest sum of the number of lexical chains beginning or ending at this gap are chosen as segment boundaries.

The authors reported preliminary but encouraging results on five Japanese texts. However they did not present any comparison of the performance of their algorithm with that of a baseline or of another algorithm such as TextTiling.

C99 (Choi 2000) is a domain-independent algorithm for linear text segmentation. A dictionary of word-stem frequencies in vector form is built for each tokenized sentence, and a similarity matrix is generated by computing the cosine similarity between every pair of sentences. Next, each value in the similarity matrix is replaced by its rank in the local region to generate a rank matrix. A text segment k is defined by two sentences, i and j, represented as a square region along the diagonal of the rank matrix. Segments are identified using divisive clustering based on Reynar's maximization algorithm (Reynar 1998).

C99 was shown to outperform TextTiling, DotPlot (Reynar 1998) and Segmenter (Kan et al. 1998) on an artificial test corpus.

2.3 Measures of Semantic Distance

We present a brief overview of the three major classes of methods used to compute semantic distance. For a more complete discussion, please refer to Mohammad and Hirst (2005), and Budanitsky and Hirst (2006).

Resource-based measures are computed using dictionaries, thesauri or wordnets. In a dictionary the semantic distance between two words may, for instance, be defined as the number of common words in the definitions of the two words (Lesk 1986). In a wordnet it could be defined by the amount of information shared by the nodes corresponding to the two words (Lin 1998b). In a thesaurus, semantic distance can be defined in terms of the length of the path between the two words through the category structure or index (Morris and Hirst 1991).

Most of these methods correlate well with human judgements (see Budanitsky and Hirst 2006), but they have several shortcomings due to their dependence on a specific resource, such as the inability to operate across parts of speech (e.g., the semantic distance between a verb and a noun); or the lack of consideration for non-classical relations (e.g., semantic role relation). It also means that they cannot be applied to languages in which those resources do not exist.

Distributional measures treat two words as semantically related if they tend to co-occur with similar contexts. These methods build one distributional profile (DP) per word, consisting of the number of occurrences of that word in various contexts. For example, if the target word is *deluminator* and the corpus contains the sentence *'It was a curious device, his deluminator.'*, the method increments the count of occurrences of *deluminator* in the context of *curious* and of *device*.

Measures of distributional similarity typically differ from each other in their notion of context (e.g., a window of *n* tokens *vs*. a syntactic argument relationship) and the technique used to incorporate co-occurrence information (e.g., conditional probability *vs*. pointwise mutual information).

These measures can be applied across parts of speech and they can also detect non-classical relationships provided these are reflected in the corpus. However, their correlation with human judgements is observed to be fairly low (Weeds 2003), and they require extremely large corpora in order to gather sufficient data. In addition, the methods run into problems with word sense ambiguity because they consider only the surface forms of words and not their meanings.

Hybrid methods aim to combine the advantages of resource-based and distributional methods by using both distributional information and a linguistic resource. Multiple hybrid methods have been proposed, but we discuss here the framework proposed by Mohammad and Hirst (2006).

Their framework of *distributional measures of concept distance (DMCDs)* combines distributional co-occurrence information with the semantic information from a lexicographic resource. Mohammad and Hirst used the categories from the *Macquarie Thesaurus* (Bernard 1986) as a set of coarse-grained word senses or concepts to build a word-category co-occurrence matrix (WCCM) using the sense-annotated *British National Corpus (BNC)*. Cell m_{ij} in the WCCM contained the number of times word i co-occurred (in a window of ± 5 words in the corpus) with any of the words listed under category j in the thesaurus. Distributional profiles of concepts (DPCs) could be derived from the WCCM by applying a suitable statistic, such as odds ratio or pointwise mutual information.

A DMCD is defined as any distributional measures in which DPCs of the categories of the target words are used as the context, in place of DPs of the words themselves. A DMCD is thus completely defined by choosing the window size (usually ± 5 words), the measure of distributional similarity, and the statistic used to measure the strength of association.

DMCDs were evaluated in comparison with distributional and WordNet-based measures on two tasks: ranking word pairs in order of semantic distance with human norms; and correcting real-world spelling errors. DMCDs outperformed distributional measures on both tasks. They did not perform as well as the best WordNet-based measures in ranking word pairs, but in the spelling correction task, DMCDs beat all WordNet-based measures except that of Jiang and Conrath (1997).

3 Method

In this section we describe the general algorithm used for building lexical chains, the procedure used for segmenting text using chains, and the two variants of the chaining algorithm that were implemented.

3.1 A General Algorithm for Lexical Chains

The lexical chaining algorithm is adapted from the one proposed by Morris and Hirst (1991). It requires the setting of three parameters: an indicator of lexical cohesion I (e.g.,

a measure of semantic distance); the threshold for adding a word to a chain, $threshold_a$; and the threshold for merging two chains, $threshold_m$. The range of acceptable values for the two thresholds depends upon the range of scores assigned by the method I. The algorithm requires a method $sim\_ww(x,y)$ that computes the lexical cohesion score between words x and y according to indicator I; and expects text in the form of a list of sentences from which punctuation and stop words have been eliminated.

For each word in the text, the algorithm computes the similarity score between that word and each existing chain using equation 1. If there are no existing chains, or if the maximum score obtained is lesser than $threshold_a$, a new chain containing that word is created.

$$sim\_wc(token, chain) = \underset{word \in chain}{average} \left(sim\_ww(token, word)\right) \tag{1}$$

If there is only one existing chain that obtains the maximum score, the word is added to that chain. If, however, more than one chain obtains the maximum score, these chains become candidates for merging. Similarity scores are computed between each pair of candidate chains using equation 2. If this score exceeds $threshold_m$, the two chains are merged; else the pair is removed from the candidate pairs. This eventually leads to one surviving candidate, to which the word is added. If no chains are merged, the word is added to the first merge candidate.

$$sim\_cc(chain1, chain2) = \underset{w1 \in chain1, w2 \in chain2}{average} \left(sim\_ww(w1, w2)\right) \tag{2}$$

Once all the words in the text have been processed, the algorithm halts, producing a list of lexical chains. Please refer to algorithm 1 for the pseudocode.

Algorithm 1. Building lexical chains

```
list_of_chains = empty
for each word in text do
  max_score = max_{c ∈ list_of_chains} (sim_wc(word, c))
  max_chain = argmax_{c ∈ list_of_chains} (sim_wc(word, c))
  if list_of_chains = empty OR max_score < threshold_a then
    Create new chain c containing word.
    Add c to list_of_chains.
  else if more than one max_chain then
    Merge chains if needed, adding the word to the resultant chain.
  else
    Add word to the chain max_chain.
  end if
end for
return list_of_chains
```

Interpretation of Parameter Values. Assuming that the indicator I assigns cohesion scores in the range $(0,1)$ (where 0 is assigned to semantically distant pairs of words), increasing $threshold_a$ beyond 0.8 yields highly conservative chains built mainly using

term reiteration, whereas decreasing it below 0.5 yields low-coherence chains where the relationship between words is often not clear. Similarly, a high value of $threshold_m$ leads to very infrequent merging; whereas a low value leads to merging of chains that are not very related to each other.

Chain Strength. As noted in section 2.1, chain strength calculations commonly make the assumption that the same relation is used to add every occurrence of a word-type to a specific chain. However, our algorithm uses $sim\_ww(x,y)$ scores to add words to chains, instead of directly perceptible relations. Thus different occurrences of the same word-type may be added to a chain with different scores. Hence, we eliminate weighting from the calculation of chain strength, effectively reducing it to the length or size of the chain.

3.2 Predicting Segment Boundaries

To choose segment boundaries, we use the scoring system described by Okumura and Honda (1994) coupled with a different way of determining the number of boundaries to predict. After chaining, every gap between a pair of consecutive sentences in the text is assigned a score equal to the number of chains beginning and ending at that gap. Boundaries are predicted at gaps whose score exceeds $threshold_{seg}$, computed as a function of the mean gap-score (see procedure 1). The parameter α can either be an absolute value (chosen by tuning it on a development set) or a function of the gap-scores (e.g., variance).

Procedure 1. $predict\_boundaries(text, \alpha)$

$score = empty$
$segment\_boundaries = empty$
for each gap i in $text$ **do**
  $score_i =$ number of chains beginning at $i +$ number of chains ending at i
end for
$threshold_{seg} = \underset{gap\ i \in text}{average}(score_i) + \alpha$
for each gap i in $text$ **do**
  if $score_i \geq threshold_{seg}$ **then**
    Add i to $segment\_boundaries$.
  end if
end for
return $segment\_boundaries$

3.3 Variants

In order to compare performance not only with a state-of-the-art segmentation method, but also with a resource-based semantic measure, we experiment with two variants of the general algorithm. Both use $threshold_a = 0.8$, $threshold_m = 0.5$, and $\alpha = 3$ (tuned using a development set), but differ in their choice of the indicator I:

- *LexChains-Lin*: Here I is Lin's WordNet-based measure (Lin 1998b), implemented in the WordNet::Similarity package (Pedersen et al. 2004). This measure estimates the semantic distance between two words using the amount of information shared by the nodes in WordNet corresponding to these words.

– *LexChains-Saif*: Here I is obtained using Mohammad and Hirst's framework of *distributional measures of concept distance*. In particular, we used Lin's measure of distributional similarity (Lin 1998a) with point-wise mutual information (PMI) as the measure of the strength of association. The Lin-PMI measure was chosen because it consistently performed as well as, if not better than, other DMCDs.

4 Evaluation

This section describes the data and methodology used and the results obtained in the evaluation of the lexical chaining method presented in the earlier section.

4.1 Data Preparation

Creating gold-standard text-segmentation data based on human judgements is very difficult, because intercoder agreement is fairly low (Hearst 1997; Passonneau and Litman 1993). To avoid this problem we used a corpus of research papers, with section- and subsection-boundaries acting as reference segments. Since research papers are written with a view of presenting information in a coherent and structured manner, we believe that the reference segments are a close approximation of gold-standard segments.

The ACL Anthology[3], sponsored by the Association for Computational Linguistics, is the NLP community's research repository. The ACL Anthology Reference Corpus (Bird et al. 2008) is an ongoing effort to provide a standardized reference corpus based on the ACL Anthology. It consists of:

– the source PDF files for articles in the Anthology, as of February 2007;
– raw text for all these articles, extracted automatically from the PDFs using non-OCR based text extraction; and
– metadata for the articles, in the form of BibTeX records.

When we say the text is *"raw"*, we mean that there is no mark-up (to delineate headings or sentences) and that extraction errors (e.g., '...' transcribed as 'Ä¢') have not been corrected. We used 20 raw-text documents from the ACL ARC corpus, manually marking segment boundaries at the end of each section or subsection larger than 2–3 sentences. A simple heuristic-based sentence boundary detection algorithm was used to convert the text into a list of sentences, from which punctuation and stop words were then stripped. This list was given as input to the text segmentation method.

4.2 Methodology

In order to test our hypothesis from section 1, we compare the performance of the two variants of the lexical chaining method on the task of text segmentation with that of JTextTile (Choi 1999), an improved version of TextTiling; and C99; both with default parameter settings.

A segment-boundary is defined by the number of the sentence it occurs after. A *strictly-correct* boundary is one that occurs at the same sentence-gap as a boundary

[3] Available at http://www.aclweb.org/anthology/

in the reference segmentation. A *nearly-correct* boundary is one that is either strictly correct or occurs one gap before or after a boundary in the reference segmentation. We evaluate the segmentation proposed by each method using three sets of measures:

- *Strict precision, strict recall, strict F-score*: Strict precision is the number of strictly-correct proposed segments divided by the total number of segments in the hypothesized segmentation. Strict recall is the number of strictly-correct proposed segments in the hypothesized segmentation divided by the number of segments in the gold-standard segmentation. Strict F-score is the harmonic mean of strict precision and strict recall. For all three measures, the higher the value, the better.
- *Relaxed precision, relaxed recall, relaxed F-score*: These measures are defined the same as their strict counterparts, except for nearly-correct boundaries.
- *Weighted and unweighted WindowDiff*: This metric (Pevzner and Hearst 2002) assigns a score in the range $(0,1)$ to a hypothesized segmentation, where a score of 0 indicates an exact match with the reference segmentation, and a score of 1 indicates that none of the proposed boundaries lie within k sentences of a reference boundary, k being half the average segment length. Weighted *WindowDiff* is defined as follows:

$$WindowDiff(ref, hyp) = \frac{1}{N-k} \sum_{i=1}^{N-k} \left| b(ref_i, ref_{i+k}) - b(hyp_i, hyp_{i+k}) \right| \quad (3)$$

  Here *ref* is the reference segmentation; *hyp* is the proposed segmentation; $b(p,q)$ is the number of boundaries between positions p and q in the text; and N is the total number of sentences in the text. The i is incremented at each sentence-boundary.

  On the other hand, unweighted *WindowDiff* assigns a penalty of one whenever the absolute difference between the number of boundaries in the reference and hypothesized segmentations (i.e. the value being summed over) exceeds zero.

4.3 Results

The precision, recall, F-score, and *WindowDiff* values for the four methods are reported in Table 1. The best score in each column is rendered in boldface. From the table, it is clear that the two lexical chaining methods, especially LexChains-Saif, outperform the other methods in all metrics.

The difference in the strict and relaxed scores of LexChains-Saif and LexChains-Lin is statistically insignificant[4]. The strict and relaxed scores for LexChains-Saif differ from those of C99 with a confidence interval of 90% and 98% respectively. Similarly, strict precision, and all relaxed scores for LexChains-Saif differ from those of JTextTile, with a confidence interval of 90% and 99% respectively.

While C99 performs nearly as well as LexChains-Saif on weighted *WindowDiff*, on unweighted *WindowDiff* LexChains-Saif outperforms C99 with a confidence interval of 90%, and JTextTile with an interval of 99%.

[4] We used the independent Student's t-test and the Wilcoxon signed-rank test to check whether two sets of samples (scores) arise from statistically different populations.

Table 1. Precision, recall, f-score, and *WindowDiff* values for JTextTile, C99, LexChains-Lin and LexChains-Saif, averaged over 20 documents

Method	Strict			Relaxed			*WindowDiff*	
	Precision	Recall	F-score	Precision	Recall	F-score	Weighted	Unweighted
JTextTile	13.2%	16.4%	14.2%	18.0%	21.9%	19.2%	0.625	0.56
C99	13.0%	14.6%	13.4%	20.4%	23.6%	21.3%	0.595	0.537
LexC-Lin	15.0%	**22.9%**	17.5%	24.7%	**35.8%**	28.3%	0.729	0.515
LexC-Saif	**18.5%**	18.9%	**18.0%**	**29.8%**	31.0%	**29.4%**	**0.577**	**0.463**

5 Conclusion

5.1 Summary of Results

Both variants of the lexical chaining method described significantly outperformed JTextTile (Choi 1999), an improved version of TextTiling (Hearst 1994, 1997). They also outperformed or performed as well as C99 (Choi 2000), a popular domain-independent text-segmentation algorithm. Of the two variants, LexChains-Saif, which used a DMCD, performed better overall than LexChains-Lin, which used Lin's WordNet-based measure (Lin 1998b). This proves our hypothesis.

5.2 Future Work

- *Effects of Genre*: The ACL ARC corpus (Bird et al. 2008) represents the very constrained genre of research papers in the area of Computational Linguistics. It would be interesting to analyze the performance of different measures of semantic distance on a variety of genres; and to investigate the effect(s) of document genre on the evaluation task.
- *Setting Parameter Values*: In this work, $threshold_a$, $threshold_m$ and α, the parameters of the lexical chaining algorithm, were tuned using a small development set. This in itself was difficult because the parameters are interrelated, making it hard to isolate their effects. It would be worthwhile exploring ways to determine their values automatically per set of documents or per genre.

References

Barzilay, R., Elhadad, M.: Using lexical chains for text summarization. In: Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization (ISTS 1997), Madrid, pp. 10–17 (1997)

Bernard, J.R.L. (ed.): The Macquarie thesaurus. Macquarie Library, Sydney (1986)

Bird, S., Dale, R., Dorr, B., Gibson, B., Joseph, M., Kan, M.-Y., Lee, D., Powley, B., Radev, D., Tan, Y.F.: The ACL Anthology Reference Corpus: A Reference Dataset for Bibliographic Research in Computational Linguistics. In: Proceedings of Language Resources and Evaluation Conference (LREC 2008), Marrakesh, Morocco (May 2008)

Budanitsky, A., Hirst, G.: Evaluating WordNet-based measures of semantic distance. Computational Linguistics 32(1), 13–47 (2006)

Choi, F.Y.Y.: Advances in domain independent linear text segmentation. In: Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference, pp. 26–33. Morgan Kaufmann Publishers Inc., San Francisco (2000)

Choi, F.Y.Y.: JTextTile: A free platform independent text segmentation algorithm. Software (1999), http://www.cs.man.ac.uk/~choif

Fellbaum, C. (ed.): WordNet: An Electronic Lexical Database. Language, Speech, and Communication Series. MIT Press, Cambridge (1998)

Grosz, B.J., Sidner, C.L.: Attention, Intentions, and the Structure of Discourse. Computational Linguistics 12(3), 175–204 (1986)

Halliday, M.A.K., Hasan, R.: Cohesion in English. Longman, London (1976)

Hearst, M.A.: Multi-paragraph segmentation of expository text. In: Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, Las Cruces, New Mexico, USA. Association for Computational Linguistics (June 1994)

Hearst, M.A.: TextTiling: Segmenting text into multi-paragraph subtopic passages. Computational Linguistics 23(1), 33–64 (1997)

Hirst, G., St-Onge, D.: Lexical chains as representations of context for the detection and correction of malapropisms. In: Fellbaum, C. (ed.) WordNet: An electronic lexical database, pp. 305–332. The MIT Press, Cambridge (1998)

Hollingsworth, W.A.: Using Lexical Chains to Characterise Scientific Text. PhD thesis, Clare Hall College, University of Cambridge (2008)

Jiang, J.J., Conrath, D.W.: Semantic similarity based on corpus statistics and lexical taxonomy. In: Proceedings of International Conference on Research on Computational Linguistics (ROCLING X), Taiwan (1997)

Kan, M.-Y., Klavans, J.L., McKeown, K.R.: Linear segmentation and segment significance. In: Proceedings of the 6th International Workshop of Very Large Corpora (WVLC-6), Montreal, Quebec, Canada, August 1998, pp. 197–205 (1998)

Kiss, G.R., Armstrong, C., Milroy, R., Piper, J.: An associative thesaurus of English and its computer analysis. In: Aitken, A.J., Bailey, R.W., Hamilton-Smith, N. (eds.) The Computer and Literary Studies. University Press, Edinburgh (1973)

Lesk, M.: Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In: SIGDOC 1986: Proceedings of the 5th annual international conference on Systems documentation, pp. 24–26. ACM, New York (1986)

Lin, D.: Automatic Retrieval and Clustering of Similar Words. In: Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Montreal, Quebec, Canada, August 1998, vol. 2, pp. 768–774. Association for Computational Linguistics (1998a)

Lin, D.: An Information-Theoretic Definition of Similarity. In: ICML 1998: Proceedings of the Fifteenth International Conference on Machine Learning, pp. 296–304. Morgan Kaufmann Publishers Inc., San Francisco (1998b)

Mohammad, S., Hirst, G.: Distributional measures as proxies for semantic relatedness (2005), http://ftp.cs.toronto.edu/pub/gh/Mohammad+Hirst-2005.pdf

Mohammad, S., Hirst, G.: Distributional measures of concept-distance: A task-oriented evaluation. In: Proceedings, 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006), Sydney, Australia (July 2006)

Morris, J., Hirst, G.: Lexical cohesion, the thesaurus, and the structure of text. Computational Linguistics 17(1), 21–48 (1991)

Okumura, M., Honda, T.: Word sense disambiguation and text segmentation based on lexical cohesion. In: COLING 1994: The 15th International Conference on Computational linguistics, Kyoto, Japan, vol. 2, pp. 755–761 (1994)

Passonneau, R.J., Litman, D.J.: Intention-based Segmentation: Human Reliability and Correlation with Linguistic Cues. In: Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics, Columbus, Ohio, USA, June 1993, pp. 148–155. Association for Computational Linguistics (1993)

Pedersen, T., Patwardhan, S., Michelizzi, J.: WordNet:: Similarity – Measuring the Relatedness of Concepts. In: Marcu, D., Dumais, S., Roukos, S. (eds.) HLT-NAACL 2004: Demonstration Papers, Boston, Massachusetts, USA, May 2004, pp. 38–41. Association for Computational Linguistics (2004)

Pevzner, L., Hearst, M.: A critique and improvement of an evaluation metric for text segmentation. Computational Linguistics 28, 1–19 (2002)

Reynar, J.C.: Topic segmentation: Algorithms and applications. PhD thesis, Computer and Information Science, University of Pennsylvania (1998)

Stokes, N., Carthy, J., Smeaton, A.F.: SeLeCT: a lexical cohesion based news story segmentation system. AI Communications 17(1), 3–12 (2004)

Weeds, J.E.: Measures and applications of lexical distributional similarity. PhD thesis, University of Sussex (September 2003)

Yang, D., Powers, D.M.W.: Word Sense Disambiguation Using Lexical Cohesion in the Context. In: Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions, Sydney, Australia, July 2006, pp. 929–936. Association for Computational Linguistics (2006)

Incorporating Cohesive Devices into Entity Grid Model in Evaluating Local Coherence of Japanese Text

Hikaru Yokono and Manabu Okumura

Precision and Intelligence Laboratory,
Tokyo Institute of Technology
yokono@lr.pi.titech.ac.jp, oku@pi.titech.ac.jp

Abstract. This paper describes improvements made to the entity grid local coherence model for Japanese text. We investigate the effectiveness of taking into account cohesive devices, such as conjunction, demonstrative pronoun, lexical cohesion, and refining syntactic roles for a topic marker in Japanese. To take into account lexical cohesion, we consider a semantic relation between entities using lexical chaining. Through the experiments on discrimination where the system has to select the more coherent sentence ordering, and comparison of the system's ranking of automatically created summaries against human judgment based on quality questions, we show that these factors contribute to improve the performance of the entity grid model.

1 Introduction

Models of evaluating text quality have been often used for evaluating not only human-written text in educational applications[1] but also automatically created one in the task of text generation, automatic summarization and machine translation. Text quality tends to be assessed with respect to grammaticality, coherence, and so on. We concentrate on modeling and evaluating text coherence in this paper.

Models of text coherence tend to be classified into two classes: local and global coherence. Local models of text coherence describe the relationships between adjacent sentences, and local coherence tends to be modeled mainly with surface information. Therefore, they have been used to impose an order on sentences in text. Barzilay and Lapata[2] recently proposed a promising local coherence model that is based on (so called) an entity grid and aims to learn abstract coherence properties, similar to those stipulated by Centering Theory[3], namely that adjacent sentences in locally coherent text are likely to contain the same nouns, and that important nouns often appear in syntactically important roles such as subject and object.

Text cohesion is the grammatical relationship within a text, and is considered as the link that holds a text together[4]. Since it is related to the coherence, to incorporate factors of text cohesion into text coherence model may improve the performance of the text coherence model. While the entity grid model can capture good properties of local

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 303–314, 2010.

coherence, we can obtain and leverage more cohesive devices in text for coherence modeling, such as lexical cohesion and conjunction, as we explain in section four[1]. In this paper, we try to incorporate such kinds of information on cohesive devices into the entity grid model and investigate their effectiveness, specifically in Japanese. We will show that these kinds of information contribute to improve the performance of the original entity grid model in the following two tasks: discrimination where the model has to select the more coherent sentence ordering of text, and comparing the model's ranking of automatically created summaries against human coherence judgment based on quality questions in the summarization evaluation.

In section two we briefly describe our related work. In section three we outline what the entity grid model is. In section four we explain some cohesive devices to be incorporated into the entity grid model. In section five we explain the experimental settings and the results.

2 Related Work

Barzilay and Lapata[2] proposed the entity grid model for assessing local coherence of text. This model focuses on transitions of entities' syntactic roles. The model is based on the idea that coherent text has regularity in transitions of syntactic roles as shown by Centering Theory[3].

Elsner and Charniak[5] pointed out that the entity grid model focused only on entities' transitions and they improved the model by taking into account the other factors that affect coherence, such as coreference expressions and discourse-new/old entities. Filippova and Strube[6] also extended the entity grid model by taking into account the semantic relatedness among entities and applied the extended model to German newspaper text. Hasler[7] evaluated computer-aided summaries by using the degree of strength of cohesion.

Barzilay and Lee[8] proposed using Hidden Markov Models (HMM) to assess global text coherence. They expressed topics in text as hidden states of HMM and assessed the global coherence of texts with their transitions. Soricut and Marcu[9] and Elsner et al.[10] proposed a model that considers local and global coherence at the same time, respectively. Their models are combinations of the HMM and the entity grid model.

For Japanese text coherence assessment, Itakura et al.[11] proposed an index that expresses the coherence in a paragraph. They calculated the index by using the semantic relatedness between words.

3 Entity Grid Model

An entity grid represents a document as a matrix with rows that correspond to sentences and columns that correspond to entities (nouns in the document). A grid's cell shows the

[1] Truly, Barzilay and Lapata also presented models which use coreference resolution systems. However, successive work tend to exclude the coreference information from the model since the systems might cause erroneous outputs.

s_1 [The Justice Department]$_S$ is conducting an [anti-trust trial]$_O$ against [Microsoft Corp.]$_X$ with [evidence]$_X$ that [the company]$_S$ is increasingly attempting to crush [competitors]$_O$.
s_2 [Microsoft]$_O$ is accused of trying to forcefully buy into [markets]$_X$ where [its own products]$_S$ are not competitive enough to unseat [established brands]$_O$.

Fig. 1. Example (Quoted from [2])

Table 1. Entity grid of Figure 1

	Department	Trial	Microsoft	Evidence	Competitor	Market	Product	Brand
s_1	S	O	S	X	O	-	-	-
s_2	-	-	O	-	-	X	S	O

syntactic role of each entity in each sentence. Syntactic roles are subject (S), object (O), neither (X), and absence (-). Table 1 is an entity grid constructed from text in Figure 1.

To assess local text coherence, a document vector is derived from the entity grid of the document. The vector's features are probabilities of syntactic transitions of sentence n-gram[2]. Furthermore, to consider the salience of entities based on frequency, the model generates features by computing transition probabilities for a salient group and a non-salience group of entities separately. An example of the vector is shown in Figure 2.

	SS	SO	SX	S-	OS	OO	OX	O-	XS	XO	XX	X-	-S	-O	-X	0S	0O	0X	0-	S1	O1	X1	-1	--
d_1	0	0	0	.06	0	0	0	.02	0	0	0	.08	.06	.02	.04	.02	.02	.06	.10	.02	.01	.02	.14	.32
d_2	.02	0	0	.04	0	0	0	.02	0	0	0	.08	.04	.04	.10	.02	0	0	.18	.02	.02	.02	.14	.26

Fig. 2. Example of document vector

To apply the model to local coherence evaluation for Japanese text, we decided the assignment of syntactic roles to the entities with a Japanese particle, as shown in Table 2.[3]

Table 2. Syntactic roles

Syntactic roles	Japanese particles
subject (S)	"が ga", "は ha"
object (O)	"を wo", "に ni"
neither (X)	other case particles
absence (-)	absence

[2] In Barzilay and Lapata[2], sentence 1∼3-gram were used. Therefore, we also use sentence 1∼3-gram in this paper.

[3] The particle "は ha" is usually used as a topic marker. However, as it often appears in the position of subject, we consider it as the subject role in this setting.

4 Cohesive Devices to Be Incorporated

Cohesion is roughly classified into *reference*[4], *conjunction*, and *lexical cohesion*[5].

Except conjunction that explicitly indicates the relationship between sentences, the other two classes are considered to be similar in that the relationship between sentences is indicated by two semantically same (or related) words. But lexical cohesion is far easier to identify than reference because both words in lexical cohesion relation appear in text while one word in reference relation is a pronoun or elided and has less information to infer the other word in the relation automatically.

In this paper, we investigate the effectiveness of the above mentioned three classes of cohesive devices: conjunction, reference, and lexical cohesion. In conjunction, we consider the relation between adjacent sentences and make entity grids for each class of conjunction relations separately. In reference, we do not take into account ellipsis and consider only demonstrative pronouns whose referents are explicit (for example, "この本 kono-hon" (this book) referring to a book in text), because we do not have a good coreference resolver for Japanese. For lexical cohesion, we devise the method to construct clusters of entities by lexical chaining. Furthermore, we try to refine syntactic roles for taking into account a topic marker in Japanese.

4.1 Calculation of Transition Probabilities for Each Conjunction Type

Conjunction is an important marker to indicate the relation between sentences. Assuming that distribution of entities' transitions might differ depending on the conjunction between adjacent sentences, we calculate transition probabilities of syntactic roles for each group of conjunctions.

We classify Japanese conjunctions into three groups as shown in Table 4, based on Ichikawa's classification[12] in Table 3.

Table 3. Types of conjunction in [12]

Types	Explanation	Example
copulative	the former sentence is a condition and the latter is its consequence.	"従って shitagatte" (therefore)
adversative	the latter sentence represents adversative content of the former.	"しかし shikashi" (however)
additive	the latter sentence represents additive content of the former.	"そして soshite" (and)
contrastive	the latter sentence represents contrastive content of the former.	"または mataha" (or)
transitive	the latter sentence makes a shift in content of the former.	"さて sate" (by the way)
parallel	the latter sentence represents similar content of the former.	"つまり tumari" (that is)
supplementary	the latter sentence represents supplementary content of the former.	"なぜなら nazenara" (because)
consecutive	the latter sentence represents consecutive content of the former.	"そうして sousite" (with that)

In adjacent sentences, we use a conjunction in the beginning of the latter sentence. When there is no conjunction between the sentences, we regard the type of conjunction as consecutive.

Consider the example of Fig. 3 and its entity grid (Table 5).

[4] Reference by pronouns and ellipsis in Halliday and Hasan's classification[4] are included here.

[5] Reference by full NPs, substitution and lexical cohesion in Halliday and Hasan's classification are included here.

Table 4. Three groups of Japanese conjunction

Groups	Types of conjunction	Explanation
Group 1	copulative, adversative	the relation connecting two matters in a logical way
Group 2	additive, contrastive, transitive	the relation connecting two separate matters
Group 3	parallel, supplementary, consecutive	the relation connecting two sentences of a matter

s_1. $[e_1]_S$ ... $[e_2]_O$ ...
s_2. **そして soshite (And),** $[e_2]_S$ ... $[e_3]_O$ ...
s_3. $[e_3]_S$ ... $[e_1]_X$ ...
s_4. $[e_4]_S$ ...

Fig. 3. Example 2

Table 5. Entity grid for Fig. 3

	e_1	e_2	e_3	e_4
s_1	S	O	-	-
s_2	-	S	O	-
s_3	X	-	S	-
s_4	-	-	-	S

Since "そして soshite" (And) is an additive conjunction in Group 2, the relation between s_1 and s_2 is Group 2. Relations between s_2 and s_3, and between s_3 and s_4 are Group 3, because s_3 and s_4 have no conjunctions. We calculate probabilities of entities' transitions for each group. For example, the probability of transition [S-] is 0.06 (computed as a ratio of its frequency in Group 3 (i.e. 1) divided by the total number of total transitions in Group 3 of length two (i.e. 16)). Fig. 4 show a document vector for Fig. 3.

SS_{G_1}	$\cdots$	$--_{G_1}$	SS_{G_2}	$\cdots$	$S\text{-}_{G_2}$	$\cdots$	$--_{G_2}$	SS_{G_3}	$\cdots$	$S\text{-}_{G_3}$	$\cdots$	$--_{G_3}$
0	$\cdots$	0	0	$\cdots$	.25	$\cdots$	.25	0	$\cdots$	.06	$\cdots$	.13

Fig. 4. Example of document vector (a part)

4.2 Reference

In reference, we consider only demonstrative pronouns whose referents are explicit. Specifically, we treat noun phrases containing a demonstrative pronoun, such as "この kono" (this) and "あの ano" (that), as our target anaphoric expressions. The antecedents of these anaphoric expressions tend to occur in the preceding sentence. Thus, if there is a demonstrative pronoun in one sentence and its antecedent does not appear in the preceding, we can presume the relation between these sentences is not so coherent.

To incorporate the reference information into the model, we add to a document vector another feature representing the probability that the referent entity occurs in the preceding sentence. The value of the feature is the ratio of the number of cases where the preceding sentence of a sentence containing an entity with a demonstrative pronoun contains the entity, by the total number of sentences containing the anaphoric expression.

Barzilay and Lapata[2] used a coreference resolution system to attempt to improve the entity grid, but with mixed results. In contrast, we do not attempt to use any automatically identified coreference links.

4.3 Lexical Cohesion

The entity grid model treats entities independently when computing transition probabilities. Therefore, it cannot capture the factor of lexical cohesion between entities. We address this problem by clustering entities semantically. In this paper, we consider lexical chaining.

A lexical chain[13] is a sequence of words that are semantically related to each other. In this paper, we regard a chain as a cluster. We construct lexical chains in text by using Mochizuki et al.[14]'s system.

In their method, we first calculate the cooccurrence score of entities X,Y with the cosine measure (1):

$$cos(X,Y) = \frac{\sum_{i=1}^{n} x_i \times y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \times \sqrt{\sum_{i=1}^{n} y_i^2}}, \tag{1}$$

where x_i,y_i are the number of occurrences of X,Y in text i and n is the number of texts in the corpus. Then, we calculate the similarity between clusters with equation (2):

$$sim(C_i, C_j) = \max_{X,Y} cos(X \in C_i, Y \in C_j), \tag{2}$$

where X,Y are entities in clusters C_i,C_j respectively.

The clustering algorithm is as follows. We pick a sentence at the beginning of the text and group entities whose similarity is above the threshold into a cluster in descending order of similarity. Then, we repeat the process on calculated clusters until no sentences left in the text. In the experiment in section five, we calculated the cooccurrence score using 100 articles in 2003 of Asahi newspaper corpus and set the threshold of 0.35.

When clustering entities, a cluster sometimes consists of entities that have several syntactic roles. In that case, we can merge those syntactic roles into a role by using one of the following two methods:

method 1 (1st): choosing a syntactic role transition based on the preference order of syntactic roles[6].

method 2 (comb): using all the transitions by combining syntactic roles in a cluster.

Figure 5 shows an example. In the figure, e_1, e_2, and e_3 are entities and c1 is a cluster that consists of e_2 and e_3. With method 1, [OS] is the only transition between sentences s_1 and s_2 for cluster c_1 based on the preference. On the other hand, with method 2, all the combinations of syntactic roles, [OO], [OS], [-O] and [-S], are the transitions for cluster c_1.

4.4 Refining Syntactic Roles

In Japanese, a particle "は ha" is used as a topic marker. A document is considered to have low coherence if its topic changes frequently. Furthermore, an entity with a particle that modifies a predicate of a sentence is considered to be more important than an entity that does not modify it.

[6] Preference order: S>O>X.

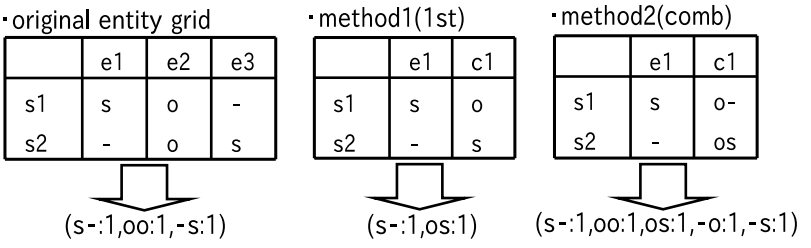


Fig. 5. Selection of syntactic roles

Table 6. Syntactic roles (refined)

Syntactic roles	Japanese particles
subject (S)	“が ga”
object (O)	“を wo”, “に ni”
topic (H)	“は ha”
predicate factor (R)	particles modifying a predicate
others (X)	other case particles
absence (-)	absence

Therefore, we refine the original four types of syntactic roles into the six types in Table 6. In this set, we assume the following preference order: H>S>O>R>X (c.f. [15]). We used a Japanese dependency analyzer, CaboCha[7], for obtaining the modification relation.

5 Experiments

We evaluated our models using the following two tasks: discrimination where the model has to select the more coherent sentence ordering of text, and comparing the model’s ranking of automatically created summaries against human coherence judgment based on quality questions in the summarization evaluation. Those are based on the ones in Barzilay and Lapata[2].

The models to be compared are shown in Table 7. In each model, we separated salient entities from non-salient entities in creating grid, and the threshold of salience is two. We evaluated both the case taking conjunction into account (+CONJ) and the case without conjunction (no CONJ). Baseline is the original entity grid model with four syntactic roles.

5.1 Task 1: Discrimination of More Coherent Sentence Ordering

Since it is very difficult to judge which text is more coherent, we made an assumption, like Barzilay and Lapata, that an original human-authored text is in general more coherent than its permutation. In the discrimination task, a document is compared with a

[7] http://www.chasen.org/~taku/software/cabocha/

Table 7. Models to be compared

Models	Explanation
Baseline	original model
CONJ	with conjunction
REF	with reference
LC(1st)	with lexical chains(1st)
LC(comb)	with lexical chains(comb)
SR(H)	with refined syntactic roles
ALL-LC(1st)	SR(H)+REF+LC(1st)
ALL-LC(comb)	SR(H)+REF+LC(comb)

Table 8. Types of reordering

Permutations	Explanation
random	reordering randomly
swap1	swapping a pair of sentences
swap2	swapping two pairs of sentences
swap3	swapping three pairs of sentences
mix	mixing 1, 2, and 3 swaps equally

random permutation of its sentences, and we score the system correct if it indicates the original as more coherent.

Discrimination might be easier since a random permutation is likely to be much less similar to the original. Therefore, we also tested our models on the task of other four permutation settings, as in Table 8. The smaller the difference is, the more difficult the task is.

We used 100 and 300 articles in 2003 from Asahi newspaper corpus. This dataset was composed of three domains: politics, health, and education.

As mentioned, the task was to estimate the rank of a given pair of texts. We used ranking SVM with SVM^{light}[8] to train parameters of the ranking function. In the experiment, we performed 10-fold cross validation and evaluated the accuracy.

Table 9 shows the results of models for 100 and 300 articles with mix permutation. Italics indicates that accuracy was lower than the baseline, and bold face indicates the highest accuracy. Symbols $^{**}(p < 0.01)$ and $^{*}(p < 0.05)$ indicate the accuracy was significantly different from the baseline accuracy (using the sign test).

Table 9. Accuracy for each model (task 1)

	mix(100)		mix(300)	
	no CONJ	+CONJ	no CONJ	+CONJ
Baseline	0.547	–	0.580	–
CONJ	–	0.551	–	*0.573*
REF	*0.546*	0.550	*0.579*	*0.573*
LC(1st)	0.567	0.582*	0.585	0.677**
LC(comb)	0.549	0.557	**0.614**	**0.773****
SR(H)	0.558	0.562	*0.579*	*0.560***
ALL-LC(1st)	**0.574**	0.577*	*0.574*	*0.563*
ALL-LC(comb)	0.564	**0.594****	*0.579*	0.586

[8] http://www.cs.cornell.edu/people/tj/svm_light/

Table 10. Accuracy for each permutation (task 1)

Model		swap1	swap2	swap3	mix	random
no CONJ	Baseline	0.532	0.552	0.584	0.547	0.687
	ALL-LC(comb)	0.544	0.566	0.609	0.564	**0.725****
+CONJ	CONJ	0.560**	0.560	**0.666****	0.551	0.696
	ALL-LC(comb)	**0.565****	**0.633****	0.642**	**0.594****	0.719*

On the whole, our model with new factors were better than the baseline model. The models that transition probabilities was calculated for each conjunction type (the column of "+CONJ") was better than the models without conjunction (the column of "no CONJ"). Therefore, we think that the conjunction information between sentences contributes the evaluation of text coherence.

With refined syntactic roles, the combination of transitions increased and the number of features also increased. We think this is why this model (SR(H)) suffered from the effect when the size of dataset was small. When considering reference, the model (REF) made little improvements, because our target expressions were limited. The model with lexical chains could improve the performance.

The models with all our factors (ALL-LC(1st), ALL-LC(comb)) were not the best in all cases. We think the reason was the difference of the number of features in a document vector, by taking into account the refined syntactic roles.

Table 10 shows the results for each permutation. With respect to the difficulty of the task, we obtained the valid results showing that random permutation, which was the easiest task, was the best and swap1 permutation, which was the most difficult, was the worst. Compared with the baseline, the model with three groups of conjunction was effective for the more difficult tasks.

5.2 Task 2: Summary Evaluation

In the discrimination task, the distribution of entity frequency is same between the compared texts because they were the same originally. However, we consider this is a rare situation. Therefore, we did another experiment on text coherence evaluation of automatically created summaries.

We used the results of human judgment of summaries created by systems in TSC3 (Text Summarization Challenge 3)[9], which is a subtask of NTCIR-4[10] [16]. In the human judgment, human subjects used fifteen quality questions on readability to evaluate the quality of the summary texts[17]. These questions are a Japanese modified version of DUC's quality questions[18]. Appendix A shows the nine questions on coherence used for the following score calculation. We calculated the scores of the summaries based on the results of the human judgment. An answer to each question was basically the number of problematic places of the target summary. We calculated the sum of the answer for each question normalized by the number of sentences as the summary's score.

[9] http://www.lr.pi.titech.ac.jp/tsc/index-en.html
[10] http://research.nii.ac.jp/ntcir/ntcir-ws4/ws-en.html

Table 11. Accuracy for each model (task 2)

	no CONJ	+CONJ
Baseline	0.502	–
CONJ	–	0.507
REF	0.502	0.507
LC(1st)	0.540**	**0.563****
LC(comb)	0.586**	0.557**
SR(H)	0.503	0.505
ALL-LC(1st)	0.552**	0.549**
ALL-LC(comb)	**0.589****	0.546**

Table 12. Accuracy for score difference (task 2)

Model		all (6434)	0∼0.5 (2986)	0.5∼1.0 (2014)	1.0∼1.5 (930)	1.5∼2.0 (334)
no CONJ	Baseline	0.502	0.500	0.502	0.505	0.515
	ALL-LC(comb)	**0.589****	**0.532***	**0.592****	**0.669****	**0.775****
+CONJ	CONJ	0.507	0.508	0.502	*0.503*	0.536
	ALL-LC(comb)	0.546**	0.510	0.547**	0.604**	0.689**

In a pair of summaries which are created from the same text, we assume that the summary with the lower score was more coherent than the other and used the same criterion for the discrimination task. We used 300 articles of mix and random permutation in the discrimination task as training data.

Table 11 shows the results using all test data. Symbols and font types have the same meaning as in the results of the discrimination task.

Because the domain is different between the training set and test set, the accuracy was lower than the one of the discrimination task. However, most of our models are better than the baseline model.

The models considering conjunction type (+CONJ) were not always better than the models without conjunction type (no CONJ). The results for other factors showed the same tendency as the ones of the discrimination task.

The bigger the difference between compared texts is, the easier evaluating its coherence is. Therefore, we calculated the accuracy for the pairs of summaries by dividing the difference of the scores. Table 12 shows the results. The "all" column shows the result for all the pairs. The number in parentheses shows the number of pairs in the range. We obtained the valid results showing that the accuracy for the biggest difference (1.5∼2.0) was the best, and the accuracy for the smallest difference (0∼0.5) was the worst. The bigger the difference was, the more effective the models with reference, lexical chains and refined syntactic roles were.

6 Conclusion

We tried to incorporate kinds of cohesive devices into the entity grid model and investigated their effectiveness, specifically in Japanese: conjunction, demonstrative pronoun, and lexical cohesion. We showed that these kinds of information contribute to improve the performance of the original entity grid model in the two tasks: discrimination where

the model has to select the more coherent sentence ordering of text, and comparing the model's ranking of automatically created summaries against human coherence judgment based on quality questions in the summarization evaluation.

In a local coherence model based on the entity grid, we cannot evaluate the coherence of a part of a document because the document is transformed into a vector. If text coherence evaluation is used not for assessment of text but for assisting in revision, we need to evaluate whether parts of a document are coherent. Therefore, future work will include using a text coherence model for fragments of a text.

References

1. Miltsakaki, E., Kukich, K.: Evaluation of text coherence for electronic essay scoring systems. Natural Language Engineering 10(1), 25–55 (2004)
2. Barzilay, R., Lapata, M.: Modeling local coherence: An entity-based approach. Computational Linguistics 34(1), 1–34 (2008)
3. Grosz, B.J., Weinstein, S., Joshi, A.K.: Centering: a framework for modeling the local coherence of discourse. Computational Linguistics 21(2), 203–225 (1995)
4. Halliday, M.A.K., Hasan, R.: Cohesion in English. Longman, London (1976)
5. Elsner, M., Charniak, E.: Coreference-inspired coherence modeling. In: Proceedings of ACL 2008: HLT, Short Papers, pp. 41–44 (2008)
6. Filippova, K., Strube, M.: Extending the entity-grid coherence model to semantically related entities. In: Proceedings of the 11th European Workshop on Natural Language Generation (2007)
7. Hasler, L.: Centering theory for evaluation of coherence in computer-aided summaries. In: Proceedings of the Sixth International Language Resources and Evaluation (2008)
8. Barzilay, R., Lee, L.: Catching the drift: Probabilistic content models, with applications to generation and summarization. In: HLT-NAACL 2004: Main Proceedings, pp. 113–120 (2004)
9. Soricut, R., Marcu, D.: Discourse generation using utility-trained coherence models. In: Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions, pp. 803–810 (2006)
10. Elsner, M., Austerweil, J., Charniak, E.: A unified local and global model for discourse coherence. In: Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference, pp. 436–443 (2007)
11. Itakura, Y., Shirai, H., Kuroiwa, J., Odaka, T., Ogura, H.: Availability of evaluation method of paragraph consistency with words semantic relation. IPSJ SIG Technical Reports NL-183, pp. 107–113 (2008) (in Japanese)
12. Ichikawa, T.: KokugoKyouiku no tame no BunshouronGaisetsu. KyouikuShuppan (1978) (in Japanese)
13. Morris, J., Hirst, G.: Lexical cohesion computed by thesaural relations as an indicator of the structure of text. Computational Linguistics 17(1), 21–48 (1991)
14. Mochizuki, H., Iwayama, M., Okumura, M.: Passage-level document retrieval using lexical chains. Natural Language Processing 6(3), 101–126 (1999) (in Japanese)
15. Walker, M., Iida, M., Cote, S.: Japanese discourse and the process of centering. Computational Linguistics 20(2), 193–232 (1994)
16. Kando, N.: Overview of the fourth ntcir workshop. In: Working Notes of the 4th NTCIR Workshop meeting (2004)

17. Hirao, T., Okumura, M., Fukushima, T., Nanba, H.: Text summarization challenge 3 - text summarization evaluation at ntcirworkshop 4. In: Working Notes of the 4th NTCIR Workshop Meeting (2004)
18. Over, P., Liggett, W.: Introduction to duc: An intrinsic evaluation of generic news text summarization systems (2002), http://www-nlpir.nist.gov/projects/duc/pubs/2002slides/overview.02.pdf

Appendix A: Quality Questions Used for Score Calculation

- How many redundant or unnecessary sentences are there?
- How many places are there where (zero) pronouns or referring expressions to be used?
- How many pronouns are there whose antecedents are missing?
- How many proper nouns which appeared in the unsuitable position are there?
- How many expressions which have same meanings but different term are there?
- How many of the sentences are missing important constituents?
- How many places are there where conjunctions should be supplied or conjunctions should be deleted?
- How many unnecessary words (adverbs, adjectives, etc.) are there?
- Does the summary has wrong chronological ordering?

A Sequential Model for Discourse Segmentation

Hugo Hernault, Danushka Bollegala, and Mitsuru Ishizuka

Graduate School of Information Science and Technology,
The University of Tokyo 7-3-1 Hongo,
Bunkyo-ku, Tokyo 113-8656, Japan
hugo@mi.ci.i.u-tokyo.ac.jp, danushka@mi.ci.i.u-tokyo.ac.jp,
ishizuka@i.u-tokyo.ac.jp

Abstract. Identifying discourse relations in a text is essential for various tasks in Natural Language Processing, such as automatic text summarization, question-answering, and dialogue generation. The first step of this process is segmenting a text into elementary units. In this paper, we present a novel model of discourse segmentation based on sequential data labeling. Namely, we use Conditional Random Fields to train a discourse segmenter on the RST Discourse Treebank, using a set of lexical and syntactic features. Our system is compared to other statistical and rule-based segmenters, including one based on Support Vector Machines. Experimental results indicate that our sequential model outperforms current state-of-the-art discourse segmenters, with an F-score of 0.94. This performance level is close to the human agreement F-score of 0.98.

1 Introduction

Discourse structures have an important role in various computational tasks, such as creating text summaries [1], performing question-answering [2], generating dialogues [3], or improving the processing of clinical guidelines [4]. For example, in automatic text summarization, if we know that a particular segment of text further elaborates an already stated fact, then we can safely ignore the elaborated segment to create a concise summary of the text. However, despite the wide uses of discourse parsing, true automatization of these tasks is preconditioned by the availability of efficient discourse parsers. In the past few years, several research efforts have aimed at building automatic parsers. In particular, a number of authors have attempted to create discourse parsers in the framework of the Rhetorical Structure Theory (RST) [5], one of the most prevalent discourse theories.

The general problem of automatically annotating a text with a set of discourse relations can be decomposed into three sub-problems. First, the text is divided into non-overlaping units, called *elementary discourse units* (EDUs). Each discourse theory has its own specificities in terms of segmentation guidelines and size of units. In RST, units are essentially clauses, and a sentence may be segmented in the fashion of Figure 1.

Second, we must select, from a pre-defined set, which discourse relations hold between consecutive pairs of EDUs. In previous work on discourse tagging, this

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 315–326, 2010.

[The posters were printed on paper]^{1A} [pre-signed by Mr. Dali,]^{1B} [the attorneys said.]^{1C}
(wsj$_{1331}$)

Fig. 1. A sentence split into three EDUs

problem has been modeled as a supervised classification task: a multi-class classifier is trained to identify the discourse relation holding between two given EDUs [6]. Last, we must construct a single discourse tree depicting the way all discourse relations and EDUs of the text relate to each-other. This paper focuses on the first sub-problem, segmenting a given text into a sequence of EDUs.

The overall accuracy of discourse parsing depends on the initial segmentation step. Indeed, if the text is wrongly segmented during this first stage, it becomes unreliable to assign correct discourse relations or build a consistent discourse tree for the text [7]. Therefore, the discourse segmentation task is of paramount importance to any discourse parsing algorithm.

As described later in Section 1.1, existing discourse segmentation algorithms either use a set of hand-coded rules or a supervised classifier. Considering the heterogenous texts that must be processed by a discourse parser, it is not feasible to write rules to cover every segmentation criteria. Thus, modeling the discourse segmentation problem as a classification task [8], and training a classification model from human-annotated data, is an interesting solution to the problems encountered with rule-based discourse segmentation. In the classification approach, given a word and its context in the text, the system determines whether there is likely an EDU boundary. Commonly-used features include punctuation and cue phrases (e.g., *but*, *and*, *however*). However, this method does not consider prior segmentation decisions as it encounters a new word. Each word is considered individually by the classifier. Therefore, the decisions made by a classification approach are locally-motivated. We propose a sequence labeling approach to discourse segmentation, that finds the globally optimum segmentation into EDUs for a given text. In contrast to the classification approach to segmentation, the model proposed in this paper considers its own previous decisions before it determines whether it should impose a discourse boundary.

1.1 Related Work

In 'SPADE' [7], a segmenter based on a probabilistic model is implemented, as the first step to a sentence-level discourse parser. This classifier is trained on the RST Discourse Treebank corpus (RST-DT) [9], and then used to differentiate between EDU boundary and non-boundary words. The boundary probability is calculated by counting occurrences of certain lexico-syntactic patterns in the training corpus. The segmenter yields an F-score of 0.831 (0.847 when using perfect parse trees). Although the features used by the authors are very relevant, the model shows its limits when compared to more elaborated probabilistic approaches.

A statistical discourse segmenter, based on artificial neural networks, is presented in [10]. The system is also trained on the RST-DT, and uses syntactic and lexical information, in particular discourse cues. A multilayer perceptron model is employed, and bagging is used in order to reduce overfitting. The performance of the segmenter is comparable to [7], with an F-score of 0.844 (0.860 when using perfect parse trees).

Rule-based discourse segmenters have also been created: In [11], segmentation is done by a multi-step algorithm, which uses syntactic information and discourse cues. An F-score of 0.80 is reported for this system. Then, in [12], it is argued that using rules has certain advantages over automatic learning methods. Indeed, the proposed model does not depend on a specific training corpus, and allows for high-precision segmentation, as it inserts fewer but 'quality' boundaries. However, in the latter system, segmentation is done in a manner different from the segmentation guidelines used in the RST-DT corpus. First, the authors avoid building short EDUs, in order to avoid relations with lesser informative content, such as SAME-UNIT or ELABORATION. Then, complement clauses are not placed in autonomous units. For instance, *'He said that'* is not considered an EDU. In this paper, we will not enter the discussion of what constitutes the best segmentation guidelines, and focus instead on the supervised methods for learning segmentation efficiently from the RST-DT corpus.

We already pointed out that discourse segmentation is necessary in order to build an automatic discourse parser. It is however interesting to note that the conception of a discourse relation analyzer is possible without treating the segmentation problem. In [6], a discourse parser using a multi-class Support Vector Machines classifier for relation identification, and a greedy bottom-up tree-building algorithm is described. The algorithm is first evaluated on perfectly-segmented EDUs from the RST-DT. Then, in order to create a fully automatic system, the authors build on top of the discourse segmenter of [7]. When using perfect segmentation, the discourse parsing pipeline returns a F-score of 0.548, but this score drops to 0.443 when using SPADE's segmenter. Here, we clearly see the critical role of the segmentation component, which can easily act as a bottleneck, and pull down the performance of the whole system.

Finally, it is worthy to note that the study of discourse segmentation, although seemingly restricted in its scope, can potentially be beneficial to all applications in which discourse relations or discourse structures are used.

2 Method

2.1 Outline

We model the problem of discourse segmentation as a sequential labeling task. Although there have been classifier-based approaches to discourse segmentation such as neural network-based methods [10], to our knowledge, this is the first attempt to model discourse segmentation as a sequential labeling problem. To illustrate the proposed sequential labeling approach, let us consider the example sentence shown in Figure 2. Therein, BOS and EOS respectively denote the

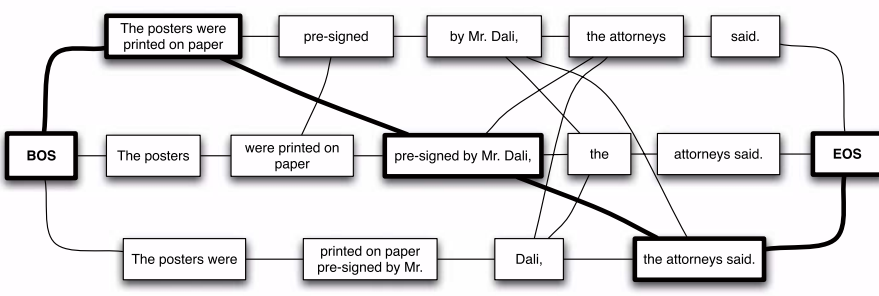


Fig. 2. Some possible segmentation decisions

beginning of the sentence and the *end of the sentence*. Because EDUs are not allowed to cross sentence boundaries, BOS and EOS act as segment boundaries. There are multiple possible ways in which we can segment the text inside a sentence. These different ways can be represented by a lattice structure, like the one shown in Figure 2. Then, the problem of segmenting a given sentence into EDUs can be seen as searching for the best path connecting BOS to EOS on this lattice. This best path is the one that maximizes the likelihood – alternatively, we can consider minimizing a cost function – of the segmentation process. Modeling a sentence as a lattice and then searching for the best path is a technique used in various related tasks in Natural Language Processing, such as part-of-speech (POS) tagging, chunking, and named-entity recognition (NER). The best segmentation path in our example sentence is shown in bold in Figure 2.

In Section 2.2, we introduce Conditional Random Fields [13], the sequential labeling algorithm that we use to find the best path on the lattice. CRFs have shown to outperform other sequential labeling algorithms such as Hidden Markov Models (HMMs) and Maximum Entropy Markov Models (MEMMs), on a wide range of tasks including POS tagging, chunking and NER [13]. In Section 2.3 we describe how we employ CRFs for the task of discourse segmentation. The features that we use for training are described in Section 2.4.

2.2 Conditional Random Fields

Conditional Random Fields or Markov Random Fields are undirected graphical models that express the relationship between some random variables. Each vertex in the undirected graph represents a random variable. Some of those variables might be observable (e.g., the frequency of a particular word), whereas other variables cannot be observed (i.e., the POS tag of the word). In Conditional Random Fields, by definition each random variable must obey the Markov property with respect to the graph (i.e., each node is independent from the other nodes, when conditioned upon its Markov blanket). In the general setting where the Conditional Random Field represents an arbitrary undirected graph, each clique in the graph is assigned with a potential function. However, in the sequence labeling task, the undirected graph reduces to a linear chain, as the one shown in Figure 2.2. In

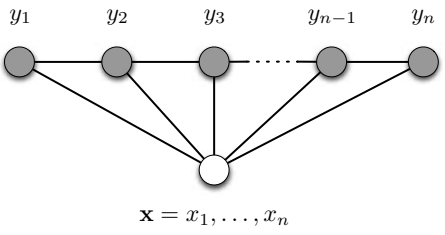


Fig. 3. Graphical model of a linear-chain CRF

Figure 2.2, the hidden variables are shown in shaded circles, whereas observed variables are shown in white. From the Markov assumption, it follows that each hidden variable only depends on its neighboring nodes.

Log-linear models have been used as potential functions in CRFs for their convenience in computation. Given an input observation $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathcal{X}^n$, CRF computes the probability $p(\boldsymbol{y}|\boldsymbol{x})$ of a possible output $\boldsymbol{y} = (y_1, \ldots, y_n) \in \mathcal{Y}^n$. The general formulation of the linear-chain CRF is,

$$p(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m} \lambda_i f_i(y_{j-1}, y_j, x_j)\right). \tag{1}$$

Here, $f_i(y_{j-1}, y_j, \boldsymbol{x})$ is a binary-valued feature that returns the value 1 if the corresponding feature is fired when moving from the hidden state y_{j-1} to y_j, after observing x. For example, we can define a feature that encodes the property *previous position is not an* EDU *boundary and the current token is a comma.* λ_i is the weight associated with feature f_i. The linear sum of weights over features is taken inside the exponential and the normalization constant Z (also known as the partition function) is set such that the sum of probabilities over all possible label sequences, $\mathcal{Y}(\boldsymbol{x})$, for $\boldsymbol{x}$, equals one. The expression of the normalization constant is,

$$Z(\boldsymbol{x}) = \sum_{\boldsymbol{y} \in \mathcal{Y}(x)} \exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m} \lambda_i f_i(y_{j-1}, y_j, x_j)\right). \tag{2}$$

We can further simplify the notation by introducing the global feature vector, $\boldsymbol{F}(\boldsymbol{y}, \boldsymbol{x}) = \{F_1(\boldsymbol{y}, \boldsymbol{x}), \ldots, F_m(\boldsymbol{y}, \boldsymbol{x})\}$, where $F_i(\boldsymbol{y}, \boldsymbol{x}) = \sum_{j=1}^{N} f_i(y_{j-1}, y_j, x_j)$. Moreover, we define the weight vector $\Lambda = \{\lambda_1, \ldots, \lambda_m\}$. Then, Equation 1 can be written as,

$$p(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \exp(\Lambda \cdot \boldsymbol{F}(\boldsymbol{y}, \boldsymbol{x})). \tag{3}$$

Here, $\cdot$ denotes the inner-product between the global feature vector and weight vector.

Given a training dataset, $T = \{(\boldsymbol{x_j}, \boldsymbol{y_j})\}_{j=1}^{N}$, the weights λ_i are computed such that the log-likelihood L_Λ of T is maximized. Log-likelihood over the training dataset can be written as follows,

$$\begin{aligned} L_\Lambda &= \sum_{j=1}^{N} \log(P(y_j|x_j)) \\ &= \sum_{j=1}^{N} \left(\sum_{\boldsymbol{y} \in \mathfrak{Y}(x_j)} \exp(\Lambda \cdot (\boldsymbol{F}(y_j, x_j) - \boldsymbol{F}(y, x_j))) \right) \\ &= \sum_{j=1}^{N} \left(\Lambda \cdot \boldsymbol{F}(y_j, x_j) - \log(Z_{x_j}) \right). \end{aligned}$$

Therefore, to maximize the log-likelihood, we must maximize the difference between the inner-products of the correct labelling $\Lambda \cdot \boldsymbol{F}(y_j, x_j)$ and all other candidate labellings $\Lambda \cdot \boldsymbol{F}(y, x_j)$ for $y \in \mathfrak{Y}(x_j)$. In practice, to avoid overfitting the weights to the training data, the vector Λ is regularized. Two popular choices for vector regularization are L_1 and L_2 regularizations. In general, the $\mathrm{L_k}(\boldsymbol{x})$ regularization of an n-dimensional vector $\boldsymbol{x}$ is defined as,

$$\mathrm{L_k}(\boldsymbol{x}) = \sqrt[k]{\sum_{i=1}^{n} {x_i}^k}. \tag{4}$$

The final optimization function $\mathrm{H}(\Lambda)$ with regularization can be written as,

$$\mathrm{H}(\Lambda) = L_\Lambda - \sigma L_k. \tag{5}$$

Here, σ is a regularization coefficient that determines the overall effect of regularization towards training. This optimization problem can be efficiently solved by using gradient descent algorithms. We used CRFsuite [14] with L_1 regularization, which implements the orthant-wise limited memory quasi-Newton (OWL-QN) method. The regularization coefficient is set to its default value of 1.

Because CRFs are discriminative models, they capture many correlated features of the input. This property of CRFs is particularly useful, because we can define as many features as we like, irrespective of whether those features are mutually independent or not. If a particular feature is not useful, then it will be assigned a lower weight and will effectively get removed in the final trained model. In particular, L_1 regularization yields sparser models compared to L_2, thereby removing many useless features automatically [15].

Once trained, the CRF can then be used to find the optimal labeling sequence $\hat{\boldsymbol{y}}$ for a given input $\boldsymbol{x}$ as,

$$\hat{\boldsymbol{y}} = \underset{\boldsymbol{y} \in \mathfrak{Y}(\boldsymbol{x})}{\arg\max}\, p(\boldsymbol{y}|\boldsymbol{x}). \tag{6}$$

2.3 Application to Discourse Segmentation

In the case of discourse segmentation, the problem is to assign each word in the input text an observation category $c \in \{C, B\}$, where B denotes a *beginning* of EDU, and C a *continuation* of EDU. For example, given the snippet from Figure 1:

```
The posters were printed on paper pre-signed by Mr. Dali ,
the attorneys said .
```

The output sequence is the following:

```
B C C C C C B C C C C B C C C
```

2.4 Features

We use a combination of syntactic and lexical features: words, POS tags, lexical heads. In particular, we use the lexico-syntactic features of [7], which were found to constitute a good indication of the presence of EDU boundaries.

Figure 2.4 shows part of the sentence's syntax tree, in which lexical heads have also been indicated, using projection rules from [16]. For a word w, we look at its highest ancestor in the parse tree with a lexical head equal to w, and with a right-sibling. We call this highest-ancestor node N_w, N_p its parent, and N_r its right-sibling. For instance, when following this process for the word 'paper', we get $N_w = \text{NP(paper)}$, $N_p = \text{NP(paper)}$, $N_r = \text{VP(pre-signed)}$.

We define as *contextual features at position i* in the text, the set composed of the word w_i, its POS, as well as the POS and lexical heads of N_{w_i}, N_{p_i}, and N_{r_i}. In the experiments of Section 3, unless stated otherwise, the features for position i in the text are created by concatenating the contextual features at positions

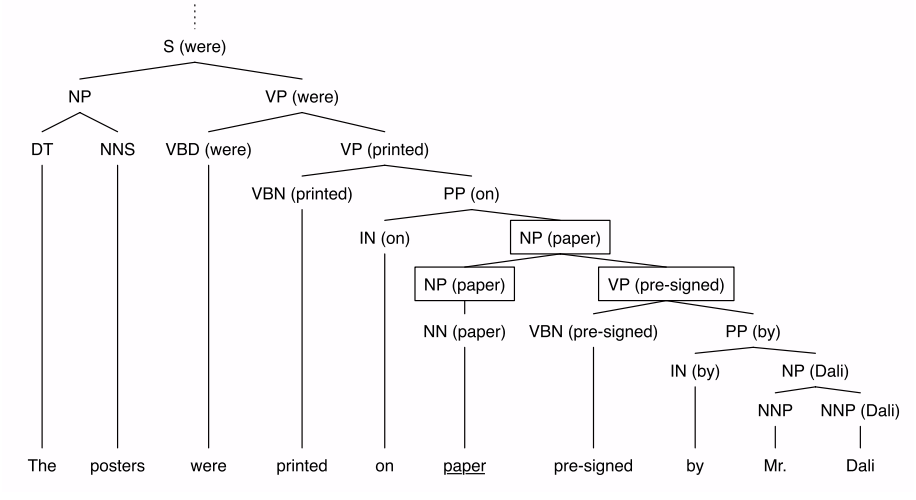


Fig. 4. Partial lexicalized syntax tree

$i-2$, $i-1$, and i. These elements are then encoded into feature templates. For instance, the template encoding the property *the current word is 'paper'* is,

$$g(\boldsymbol{x}, j) = \begin{cases} 1 \text{ if } x_j = \texttt{paper} \\ 0 \text{ otherwise} \end{cases} .$$

These templates are used as a basis to generate the CRF feature functions defined in Section 2.2. Our working corpus, the RST-DT, contains 385 texts from the Wall Street Journal (347 for training, 38 as a test subset). After feature extraction, we obtain 177,633 training vectors and 21,667 test vectors.

3 Experiments

We first implement our CRF-based segmenter, referred to as 'CRFSeg' in the rest of the paper. Three versions of the segmenter are created, using parse trees from different sources. First, we use trees from the Penn Treebank [17], which are gold-standard, human-annotated syntax trees. Then, we use trees generated by Charniak's syntax parser [18]. Last, we use trees generated by the Stanford parser [19].

Evaluation is done on the test subset of the RST-DT. We use the metric commonly agreed by most authors ([7], [12]), i.e., we only evaluate intra-sentence boundaries. Thus, the score is not artificially boosted by including obvious end-of-sentence or start-of-sentence boundaries. For instance, the sentence of Figure 1 is made of three EDUs, but we only take into account two boundaries. The performance of our segmenter is reported in Table 1.

As expected, using gold-standard parse trees from the Penn Treebank yields the best results, with an F-score of 0.953. Using syntax parsers instead produces slightly lower scores, particularly in terms of recall for the B label. However, the macro-average scores are almost identical for both software, with an F-score

Table 1. Detailed performance of CRFSeg

Trees	Label	Precision	Recall	F-score
Penn	B	0.927	0.897	0.912
	C	0.992	0.995	0.993
	Macro-average	0.960	0.946	0.953
Charniak	B	0.915	0.876	0.895
	C	0.990	0.993	0.992
	Macro-average	0.952	0.935	0.943
Stanford	B	0.910	0.872	0.890
	C	0.990	0.993	0.991
	Macro-average	0.950	0.932	0.941

Table 2. Performance comparison with other segmenters

System	Trees	Precision	Recall	F-score
SPADE	Penn	0.841	0.854	0.847
NNDS	Penn	0.855	0.866	0.860
CRFSeg	Penn	**0.960**	**0.946**	**0.953**
SPADE	Charniak	0.835	0.827	0.831
NNDS	Charniak	0.839	0.848	0.844
CRFSeg	Charniak	**0.952**	**0.935**	**0.943**
CRFSeg	Stanford	0.950	0.932	0.941
Human agreement	–	0.985	0.982	0.983

of 0.943 for Charniak, and 0.941 for Stanford. This suggests that the proposed method is not constrained by the choice of a specific parser.

Next, we compare the performance of our segmenter to other works. Results are presented in Table 2. NNDS indicates the Neural-Networks Discourse Segmenter [10] ; SPADE is the system described in [7]. Here, CRFSeg significantly outperforms other discourse segmenters. When using gold-standard trees, SPADE and NNDS yield respectively F-scores of 0.847 and 0.860, versus 0.953 for CRFSeg. The measure of the human annotator's agreement for the segmentation task has been calculated in [7], with a F-score of 0.983. Using CRFSeg with perfect parse trees, we reach 96.9% of this score, while we reach 95.7% of this score when using the Stanford parser.

We chose not to include in Table 2 the rule-based segmenters of [11] and [12], for several reasons. First, [11] report their results using a 'softer' metric, in which end-of-sentence boundaries are taken into account. The authors used Penn Treebank parse trees, and after evaluation on 8 texts of the RST-DT, obtain a precision of 0.814 and recall of 0.792. With the same parse trees and metric, but using the 38 texts from the standard test subset, CRFSeg obtains a precision of 0.973 and recall of 0.969. Finally, the results of [12] cannot be directly compared to ours, as different segmentation guidelines were used. The authors report there a precision of 0.890 and recall of 0.860 when using Charniak parse trees, a precision of 0.820 and recall of 0.860 when using Stanford trees. Moreover, this score is measured on 3 texts of the RST-DT only, which makes comparison even more risky.

3.1 Comparison to a Segmenter Based on Support Vector Machines

To compare the sequential model of discourse segmentation with a classification model, we implement a discourse segmenter using Support Vector Machines (SVM) [8]. SVMs have reported state-of-the-art performances in a wide range of tasks in NLP. We employ the same training data and features we previously used with CRFs, and [20] is used for the implementation. We select the RBF kernel,

Table 3. Comparison of performance, using contextual features at various positions

	CRFSeg			SVMSeg		
Positions	**Precision**	**Recall**	**F-score**	**Precision**	**Recall**	**F-score**
$(-3,-2,-1,0)$	0.960	0.949	0.954	0.960	0.952	**0.956**
$(-2,-1,0)$	0.960	0.946	0.953	0.965	0.954	**0.959**
$(-1,0,1)$	0.941	0.928	0.934	0.943	0.932	**0.938**
$(0,1,2)$	0.846	0.834	**0.840**	0.831	0.807	0.819
$(-1,0)$	0.938	0.929	0.934	0.940	0.934	**0.937**
$(0,1)$	0.843	0.830	**0.836**	0.834	0.804	0.819
(0)	0.845	0.827	**0.836**	0.821	0.801	0.811

and optimal parameters are found using grid search with 5-fold cross-validation. We dub this segmenter 'SVMSeg'.

In order to see how SVMSeg and CRFSeg perform under varied settings, we run several experiments, using contextual features from various positions. For instance, given the vector of relative positions $(-2,-1,0)$, we build the feature vector for text position i as the concatenation of contextual features from positions $i-2$, $i-1$, and i. Results of the experiments with perfect parse trees are shown in Table 3.

The first striking observation is that, when using contextual features located before the current position, CRFSeg and SVMSeg perform similarly, with a slightly higher score for SVMSeg. For example, using positions $(-2,-1,0)$, CRFSeg and SVMSeg yield respectively F-scores of 0.953 versus 0.959, which is not a statistically significant difference. In this case, there is no clear benefit of the sequential model over a classification approach. It is also interesting to note that the cases $(-3,-2,-1,0)$ and $(-2,-1,0)$ produce identical results, which suggests that context that appears at a distance farther than two words from the current position is not useful for segmentation.

When using only the context of the current position, (0), CRFSeg outperforms SVMSeg (respective F-score of 0.836 versus 0.811). Here the CRF model has the upper-hand, as it is able to remember its past input data and decisions. However, including contextual features for positions ahead does not improve the score, which confirms that segmentation does not require the knowledge of future words and contexts – excepted for the interaction with the immediate next word, which is already encoded in our features, c.f. Section 2.4.

Finally, we run a head-to-head error comparison of the two models. In this experiment, we use the results from case $(-2,-1,0)$. In order to account for all errors, the metric is changed so that we consider all EDU boundaries without restriction. Results are shown in Table 4.

We measure an error rate of 10^{-2} for CRFSeg, while SVMSeg has an error rate of $9.4 \cdot 10^{-3}$. However, 30% of the errors made by CRFSeg happen on cases where SVMSeg is correct, and reciprocally, 20% of errors made by SVMSeg occur on cases where CRFSeg is correct. A possible extension could then be to

Table 4. Comparison of errors between CRFSeg and SVMSeg

		SVMSeg		
		OK	**¬OK**	**Total**
CRFSeg	**OK**	21393	41	21434
	¬OK	70	163	233
	Total	21463	204	21667

combine both systems, and create a hybrid segmenter with a lower error rate. For instance, it is possible to measure, for each input data, the confidence of the two models, and use only the result of the model with the highest confidence. In this case, the expected error rate of the hybrid system is $7.5 \cdot 10^{-3}$ (163/21667).

4 Conclusion

We have presented a sequential model of discourse segmentation, based on Conditional Random Fields. The proposed model finds the globally optimum sequence of discourse boundaries, which makes for one of the most efficient supervised discourse segmentation methods we are aware of. Using standard automatic syntax parsers, our system reaches 96% of the human performance level. We also found that this approach performs comparably to a SVM-based discourse segmenter using contextual features. We suggested to build a hybrid system combining both models, in order to further reduce the number of incorrect segmentation decisions. In this case, we expect an error rate of $7.5 \cdot 10^{-3}$. These results validate that our segmenter is usable in a real-time discourse parsing system, in which the segmentation step is decisive for the rest of the process.

In the continuation of this work, we are currently exploring the benefits of sequential approaches for discourse relation labeling and tree construction.

References

1. Marcu, D.: The Theory and Practice of Discourse Parsing and Summarization. MIT Press, Cambridge (2000)
2. Chai, J.Y., Jin, R.: Discourse structure for context question answering. In: Harabagiu, S., Lacatusu, F. (eds.) HLT-NAACL 2004: Workshop on Pragmatics of Question Answering, Boston, Massachusetts, USA, pp. 23–30. Association for Computational Linguistics (2004)
3. Hernault, H., Piwek, P., Prendinger, H., Ishizuka, M.: Generating dialogues for virtual agents using nested textual coherence relations. In: Prendinger, H., Lester, J.C., Ishizuka, M. (eds.) IVA 2008. LNCS (LNAI), vol. 5208, pp. 139–145. Springer, Heidelberg (2008)
4. Georg, G., Hernault, H., Cavazza, M., Prendinger, H., Ishizuka, M.: From rhetorical structures to document structure: shallow pragmatic analysis for document engineering. In: DocEng 2009, pp. 185–192. ACM, New York (2009)

5. Mann, W.C., Thompson, S.A.: Rhetorical structure theory: Toward a functional theory of text organization. Text 8, 243–281 (1988)
6. du Verle, D., Prendinger, H.: A novel discourse parser based on support vector machine classification. In: ACL 2009, Suntec, Singapore, pp. 665–673. Association for Computational Linguistics (2009)
7. Soricut, R., Marcu, D.: Sentence level discourse parsing using syntactic and lexical information. In: NAACL 2003, Morristown, NJ, USA, pp. 149–156. Association for Computational Linguistics (2003)
8. Vapnik, V.N.: The nature of statistical learning theory. Springer, New York (1995)
9. Carlson, L., Marcu, D., Okurowski, M.E.: Rst discourse treebank (2002)
10. Subba, R., Di Eugenio, B.: Automatic discourse segmentation using neural networks. In: Proceedings of the 11th Workshop on the Semantics and Pragmatics of Dialogue, Trento, Italy, pp. 189–190 (2007)
11. Le, H.T., Abeysinghe, G., Huyck, C.: Automated discourse segmentation by syntactic information and cue phrases. In: AIA 2004, Innsbruck, Austria (2004)
12. Tofiloski, M., Brooke, J., Taboada, M.: A syntactic and lexical-based discourse segmenter. In: ACL 2009, Suntec, Singapore, pp. 77–80. Association for Computational Linguistics (2009)
13. Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: ICML 2001, pp. 282–289. Morgan Kaufmann Publishers Inc., San Francisco (2001)
14. Okazaki, N.: Crfsuite: a fast implementation of conditional random fields, crfs (2007)
15. Ng, A.Y.: Feature selection, l1 vs. l2 regularization, and rotational invariance. In: ICML 2004, p. 78. ACM, New York (2004)
16. Magerman, D.M.: Statistical decision-tree models for parsing. In: ACL 1995, Morristown, NJ, USA, pp. 276–283. Association for Computational Linguistics (1995)
17. Marcus, M.P., Marcinkiewicz, M.A., Santorini, B.: Building a large annotated corpus of english: the penn treebank. Comput. Linguist. 19, 313–330 (1993)
18. Charniak, E.: A maximum-entropy-inspired parser. In: NAACL 2000, pp. 132–139. Morgan Kaufmann Publishers Inc., San Francisco (2000)
19. Klein, D., Manning, C.D.: Fast exact inference with a factored model for natural language parsing. In: Advances in Neural Information Processing Systems, vol. 15. MIT Press, Cambridge (2003)
20. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines (2001), http://www.csie.ntu.edu.tw/~cjlin/libsvm

Towards Automatic Detection and Tracking of Topic Change

Florian Holz and Sven Teresniak

NLP Group, Department of Computer Science, University of Leipzig
{holz,teresniak}@informatik.uni-leipzig.de

Abstract. We present an approach for automatic detection of topic change. Our approach is based on the analysis of statistical features of topics in time-sliced corpora and their dynamics over time. Processing large amounts of time-annotated news text, we identify new facets regarding a stream of topics consisting of latest news of public interest. Adaptable as an addition to the well known task of topic detection and tracking we aim to boil down a daily news stream to its novelty. For that we examine the contextual shift of the concepts over time slices. To quantify the amount of change, we adopt the volatility measure from econometrics and propose a new algorithm for frequency-independent detection of topic drift and change of meaning. The proposed measure does not rely on plain word frequency but the mixture of the co-occurrences of words. So, the analysis is highly independent of the absolute word frequencies and works over the whole frequency spectrum, especially also well for low-frequent words. Aggregating the computed time-related data of the terms allows to build overview illustrations of the most evolving terms for a whole time span.

1 Introduction

Large collections of digital diachronic text such as the New York Times corpus and other newspaper or journal archives in many ways contain temporal information related to events, stories and topics. To detect the appearance of new topics and tracking the reappearance and evolution of them is the goal of topic detection and tracking [2,1]. For a collection of documents, relevant terms need to be identified and related to a particular time-span, or known events, and vice versa, time-spans need to be related to relevant terms. To identify relevant and new terms in a stream of text (within a predefined period of time), three main approaches have been followed. [7,8,6] measure the relevance of terms using multiple document models and thresholds based on a tf/idf comparison of text stream segments. [5] introduces the burstiness of terms during certain periods of time as an additional dimension for topic detection, and models the temporal extension of relevant terms using a weighted finite state automaton. [10] use co-occurrence patterns and their local distribution in time to detect topics over time. By their approach, every topic is represented by a co-occurrence set of terms representative of a certain period of time. Assuming topics and the terms representing them to be constant over time, topics can efficiently be related to times.

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 327–339, 2010.

However, topics not only depict events in time, they also mirror an author's, or society's, view on the events described. And this view can change over time. In language, the relevance of things happening is constantly rated and evaluated. In our view, therefore, topics represent a conceptualization of events and stories that is not statically related to a certain period of time, but can itself change over time. Tracking these changes of topics over time is highly useful for monitoring changes of public opinion and preferences as well as tracing historical developments.

In what follows, we shall argue that

1. changing topics can be detected by looking at their change of meaning,
2. changing topics are interesting, i. e. they generally represent topics that for some period of time are "hotly discussed", and
3. tracking the change of topics over time reveals interesting insights into a society's conceptualization of preferences and values.

If we consider the disclosure of new aspects regarding an already established and identified topic, the extraction of the meaning's change can be used to distinguish between real novelty and already known facts or recurring events.

While the public attention to a topic determines the topic's presence in media coverage, the novelty regarding this topic is somewhat independent from the amount of coverage of a story: even without worldshaking new facts, a topic can be important to society and therefore on the agenda of an editorial departement. From a text mining perspective which takes the past news as given facts and aims to extract unknown and novel aspects and developments this reporting is to some degree redundant.[1] With the ability to discriminate between *novel* news on the one hand and news just referencing the *recent* (but not necessarily new) knowledge about a certain topic on the other hand, we are able to identify the novelty bearing parts in news streams.

In addition to term frequency, we consider a term's global context (see below) as a second dimension for analyzing its relevance and temporal extension and argue that the global context of a term may be taken to represent its meaning(s). Changes over time in the global context of a term indicate a change of meaning. The rate of change is indicative of how much the "opinion stakeholders" agree on the meaning of a term. Fixing the meaning of a term can thus be compared to fixing the price of a stock. Likewise the analysis of the volatility of a term's global contexts can be employed to detect topics and their change over time. We first explain the basic notions and assumptions of our approach and then present first experimental results.

2 Motivation for Our Method

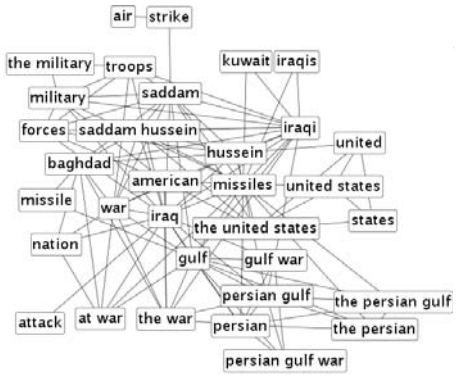
Following [4], we take a term to mean the inflected type of a word, where the notion of a word is taken to mean an equivalence class of inflected forms of a

[1] For example consider the global warming. While for the last few years, this topic became more and more important, no fundamental new aspects were discovered and thus no novel facts are added to the discussion.

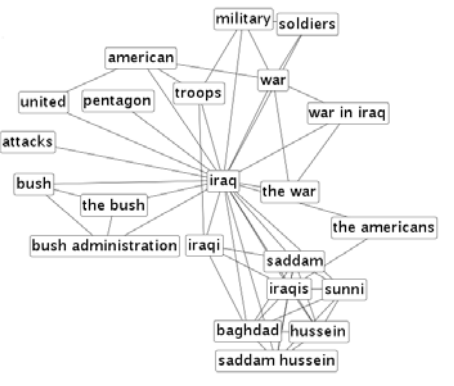
base form. Likewise we take the notion of a topic to mean an equivalence class of words describing an event (as computed by the global context of the topic's name), and the notion of a concept to mean an equivalence class of semantically

Table 1. The 30 most significant co-occurrences and their significance values (cut to 3 digits) in the global context of "abu ghraib" on May 10, 2004

prisoners 0.346, abuse 0.346, secretary 0.259, abu ghraib prison 0.259, iraqi 0.247, rumsfeld 0.221, military 0.218, prison 0.218, bush 0.210. prisoner 0.200, photographs 0.183, donald 0.183, secretary of defense 0.174, prisons 0.174, photos 0.174, the scandal 0.174, interrogation 0.163, naked 0.163, mistreatment 0.163, under 0.162, soldier 0.154, saddam 0.154, armed 0.154, defense 0.143, the bush 0.140, senate 0.140, videos 0.130, torture 0.130, arab 0.130, captured 0.130



(a) March 21, 2003



(b) December 14, 2003

(c) Mai 6, 2004

Fig. 1. The co-occurrance graphs of "iraq" depict the changes in the global context of the term for three days in (a), (b) 2003 and (c) 2004

1. Build a corpus where all time slices are joined together.
2. Compute for this overall corpus all significant co-occurrences $C(t)$ for every term t.
3. Compute all significant co-occurrences $C_i(t)$ for every time slice i for every term t.
4. For every co-occurrence term $c_{t,j} \in C(t)$ compute the series of ranks $\text{rank}_{c_{t,j}}(i)$ over all time slices i. This represents the ranks of $c_{t,j}$ in the different global contexts of t for every time slice i.
5. Compute the coefficient of variation of the rank series $\text{CV}(\text{rank}_{c_{t,j}}(i))$ for every co-occurrence term in $c_{t,j} \in C(t)$.
6. Compute the average of the coefficients of variation of all co-occurences terms $C(t)$ to obtain the volatility of term t

$$\begin{aligned}\text{Vol}(t) &= \underset{j}{\text{avg}} \left(\underset{i}{\text{CV}} \left(\text{rank}_{c_{t,j}}(i) \right) \right) \\ &= \frac{1}{|C(t)|} \sum_{j} \underset{i}{\text{CV}} \left(\text{rank}_{c_{t,j}}(i) \right) \ .\end{aligned}$$

Fig. 2. Computing the volatility

related words. The global context of a topic's name is the set of all its statistically significant co-occurrences within a corpus. We compute a term's set of co-occurrences on the basis of the term's joint appearance with its co-occurring terms within a predefined text window taking an appropriate measure for statistically significant co-occurrence. The significance values are computed using the log-likelihood measure following [3] and afterwards normalized according to the actual corpus size. These significance values only serve for sorting the co-occurrence terms; their absolute values are not considered at all. The position of a term in this sorted list is called the term's rank.

Table 1 exemplifies the global context computed for the term "abu ghraib" based on the New York Times corpus of May 10, 2004. The numbers in parenthesis behind a term indicate its statistical significance (normalized to the corpus size and multiplied by 10^6), which are used to rank the co-occurring terms (cf. Fig. 2).

The global context can also be displayed as a graph which contains the term and its context terms as nodes where the edges have a weight each according to the significance value of the joint appearance of the terms. Figures 1(a)–(c) illustrate the change of co-occurrences and thus the change of the global context of the word "iraq" for three different days in 2003 and 2004 based on the New York Times corpus. These Graphs show how a changing media coverage is affecting the co-occurrences of a term.

3 Method

The basis of our analysis is a set of time slice corpora. These are corpora belonging to a certain period of time, e. g. all newspaper articles of the same day.

Table 2. Volatility and global context

No. co-occurrences	Volatilität: low	Volatilität: high
constant	high-frequent terms: - stop words - "static concepts"	periodic or arbitrary concepts, e. g. "Monday", "city center" etc.
differing	rising or declining concepts, e. g. "globalization" etc.	highly dynamic notions: - low-frequent ("weak signals") - high-frequent

The assessment of change of meaning of a term is done by comparing the term's global contexts of the different time slice corpora.

The measure of the change of meaning is *volatility*. It is derived from the widely used risk measure in econometrics and finance[2], and based on the sequence of the significant co-occurrences in the global context sorted according to their significance values and measures the change of the sequences over different time slices. This is because the change of meaning of a certain term leads to a change of the usage of this term together with other terms and therefore to a (maybe slight) change of its co-occurrences and their significance values in the time-slice-specific global context of the term. The exact algorithm to obtain the volatility of a certain term is shown in Fig. 2.

In order to reduce the time complexity of our algorithm we only take the overall most important co-occurrences into account. This is done by computing the global contexts of the terms based on an overall corpus which is the aggregation of all time slice corpora. In the case of the used New York Times corpus this means a comprehension of about 7 500 days which are about 20 years. Using an overall significance threshold only the more significant terms are taken into account during the comparison of the time-slice-specific global contexts. This set of relevant co-occuring terms for a term t is named $C(t)$ in Fig. 2. Besides providing evidence for meaningful filtering the overall corpus is not used in the computation of the volatility. A co-occurring term is significant if the accoring co-occurrence, i. e. the pair of the original term and the co-occurring term is significant. Co-occurrences are taken as statistically significant if they a) occur at least two times in the corpus and b) their significance value computed using the log-likelihood measure exceeds a threshold. In our experiments the threshold was set that half the co-occurrences occurring at least two times passed it. Based on language statistics this means very careful filtering.

Concerning the relation between the volatility and the global context of a term the following picture can be sketched (cf. Tab. 2). One can expect that the volatility of terms like "Monday" is quite high because weekdays (and other

[2] But it is calculated differently and not based on widely used gain/loss measures. For an overview over miscellaneous approaches to volatility see [9].

periodic re-occurring time references) are highly ambiguous as the specification which precise day is meant is lacking. Analogously this is to be assumed for ambiguous place identifiers without specification like “city center” and “town hall” as we do no identification of the concrete referenced entity or any other semantic pre-processing.

4 Experiments

In what follows, we present results of experiments that were carried out on the basis of data based on the New York Times Annotated Corpus (NYT)[3]. Table 3 lists some general characteristics of this corpus. First tests were perfomed for German on the corpus of the project *Deutscher Wortschatz*[4] and showed comparable results. We aim to show that our method in fact works to detect topics that were “hotly discussed” during some period of time, giving also an indication, why that has been so. For performance reasons, we only took the 50 000 most frequent terms out of the 20-year NYT corpus into account and filtered this list as follows:

- remove stop words,
- remove all terms with a frequency rank higher than 50 000 (this is equivalent to less than 850 occurrences in 20 years NYT),
- include all multi word units which are wikipedia lemmata and of a rank of at most 50 000
- remove digits, numbers and so on.

The resulting term list contains “people” as its most frequent word and “benes” (a name) as its least frequent word. As described in Sect. 3, for all of these words, its most significant co-occurrences in the overall corpus of the whole 20 years were computed. Thereafter the volatility for every term was computed.

Figure 3 shows the development of the volatility of “abu ghaib” from January 2004 to December 2006. The volatility was computed per day with a window of 30 days, i. e. for the volatility for a certain day the last 30 days before were taken into account (cf. Fig. 2). The daily frequency of “abu ghraib” is also shown in Fig. 3 as a 30-day average over the last 30 days, too. The clearly outstanding peaks of the volatility are easily connectable to certain events and their related media coverage. The first peak beginning in May 2004 is caused by the initial discussion about the torture pictures and videos taken in the prison in Abu Ghraib. This is also clearly affecting the co-occurrences of “abu ghraib” as shown in Tab. 1.

Concerning the next time span, the frequency is declining but fluctuating. These fluctuations aren’t related to a change of the topic what can be seen in the uncorrelated peaks of the volatility. The volatility peak in the end of May 2005 is caused by a widening of usage of the term “abu ghraib”, e. g.

[3] http://www.ldc.upenn.edu/

[4] http://wortschatz.uni-leipzig.de

Table 3. Characteristics of the used corpus NYT

language	english
time span	Jan 87 – Jun 07
no. time slices	7 475
no. document	1.65 mil.
no. tokens	1 200 mil.
no. types	3.6 mil.
no. sig. co-occurrences	29 500 mil.
size (plain text)	5.7 GB

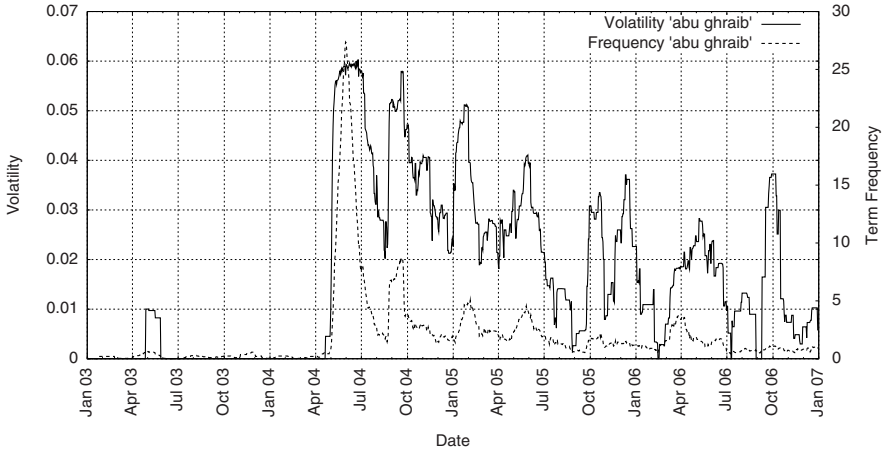


Fig. 3. 30-day volatility and frequency of "abu ghraib" from 2003 to 2006 based on the NYT corpus

the NYT reports about U. S. interrogators accused of misconducts and hold responsible for dead detainees in afghanistan as well as a discussion which actions are appropriate during a military interrogation and what methods are regarded as torture. Further, now "abu ghraib" is also used as the name of a district in the west of baghdad – where a huge military operation took place – which our algorithm couldn't differentiate from "abu ghraib" as the catchphrase for the abusive military behaviour at Abu Ghraib prison.

The new aspect and topic shift does not lead to an extended coverage in the New York Times but is measureable as a change of context. The peak in November and December 2005 is related to an exhibition which also was held in New York. There pictures from Abu Ghraib have been exhibited together with others from the Weimarer Republic and World War II. This new aspect and its reporting is cleary affecting the global context of "abu ghraib". Table 4 shows

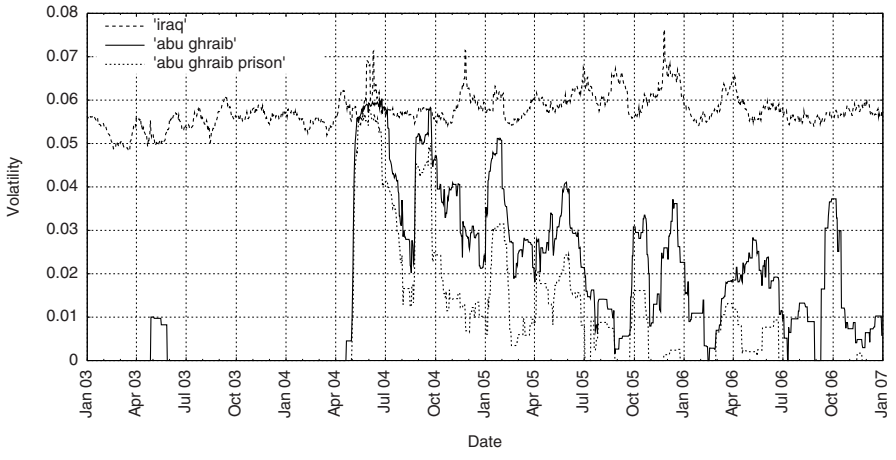


Fig. 4. Joint representation of the 30-day volatility of "abu ghraib", "abu ghraib prison" and "iraq" from 2003 to 2006 based on the NYT corpus

Table 4. The 30 most significant co-occurrences in the global context of "abu ghraib" on November 20, 2005

disasters, hook, grosz, international center, finalized, weighty, inkling, complement, partnerships, guggenheim museum, collaborative, the big city, easel, reaped, hudson river museum, blockbuster, enlarging, goya, weimar, art museums, eras, inconvenient, negatives, golub, poughkeepsie, griswold, big city, impressionist, staging, neuberger

this for the November, 20, when the reporting about the exhibition started. The event also does not cause a more frequent usage of "abu ghraib" in the New York Times, but is nevertheless detectable by the related change of context.

However, another interesting fact is the comparision of the volatility graphs of "abu ghraib", "abu ghraib prison", and "iraq" as shown in Fig. 4. Between January 2003 and May 2004 the incidents at "abu ghraib" are not an issue, whereas the invasion of Iraq and the ongoing discussions around it are clearly visible in the graph as a relatively and constantly high-volatile topic.

At the beginning of news coverage related to the abuse at the Abu Ghraib prison the peaks of "abu ghraib" and "abu ghraib prison" are strongly correlated. But at the end of 2005 the ongoing reporting was strong or long enough to establish "abu ghraib" as a synonym to the complete affair around the abuse in Abu Ghraib prison, whereupon "abu ghraib prison" is no longer a term of general interest. Now, in the western public perception, "abu ghraib" no longer stands mainly for a city with about 1 mil. inhabitants, but for the crisis in the prison of this city. From 2006 on, the torture scandal in Abu Ghraib prison – like Guantanamo Bay – is used more frequently in a symbolic way and exploited by many people for their individual needs (e. g. by Mahmoud Ahmadinejad for

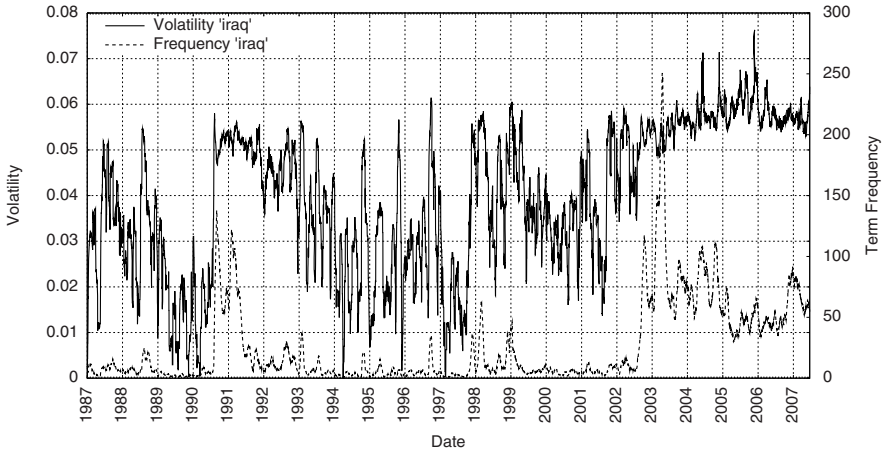


Fig. 5. 30-day volatility and frequency of "iraq" from 1987 to 2006 based on the NYT corpus

propaganda reasons to appealing to Iran's conservative base or by opponents of war to connect the war on terrorism in Iraq with cruelty, both in 2006). Once established as a symbol, the Abu Ghraib crisis is stressed controversely in many contexts and thus remains high-volatile at least until November of 2006, even though the absolute frequency of "abu ghraib" is quiet low (cf. Fig. 3).

Figure. 5 shows the development of the volatility and the frequency of "iraq" from 1987 to 2007. It clearly can be seen that volatility and frequency do not correlate. New aspects like the Gulf War in 1990 or the discussion after Sep 11, 2001 lead to an increased volatility. An increased frequency alone – like at the beginning of the invasion of Iraq in 2003 – does not affect the volatility as at this time all reported arguments have already been very well known.

5 Generating Overviews

Based on the volatility values over time, e. g. per-day values as shown in Sect. 4, it is possible to generate an overview over the collection over the whole time span or a section. This is based on the fact, that for always highly volatile terms like stop words or concepts like "Monday" the volatility remains high all the time and its variance is comparably low in comparison to less frequent but thematically evolving terms which are expected to have a low volatility in general but with peaks at moments where they are hotly discussed or new aspects show up. For examples described in detail see Sect. 4. Therefore we computed for every term the variance of the series of volatility values to get one value for each term indicating how much the topic evolved over the considered time span.

The visual overview is a 2D plot where every term's position is given by the term's absolute frequency and the term's volatility. Thus the overview depicts the relation between how present a term was in the shown time span and how

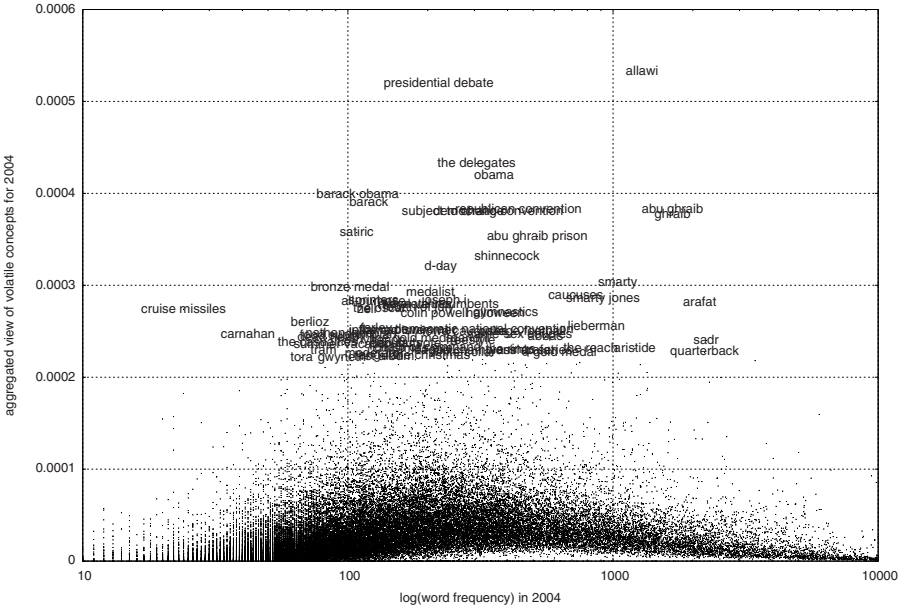


Fig. 6. Variance of volatility according to term frequency for 2004 (section, more- and less-frequent terms are not displayed)

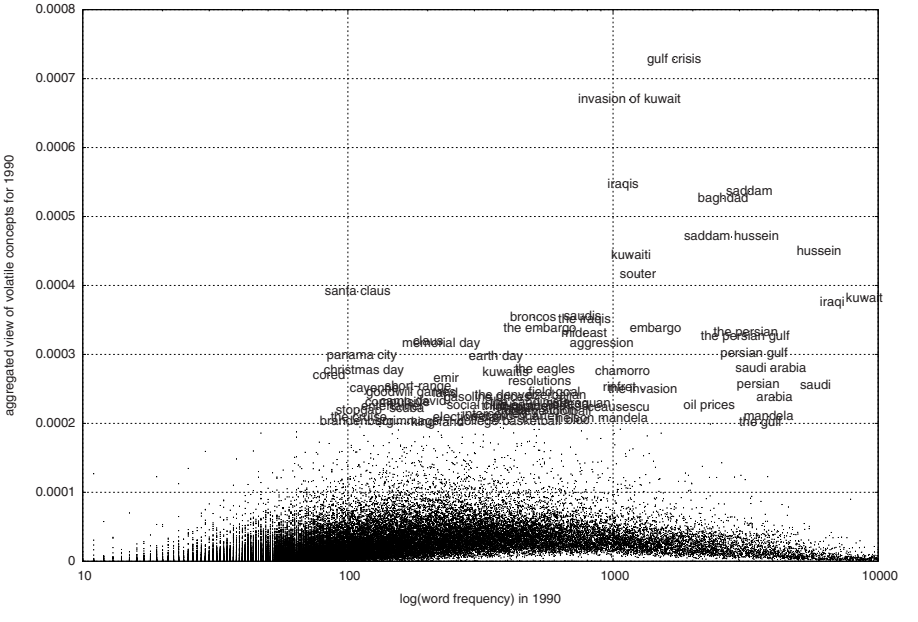


Fig. 7. Variance of volatility according to term frequency for 1990 (section, more- and less-frequent terms are not displayed)

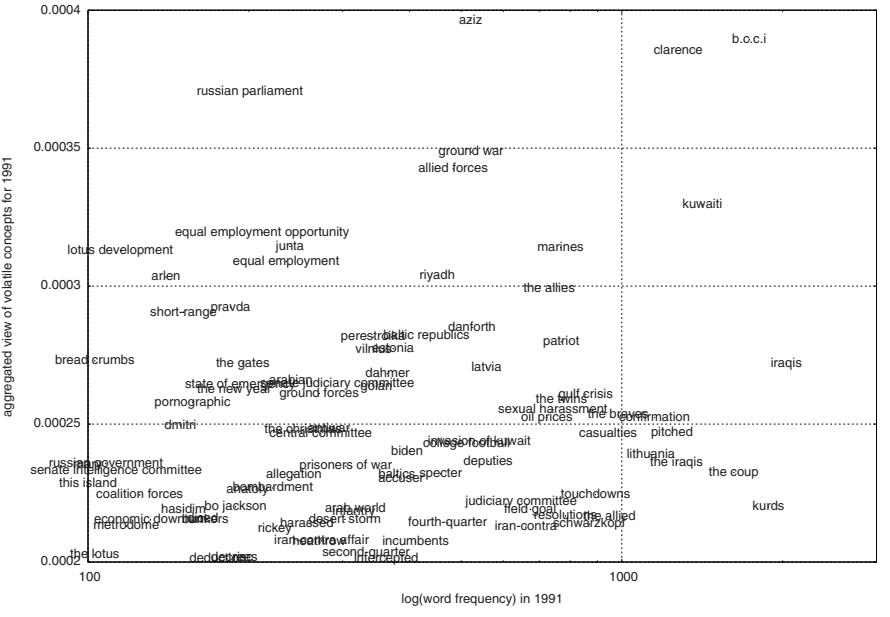


Fig. 8. Variance of volatility according to term frequency for 1991 (zoomed in for readability – more- and less-frequent terms as well as high- and low-volatile terms are not displayed)

much the related topic evolved over it. The overview provides a simple and intuitive aggregation of the document collection. Figure 6 shows such an overview computed for all articles of the New York Times corpus in 2004. For a clear arrangement in the figure only for the 75 most variable and thus evolving terms the dots are replaced by the term itself. There, for every term the variance of its volatility values over 2004 is plotted according to the term's frequency in the New York Times corpus in 2004. So, the high-frequent terms are on the right side, the low-frequent ones on the left. The x-axis displays the frequency logarithmically. Therefore, according to the power law distribution of term frequencies in natural language (cf. Zipf's law), the logarithmic view leads to a concentration of most of the terms in the middle of the x-axis which would in a linear view mostly to be found indistinguishably right next to the y-axis. Figures 7 and 8 show sections of overviews of 1990 and 1991. In these figures, only some terms are represented by their word strings for clearness, too.

6 Conclusions and Further Work

In this paper, we have presented a new approach to the analysis of topics changing over time by considering changes in the gobal contexts of terms as indicative of a change of meaning. First experiments, carried out using data from contemporary news corpora for German and English, indicate the validity of the approach.

In particular, it could be shown that the proposed measure of a term's volatility is highly independent from a term's frequency.

An aggregated representation allows the user to get a direct overview over the most evolving topics covered in the processed documents. In an interactive application the user can explore more and less evolving aspects of the covered time span by zooming into certain areas. If the user finds an interesting term, it's easy to provide him with the curve of the volatility of this term showing the term's development over the time span. Using the significant co-occurrences, the user can be provided the most related terms as well. Combining those overviews of subsequent time spans, it is possible to show the terms' developments as a trajectories, one for every term. So, rising or declining topics can be identified by having the according terms moving along the x-axis while they gain or loose variance of volatility in contrast to other concepts which may stay in their area over the different overview representations.

In a next step, the analysis proposed can be extended to look at individual topics changing over those time spans identified as interesting. Instead of only looking at the terms that change their meaning over time, it might also be of value to look at those terms that for some time span retain a "stable" meaning, expressing a society's unquestioned consensus on a topic, as it were. In the long run, this approach might lead to an infrastructure for easily analyzing diachronic text corpora with many useful and interesting applications in trend and technology mining, marketing, and E-Humanities.

Acknowledgments

This research has been funded in part by DFG project *Topology-based Visual Analysis of Information Spaces*[5] as part of the Focus Project Nr. 1335 *Scalable Visual Analytics: Interactive Visual Analysis Systems of Complex Information Spaces*[6].

References

1. Allan, J.: Introduction to topic detection and tracking, pp. 1–16. Kluwer Academic Publishers, Norwell (2002)
2. Allan, J., et al.: Topic detection and tracking pilot study final report. In: Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop, pp. 194–218 (1998)
3. Dunning, T.E.: Accurate methods for the statistics of surprise and coincidence. Computational Linguistics 19(1), 61–74 (1993)
4. Heyer, G., Quasthoff, U., Wittig, T.: Text Mining: Wissensrohstoff Text – Konzepte, Algorithmen, Ergebnisse, 2nd edn. W3L-Verlag (2008)
5. Kleinberg, J.: Bursty and hierarchical structure in streams. In: KDD 2002: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 91–101. ACM Press, New York (2002)

[5] `http://www.visualanalytics.de/node/19/`
[6] `http://www.visualanalytics.de/`

6. Kumaran, G., Allan, J.: Text classification and named entities for new event detection. In: SIGIR 2004: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 297–304. ACM, New York (2004)
7. Swan, R., Allan, J.: Extracting significant time varying features from text. In: CIKM 1999: Proceedings of the eighth international conference on Information and knowledge management, pp. 38–45. ACM, New York (1999)
8. Swan, R., Allan, J.: Automatic generation of overview timelines. In: SIGIR 2000: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval, pp. 49–56. ACM, New York (2000)
9. Taylor, S.J.: Introduction to asset price dynamics, volatility, and prediction. In: Asset Price Dynamics, Volatility, and Prediction. Introductory Chapters. Princeton University Press, Princeton (2007)
10. Wang, X., McCallum, A.: Topics over time: a non-markov continuous-time model of topical trends. In: KDD 2006: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 424–433. ACM, New York (2006)

Modelling Illocutionary Structure: Combining Empirical Studies with Formal Model Analysis

Hui Shi[2], Robert J. Ross[1], Thora Tenbrink[1], and John Bateman[1]

[1] SFB/TR8 Spatial Cognition, Universität Bremen, Germany
[2] DFKI Bremen, Germany
{shi,robertr}@informatik.uni-bremen.de
{tenbrink,bateman}@uni-bremen.de

Abstract. In this paper we revisit the topic of dialogue grammars at the illocutionary force level and present a new approach to the formal modelling, evaluation and comparison of these models based on recursive transition networks. Through the use of appropriate tools such finite-state grammars can be formally analysed and validated against empirically collected corpora. To illustrate our approach we show: (a) the construction of human-human dialogue grammars on the basis of recently collected natural language dialogues in joint-task situations; and (b) the evaluation and comparison of these dialogue grammars using formal methods. This work provides a novel basis for developing and evaluating dialogue grammars as well as for engineering corpus-tailored dialogue managers which can be verified for adequacy.

1 Introduction

To achieve natural interaction with artificial systems, we believe it necessary for dialogue systems to leverage sufficiently off linguistic knowledge at a number of different strata. In particular, and as the focus of the current paper, we target the systematic analysis of *dialogue structures* at the illocutionary force level which explicitly define underlying dialogue 'grammars' without reference to either domain specifics or internal mental state dynamics. Our main goal here is to show how a particular formal approach can be used to construct, validate and compare these grammars at the illocutionary force level, in a way that provides a solid basis for engineering dialogue systems for their intended range of applications.

Austin's [3] observation that natural language utterances can be viewed as actions that alter the state of the environment has had a long lasting effect on how the structure of both natural and artificial dialogue is conceived. Dialogue acts [20] are now a prominent feature of the understanding of pragmatic phenomena [8], of the construction of artificial discourse partners [14], and of the analysis of the characteristics of human-human dialogues [5,13]. For coherent and effective interaction, dialogue acts need to be appropriately combined into sequences of communicative moves at various levels of abstraction. One traditional approach

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 340–353, 2010.

to capturing such sequences has been to employ recursive transition networks; in Lewin and Lane [12], for example, recursive transition networks were applied to model Conversational Game Theory (cf. [18,10]), an approach which combines dialogue grammars with discourse planning. In contrast to this direction of work, the focus of the current paper is on establishing dialogue grammars from discourses collected in empirical studies without the need for deep analysis of their underlying semantics.

Research on applying transition networks to dialogue control appears particularly relevant where tighter control, or verification, are issues for the behaviour of the systems constructed. Recently, some new efforts of formal dialogue modelling have been reported, especially for multi-agent dialogue (cf. [15,4,24,25]). As Walton [24,25] argues, verification of artificial communication protocols is highly desirable for multi-agent systems in which the communication is a coherent dialogue between several agents. Here, the separation of dialogue protocols from intentional reasoning makes it possible to verify the properties of dialogue protocols using mathematically well-founded methods. Following this idea, in the work reported here we apply a well-established technique from the Formal Methods Community of computer science to capture complex state transition networks, i.e., *Communicating Sequential Processes* (CSP) [9]. The particular value of formal methods, and in particular CSP, is the ability to subject formal specifications that are well founded in mathematical logic to very powerful analysis using mechanized theorem provers and model checkers. Thus, in contrast to the work of Walton, we employ formal methods to analyse features, complexity and coverage of illocutionary structures of dialogues in which humans are involved. Moreover, we also aim at the formal development of dialogue management systems employing the same techniques.

Illocutionary structural accounts of the kind we propose contrast with the many general dialogue theories where dialogue is modelled in terms of dialogue acts and their relation to plans and mental states in the agent or society design [8,23,16]. While these more *complete* models commonly encompass an underlying illocutionary model, that model is often only implicit and remains difficult to distinguish or separate from the encompassing dialogue theory as a whole. This leads to problems for verification and extension, which increase rapidly as the complexity of the modelled dialogues grows.

We structure the paper as follows. We begin in Section 2 by briefly reviewing the description of dialogue structure at a pure illocutionary force level. Section 3 then demonstrates the construction of illocutionary structure models from empirical data, taking as an example a recent empirical study conducted in our group on human-human dialogues in joint-task situations. In Sections 4 and 5 we present the details of our approach to formal model construction, evaluation and comparison. Finally, in Section 6 we discuss the relevance of such efforts both to the general study of dialogue structure and to its application in the construction of dialogue managers for use in human-computer interaction.

2 Illocutionary Structures and Recursive Transition Networks

We view illocutionary structure models as accounts of dialogue grammars which are independent of discourse content. In such *generalised* models, sometimes referred to as pattern-based dialogue models [27], recurrent interaction patterns or regularities of dialogue acts in dialogue at the illocutionary force level are identified and formally specified. Such discourse structure oriented models have been used widely in formal structural analysis and computational research [26,18,22,12]. This structural view on dialogue organization is also seen as an important contribution to organizing linguistic knowledge for statistical dialogue manager construction [11,17,7].

To illustrate the general properties of such illocutionary structure models, we take Sitter & Stein's Conversational Roles (COR) model [22] as an example. COR, like its forerunner, Winograd and Flores' '**C**onversation **f**or **A**ction' (CfA) model [26], is a generalised account of information-seeking dialogues at the illocutionary-force level. The motivation for choosing the COR model is that it has a clean structural account drawing on the original notion of illocutionary force. Sitter and Stein formulated the COR model as a Recursive Transition Network (RTN) composed of nested instances of a primary dialogue network along with three smaller networks, each of which captures individual dialogue moves within the complete RTN. The primary network is depicted in Fig. 1, where dialogue states are labelled circles and the arcs between these states capture the allowable discourse moves at each state. The starting state, labelled 1, therefore allows two moves, a *request* of A to B or an *offer* from B to A; the first leading to state 2, the second to state $2'$. State 5 is a terminal state, shown with double circles. To focus our discussion, *withdraw* moves are removed from the primary network. Thus, in the simplified COR model, dialogue participants are not allowed to withdraw an earlier dialogue move, e.g., an offer or a request.

To capture a broader range of dialogue features, such as implicit dialogue moves (modelled as jumps), the provision of contextual information (backing statements), and complex clarification sub-dialogues, the COR model treats individual discourse move arcs as aggregate dialogue moves rather than atomic dialogue acts. Structurally, then, these networks consist of either atomic dialogue acts (e.g., *A.evaluation*, *A.reject offer*), jump or empty moves, dialogue moves including possibly complete instances of the sub-dialogues (e.g., *request(A,B)*, *offer(B,A)*), or even the initial *dialogue(A,B)* network (thus showing the notion of recursive networks).

The network *request(A,B)*, the most complex of the three sub-structures of the COR model, is presented in Fig. 2. This network applies to the *request* and *offer* moves. The main feature of the request network is its satellite-nucleic treatment of requests where a request is viewed as a structure which typically consists of a nucleic request which can be supported by some satellite statement. This implies that COR enables interleaved subdialogues for clarifying contexts of a previous utterance—however the depth of the interleaving is always limited. The rules

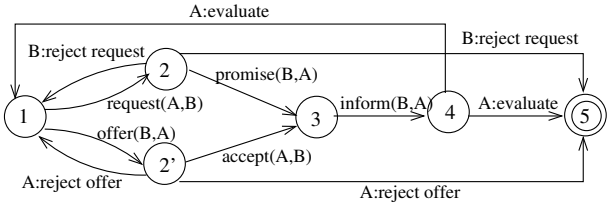


Fig. 1. Primary COR model network, adapted from Sitter and Stein (1996)

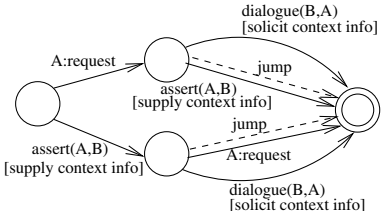


Fig. 2. COR model move rule request

promise(A,B) used to model *promise* or *accept* moves, and *inform(A,B)* to model *inform* and *assert* moves, will not be discussed further here.

3 Modelling Human-Human Dialogue

In this section, we turn to the modelling of human-human dialogue at the illocutionary structure level using discourses collected in empirical studies. We start by describing the empirical data we use and provide a linguistic analysis of specific dialogue phenomena and the dialogue structure encountered in an example. At the end of the section, we demonstrate how to construct a dialogue model using our empirical data, which is then evaluated in Section 4.

The experimental setup[1] involved pairs of identical doll's houses. Each trial had two participants of the same sex, and the task of one subject was to act as an instructor for the other, stating where particular items of doll's furniture were to be placed. The instructor was given a fully furnished doll's house with items of furniture, while the instructee only had an empty doll's house and the furniture items were placed to one side. The instructor and the instructee could not see each other's doll's house and could only communicate verbally. In contrast to the scenario in the Map Task corpus [2], there were no inconsistencies in the perceived scenes that required negotiation; instead, spatial actions were performed that changed the configuration, so as to fulfill the given task of arranging the furniture based on task-oriented dialogue. The experiment was run

[1] The experiments were carried out as part of a cooperation between Kenny Coventry (Northumbria University, UK), Elena Andonova and Thora Tenbrink (University of Bremen).

with native German speakers and so we provide English translations in this paper. 11 transcribed and annotated trials are used here. Fig. 3 presents a dialogue extract taken from one of these. In the extract the instructor and instructee are indicated by A and B, respectively. The numbering of turns reflects their position within the dialogue; the dyad's identity is marked by the identifier D301. Utterances are segmented pragmatically; apart from natural segments caused by turn shifts between speakers, each segment represents a coherent piece of information conveyed by a speaker. According to the analysis results shown in the next subsection, the COR dialogue acts and their combinations were used to annotate each utterance. Additionally, the dialogue act "hold" was attached to the utterances which hold a previous utterance rather than accepting or rejecting it.

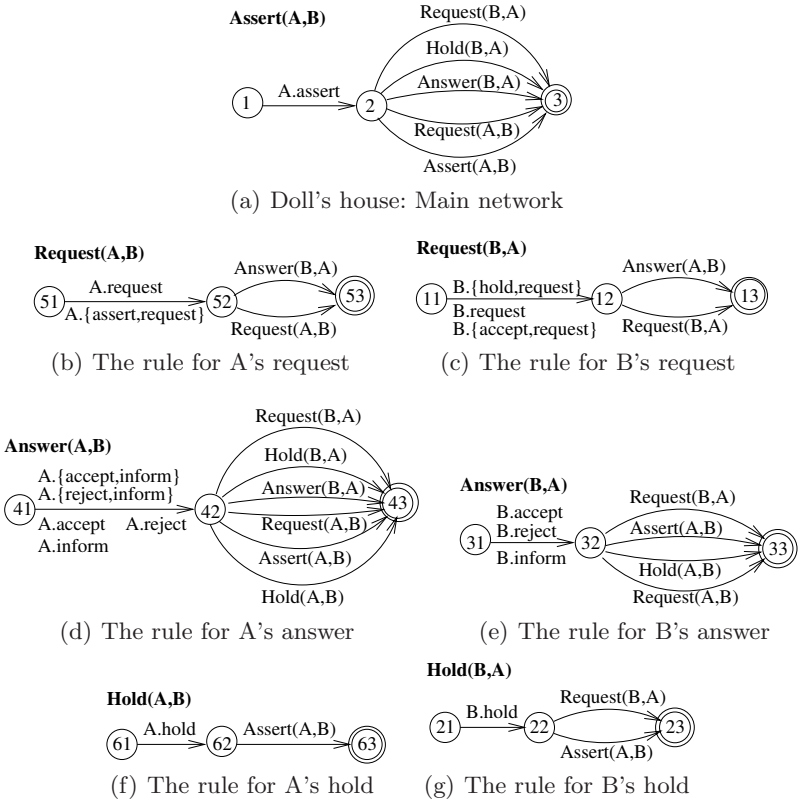
16	D301	A OK and then comes next to that, and horizontally then comes this sink	*assert*
17	D301	B OK, yes	*accept*
18	D301	B directly alongside	*request*
19	D301	A no, also at the wall, then there's this container there, these two things there	*reject, assert*
20	D301	A with a cupboard on top	*assert*
21	D301	B oh, so the thing to open up WHERE ON THE RIGHT two	*hold, request*
22	D301	A EXACTLY	*accept*
23	D301	B and that comes directly alongside	*request*
24	D301	A yes. Up to the wall	*accept, assert*

Fig. 3. A sample human-human dialogue extract from the doll's house experiment data

3.1 Analysis

We can see from the dialogue extract in Fig. 3 that the task is quite complex and there is considerable need for correction and clarification. In the following analysis we focus on the task level actions in the utterances. In segment no. 16, the instructor starts with an assertion concerning the location of an object and this is represented by the dialogue act *assert*. The instructee reacts in segment 17 by accepting it (*accept*), as indicated by "Okay, yes". In segment 18, however, this reaction is modified by a request for further information about the precise placement of the object, formulated as a Yes/No question: *request*. The instructor does not follow this up but simply rejects it in her answer in segment 19; she then proceeds with a different location description, which is again an *assert*.

Segment 20 is a delayed follow-up by the instructor on the previous utterance, providing a more precise description of the next object to be placed; again an *assert*. In segment 21, the instructee reacts to this description by providing another one of her own, based on visual information about the set of objects available to her. This reaction "holds" the instructor's utterance rather than accepting or rejecting it; additionally, it has the function of a request for information that can be answered by "yes" or "no": *hold, request*. This utterance points back to segment 19 in mentioning "two (...)"; but before this reference is completed, the



(a) Doll's house: Main network

(b) The rule for A's request

(c) The rule for B's request

(d) The rule for A's answer

(e) The rule for B's answer

(f) The rule for A's hold

(g) The rule for B's hold

Fig. 4. The doll's house dialogue grammar

utterance is interrupted by the instructor with an overlapping back-channelling act (segment 22, the overlap of speech is indicated by the capitals) who evidently considers herself to have obtained sufficient information to be able to accept B's utterance. Thus, while two utterances can overlap, this does not entail that dialogue acts at the illocutionary force level must be modelled as overlapping, since the back-channelled act is typically made only after the initialising act has been recognised (even if that initialising act is not yet fully articulated). In segment 23, the instructee, satisfied with the identification of the object, wishes to confirm the object's location by a *request*. This is confirmed by the instructor in segment 24, together with further spatial information: *accept, assert*.

3.2 Model Construction

Two of the authors independently annotated the first 50 utterances of each of 11 dialogue trials collected in the empirical study in the way shown in the above example analysis. The few cases of diverging annotation were discussed until agreement was reached, which turned out to be unproblematic. These 11 trials

were then split into two groups. The first group, or the *construction set*, consisted of 6 dialogues which were used in model construction. The second group, or the *evaluation set*, consisted of the remaining 5 dialogues used to evaluate the completeness of the constructed model, which is discussed in Section 4.3.

Our approach to constructing a dialogue model from annotated empirical data is an incremental process with three major iterative steps: *creation*, *validation* and *improvement*. In the creation step a rudimentary dialogue model in the form of a recursive transition network is manually created; this consists of the dialogue patterns identified by the illocutionary analysis as discussed in Section 3.1. The construction set is then used in the next step to improve the coverage of the current dialogue model. If new discourse patterns are discovered, the model is improved by extending it with the new structures. This process continues until the model covers all the dialogues in the construction set. The following validation step is carried out with the aid of the software system that we describe in detail in Section 4.1 and relies upon the formal specification of the dialogue model given in Section 4.2.

The dialogue grammar based on the construction set is presented in Fig. 4. Fig. 4(a) depicts the main dialogic structure in which the instructor initiates the discussion on the task with an *assert*. The three rules associated with the subdialogues initiated by the instructee are *Request(B,A)* (Fig. 4(c)) starting with an information request; *Answer(B,A)* (Fig. 4(e)) with an acceptance, rejection or inform utterance; *Hold(B,A)* (Fig. 4(g)) with a holding utterance. The rules defining the subdialogic structures initiated by the instructor are depicted in Figures 4(b), 4(d) and 4(f).

4 Formal Model Evaluation

Whereas an informal review of dialogue grammar can reveal certain characteristics of those grammars, a formal analysis provides a more powerful means to identify particular deficiencies and to allow us to concretely establish the quality of grammars with respect to target dialogue types. To date the evaluation of the illocutionary grammar of a dialogue system has mostly been integrated into the evaluation of the dialogue system as a whole (cf. [1]). One possible reason for this is that in many advanced dialogue systems, such as those based on information states, the illocutionary structure has not been defined explicitly and so is not even available for dedicated testing. In contrast to this, we adopt techniques from the Formal Methods Community in order to capture and evaluate these illocutionary structures in isolation, drawing on the detailed specification of dialogue patterns as Recursive Transition Networks (RTNs).

4.1 Formal Methodology and Tools

We have applied the well established method of *Communicating Sequential Processes* (CSP) to capture dialogue structure models. The specification language CSP is associated with a formalisation that allows verification of properties of

parallel processes by means of automatic logical deduction. The CSP language, its mathematical foundations and its possible applications within the Formal Methods Community have been thoroughly investigated [9,19]. Here we apply the method to dialogue modelling and analysis, a completely new area of application. Nevertheless, our approach is not restricted to CSP, other formal finite state methods could be applied equally.

To perform the actual analysis of the grammars, coded as CSP specifications, we adopt the FDR model-checker [6]. Its method of establishing whether a property holds is to test for refinement of the candidate machine capturing the required specification. Refinement relations can be defined for systems described in CSP in several ways—depending on the semantic model of the specification language used. In subsequent sections, we use the refinement relation in the *Traces* model, since we are at present only concerned with dialogue processes which can be covered by a given dialogue grammar. If a process P *refines* (or implements) a process Q in the Traces model, then P can accept an event (or, in our case, a dialogue move) only if Q can do so as well. If P refines Q and Q refines P in the Traces model, then we say they are equivalent in that model; this would mean in our case that two dialogue grammars are equivalent with respect to the dialogues they can accept.

FDR can, however, only be used for the analysis of dialogue grammars with finite states, although recursive transition networks specified by CSP can model dialogue structures with infinite states.

4.2 Formal Specification of Dialogue Models

We begin by specifying the dialogue actions that each dialogue participant is able to perform. As discussed in Subsection 3.2, the elementary dialogue acts used to annotate the collected doll's house dialogues are *assert*, *action-directive*, *info-request*, *answer*, *accept*, *reject* and *hold*. These dialogue acts are often combined to annotate the illocutionary functions of many utterances. The FDR datatype *act* defines all the elementary and compound dialogue acts that may be used to annotate dialogues.

```
datatype act = assertion  | action_directive | info_request
             | answer | accept | reject | hold
             | assertAction_directive | assertAction_directiveInfo_request
             | holdInfo_request | answerAccept | answerReject
             | answerRejectAction_directive | acceptInfo_request
             | answerAcceptAction_directive
```

Here, the compound dialogue acts are named as the combination of their elementary ones. Thus, *assertAction_directive* represents the illocutionary function that contains both *assert* and *action_directive* functions; and *answerAccept* and *answerReject* the accept answer and reject answer, respectively. Since every *answer* act is also an *assert* act, we just take the *answer* in compound acts. The elementary act *assert* is renamed to *assertion* to avoid conflict with the FDR keyword "assert" for defining proof obligations.

The following CSP process specifies the main dialogue rule *Assert(A,B)* in Fig. 4(a) at the illocutionary level. Here → and [ ] are two CSP operators necessary for the present specification. → defines the sequential occurrence of dialogue moves in a process, and [ ] arbitrary selection between several possibilities. *A.assert* means that the dialogue participant *A* makes an *assert* utterance. After that, *B* is allowed to make a reaction as specified by *RequestBA*, *HoldBA* or *AnswerBA*; alternatively, *A* may put forward a request by *RequestAB* or give a new assert by *AssertAB*. This specification reflects the corresponding dialogue structure. Each dialogue (sub)structure can be specified in a similar way and so will not be presented further here.

```
AssertAB =
    A.assert ->
   (RequestBA [] HoldBA [] AnswerBA [] RequestAB [] AssertAB)
```

4.3 Evaluation of the Doll's House Dialogue Grammar

As indicated, while a construction set of dialogues was used iteratively to construct the first complete version of the doll's house dialogue grammar, a second *evaluation set* was reserved for grading. After a dialogue in the evaluation set has been annotated, it is specified straightforwardly in the specification language CSP as a sequence of dialogue moves. The model checker FDR is then used to prove whether the dialogue is covered by the grammar, automatically.

Following the above procedure, we evaluated the dialogue grammar presented in Fig. 4 with an evaluation set containing five dialogue trials. The five dialogues in the evaluation set, each with 50 utterances, contain altogether 232 task relevant utterances. Among these 232 utterances, 35 utterances (about 15%) involved 8 subdialogues whose structures are not included in the doll's house dialogue grammar. These belong to the following two classes.

Undefined combinations. In the dialogue grammar depicted in Fig. 4, a set of dialogue moves and their combinations are used. However, some additional compound moves are used by both dialogue participants in the evaluation set. For instance, FDR detected a combination of the elementary moves *inform*, *accept* and *request*, which is not contained in the dialogue moves of the doll's house grammar, in A's utterance of the following subdialogue.

> 15 D220 B opposite on the right, for me, yes, is that also on the right for you at all
> 16 D220 A (laughter) yes, why not

Additional subdialogue structures. Usually it is expected that an answer should be provided by the dialogue partner after a participant has put forward a question, as in the subdialogue initiated by the instructee with a *request* (segment 23 in Fig. 3), where the instructor immediately provided an *accept* (segment 24).

However, in the evaluation set there are cases of *request* utterances that are ignored by the dialogue partner. For example, in the following subdialogue, after

the instructee's question the instructor does not answer it directly, but instead raises a new question to check whether the instructee has placed the commode in a particular way. The exceptional nature of this exchange with respect to the dialogue grammar being verified is also detected by the model-checker FDR.

47 D307 B should I push the commode leftwards
48 D307 A so you have the commode on the dividing wall, and you pushed it all the way back

The automatic evaluation then leads to an improved doll's house dialogue grammar. This grammar has been used for formal analysis and comparison as described in the next section.

5 Formal Comparison of Dialogue Models

In the previous section we demonstrated how CSP's *Trace* refinement relation can be used to evaluate an illocutionary dialogue grammar using the model-checker FDR. In this section we take advantage of the formal approach to *compare* dialogue models by relating the doll's house model to the COR model, thereby discussing their similarities and differences.

The formal comparison is based on the state transition diagrams generated by FDR from a CSP specification. The state transition diagram for the COR model, for example, consists of 26 states and 63 transitions. The model checking with FDR shows that neither the COR grammar covers the doll's house dialogue grammar, nor does the doll's house grammar cover the COR grammar. Specifically, the model checking results exemplify the following dialogic phenomena: the *offer*-pattern, which is considered in the COR model, is not covered by the doll's house grammar. The dialogue act *offer* is proposed in the COR model for the information provider to present some alternative information in case a direct answer to a question was unavailable. If we allow the instructee to provide alternative information, then we should extend the doll's house dialogue structure with an *offer* substructure. FDR confirms then that the COR model refines the doll's house model with this extension.

Alternatively, by checking whether the doll's house dialogue grammar refines the COR grammar, a number of discrepancies between the idealised computational models and natural dialogues were uncovered. Firstly, COR does not allow multiple occurrences of asserts or requests of one dialogue participant in a single dialogue turn. However, this was actually a fairly common feature of our human-human interaction. Another feature which is readily evident in the doll's house model is compound dialogue moves in a single utterance. A third significant discrepancy between the abstract models and empirical evidence concerns the effect of "holding", where one interlocutor does not state his/her attitude towards a partner's request or assertion. Thus, without extensive simplification of the doll's house grammar or extension of the COR grammar, it is not possible to prove that the doll's house dialogue model refines the COR model.

6 Discussion

Choosing the most appropriate set of communicative acts is one of the most complex tasks in the structural analysis approach pursued here. Care must be taken, on the one hand, to avoid over-simplification to the point where the structural model collapses down to a two-state initiate-response network with jumps. On the other hand, models which rely heavily on domain specific communicative acts lose their generality and prevent us from applying interaction models across genres and application scenarios.

One significant benefit of the explicit definition of illocutionary grammars is that it enables discussion of the underlying complexity of a dialogue structure. Our approach to illocutionary structure modelling and analysis is based on recursive transition networks, which can be viewed as a context-free grammar. On the other hand, any context free grammar can also be modelled by a recursive transition network. Therefore, we have applied recursive transition networks instead of finite-state machines to explicitly model dialogue structures, since a finite state machine cannot accept the complete set of context-free languages, but rather accepts the set of regular languages. Although the formal language CSP is able to specify a more general set of context-free illocutionary structures modelled by recursive transition networks, FDR can only be used for the automatic analysis and comparison of dialogue grammars with regular structures. The automatic analysis of context-free dialogue grammars requires more powerful tools. However, just how complex such grammars, and the discourse structures that they support need to be is still an open issue at this time.

While the dialogue modelling approach advocated here is dialogue theory agnostic, the approach has also brought considerable advantages for the development of dialogue management. An application of formal dialogue modelling to the design, development, and analysis of dialogue systems is reported in Shi *et al.* [21]. The techniques reported there build upon the illocutionary modelling approach and by applying the formal notion of refinement. When one component satisfies at least the same conditions as another, then that component may be replaced by a less abstract one without degrading the properties of the system. Whereas a complete dialogue manager must include domain specific data and components, which can vary wildly and make complete system modelling difficult, during initial development it is not necessary to consider such domain specific information. Instead, development can focus on the creation and specification of a generalised dialogue model, such as those discussed in this paper. Then, once the generalised dialogue model has been established, communication channels between this high level model and application specific components can be introduced without compromising the proved properties of the dialogue management component.

To support the application of the formal dialogue modelling approach in dialogue system development, we have developed a toolkit which integrates the model-checker FDR for formally analysing dialogue grammars, a generator for automatically constructing the state machine from a in CSP specified dialogue grammar, and an interpreter for controlling dialogues using the generated state machine.

Although the formal approach discussed here is based on dialogue grammars with finite-state structures, our modelling approach is quite different from traditional finite-state based dialogue management [14] in that our illocutionary grammars are independent of propositional content. Nonetheless, our dialogue grammars can be integrated with rich models of propositional state such as those provided by information state accounts, thus facilitating powerful, but nevertheless verifiable, dialogue management formalisms which tease out different aspects of dialogue modelling and control.

7 Conclusions

The modelling and characterisation of dialogue at the illocutionary force level allows us to examine the characteristics of dialogue structure in a way which is both independent of genre/application-specific features and free of any particular theory of dialogue production and understanding. In this paper, we have proposed methods which can be applied in the formal analysis of such dialogues and tools for supporting that analysis. Based on this approach we were able to prove a number of characteristics of some existing dialogue models, including whether one model completely covers another, and whether a concrete discourse sequence is a case covered by a given dialogue model. Such a formal analysis contrasts with previous work with dialogue grammars that has focused on merely constructing the models and assessing their predictive power.

One appealing direction for future work is to apply our techniques to examine the structural differences between models extracted from different genres and experimental conditions. Moreover, the ability to formally specify dialogue models in the terms set out here encourages building on such specifications in the future development of standardized generic discourse models. As described in Section 3.2 and 4.3, the construction and evaluation of dialogue grammars from annotated dialogue corpora are processes containing many routine tasks. Thus, we are now extending our toolkit with a new component to support these tasks.

Acknowledgements

We gratefully acknowledge the financial support of the Deutsche Forschungsgemeinschaft (DFG) through the SFB/TR 8 Collaborative Research Centre on Spatial Cognition—projects I3-[SharC], I1-[OntoSpace], and I5-[DiaSpace], and the German Research Center for Artificial Intelligence (DFKI) for the work reported in this paper. We also particularly wish to thank Elena Andonova and Kenny Coventry for their cooperation concerning the doll's house experiments.

References

1. Alexandersson, J., Heisterkamp, P.: Some Notes on the complexity of Dialogues. In: Dybkjaer, L., Hasida, K., Traum, D. (eds.) Proceedings of the ACL 2000 workshop 1st Workshop on Discourse and Dialogue, Hong Kong (2000)

2. Anderson, A.H., Bader, M., Bard, E., Boyle, E., Doherty, G., Garrod, S., Isard, S., Kowtko, J., McAllister, J., Miller, J., Sotillo, C., Thompson, H., Weinert, R.: The HCRC Map task Corpus. Language and Speech 34(4), 351–366 (1991)
3. Austin, J.L.: How to do things with words. Harvard University Press (1962)
4. Bordini, R.H., Dennis, L.A., Farwer, B., Fisher, M.: Automated verification of multi-agent programs. In: 23rd International Conference on Automated Software Engineering, pp. 69–78. IEEE, Los Alamitos (2008)
5. Eggins, S.: An Introduction to Systemic Functional Linguistics, 2nd edn. Continuum (2004)
6. FS00. Failures Divergence Refinement FDR2 Preliminary Manual. Formal Systems (Europe) Ltd. (2001)
7. Geertzen, J.: Dialogue Act Prediction Using Stochastic Context-Free Grammar Induction. In: Proceedings of the EACL 2009 Workshop on Computational Linguistic Aspects of Grammatical Inference, Athens, Greece, March 2009, pp. 7–15. Association for Computational Linguistics (2009)
8. Grosz, B.J., Sidner, C.L.: Attention, Intentions and the Structure of Discourse. Computational Linguistics 12(3), 175–204 (1986)
9. Hoare, C.A.R.: Communicating Sequential Processes. Prentice-Hall, Englewood Cliffs (1985)
10. Houghton, G.: The Production of Language in Discourse: A Computational Model. PhD thesis, University of Sussex (1986)
11. Kita, K., Fukui, Y., Nagata, M., Morimoto, T.: Automatic Acquisition of Probabilistic Dialogue Models. In: Proceedings of ICSLP 1996, Philadelphia, PA, October 1996, vol. 1, pp. 196–199 (1996)
12. Lewin, I., Lane, M.: A Formal Model of Conversational Game Theory. In: Fourth Workshop on the Semantics & Pragmantics of Dialogue, Gothenburg, Sweeden (2000)
13. Martin, J.R.: English text: systems and structure. Benjamins, Amsterdam (1992)
14. McTear, M.F.: Spoken dialogue technology: Enabling the conversational user interface. ACM Computing Surveys (CSUR) 34(1), 90–169 (2002)
15. Parsons, S., Wooldridge, M., Amgound, L.: Properties and complexity of formal inter-agent dialogues. Journal of Logic and Computation 13(3), 347–376 (2003)
16. Perrault, C.R., Allen, J.F.: A plan-based analysis of indirect speech acts. American Journal of Computational Linguistics 6(3-4), 167–182 (1980)
17. Poesio, M., Mikheev, A.: The predictive power of game structure in dialogue act recognition: Experimental results using maximum entropy estimation. In: Proceedings of the International Conference on Speech and Language Processing (ICSLP 1998), Australia (1998)
18. Power, R.: The organisation of purposeful dialogues. Linguistics 17, 107–151 (1979)
19. Roscoe, A.W.: The Theory and Practice of Concurrency. Prentice-Hall, Englewood Cliffs (1998)
20. Searle, J.: Speech Acts. Cambridge Univesity Press, Cambridge (1969)
21. Shi, H., Ross, R.J., Bateman, J.: Formalising control in robust spoken dialogue systems. In: Aichernig, B.K., Beckert, B. (eds.) Proceedings of Software Engineering & Formal Methods 2005, pp. 332–341. IEEE Computer Society, Los Alamitos (2005)
22. Sitter, S., Stein, A.: Modeling Information-Seeking Dialogues: The *Co*nversational *R*oles Model. Review of Information Science 1(1) (1996)
23. Traum, D.R., Allen, J.F.: Discourse obligations in dialogue processing. In: 32nd. Annual Meeting of the Association for Computational Linguistics, New Mexico State University, Las Cruces, New Mexico, pp. 1–8 (1994)

24. Walton, C.D.: Model checking agent dialogues. In: Leite, J., Omicini, A., Torroni, P., Yolum, p. (eds.) DALT 2004. LNCS (LNAI), vol. 3476, pp. 132–147. Springer, Heidelberg (2005)
25. Walton, C.D.: Verifiable agent dialogues. Journal of Applied Logic 5(2), 197–213 (2007)
26. Winograd, T., Flores, F.: Understanding computers and cognition: a new foundation for design. Ablex, Norwood (1986)
27. Xu, W., Xu, B., Huang, T., Xia, H.: Bridging the gap between dialogue management and dialogue models. In: Proceedings of the Third SIGdial Workshop on Discourse and Dialogue, Philadelphia, USA, pp. 201–210 (2002)

A Polyphonic Model and System for Inter-animation Analysis in Chat Conversations with Multiple Participants

Stefan Trausan-Matu[1,2] and Traian Rebedea[1]

[1] "Politehnica" University of Bucharest, Department of Computer Science and Engineering, 313 Splaiul Independetei, Bucharest, Romania
[2] Research Institute for Artificial Intelligence of the Romanian Academy, 13 Calea 13 Septembrie, Bucharest, Romania
stefan.trausan@cs.pub.ro, traian.rebedea@cs.pub.ro

Abstract. Discourse in instant messenger conversations (chats) with multiple participants is often composed of several intertwining threads. Some chat environments for Computer-Supported Collaborative Learning (CSCL) support and encourage the existence of parallel threads by providing explicit referencing facilities. The paper proposes a discourse model for such chats, based on Mikhail Bakhtin's dialogic theory. It considers that multiple voices (which do not limit to the participants) inter-animate, sometimes in a polyphonic, counterpointal way. An implemented system is also presented, which analyzes such chat logs for detecting additional, implicit links among utterances and threads and, more important for CSCL, for detecting the involvement (inter-animation) of the participants in problem solving. The system begins with a NLP pipe and concludes with inter-animation identification in order to generate feedback and to propose grades for the learners.

Keywords: Discourse analysis, conversation, chat, dialogism, polyphony, Computer-Supported Collaborative Learning.

1 Introduction

The goals followed by our approach and, meanwhile, the contributions of this paper are both theoretical and practical. First, we propose polyphony as theoretical model of a particular kind of online conversations: instant messenger (chat) conversations with multiple participants. The practical goal was to implement a system for analyzing such chats and providing feedback in order to encourage the appearance of multiple voices (or positions, in an extended sense [1, 2]), of parallel and intertwining threads of discussions. These aims may be obtained by catalyzing debates and the inter-animation of the participants, which are premises for supporting understanding, studying and creative thinking of virtual teams of learners or researchers. The implemented system was developed as a module in the EU FP7-IST project 'Language Technologies for Lifelong Learning' (LTfLL) and it is now in the evaluation phase [3].

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 354–363, 2010.

The implemented analysis method integrates results from NLP (content and discourse analysis), Social Networks Analysis (SNA) [4] and, a novel idea, the identification of polyphonic threading in chats [2]. The system was used for Computer Supported Collaborative Learning (CSCL) [5] in assignments for computer science and engineering students. As preparation for these assignments, the tutors group students in small teams of 4-7 participants, each of them being assigned a topic to study and then to support it in chat debates. They read some materials about that topic in order to understand the subject in detail. During the discussions, they present their points of view, they debate and inter-animate (arguing on their assigned topics), all of these improving their own and the others' understanding of the domain. After concluding a chat session, they can launch several widgets from the system, which provide graphical and textual feedback and preliminary scores both for each student and for the group as a whole. The tutors also use the system for providing them insights for writing a detailed feedback and grading the students.

The paper continues with a section introducing some basic theoretical ideas used in the system. The third section presents the implemented system.

2 Polyphony and Inter-animation

For discourse analysis in NLP two different situations are usually considered: monologue and dialogue. In monologues, an unidirectional model of communication is considered, from a speaker to a listener [6]. One of the main ways of analyzing discourse is the detection of local relations and measuring coherence, like in the Rhetorical Schema Theory (RST) [7], which considers a hierarchical decomposition of a text, like Centering Theory [8], or in other co-reference resolution systems [6].

In dialogues usually a phone-like (or face-to-face) type of conversation is considered. Typically, speech acts, dialog acts or adjacency pairs [6] are the units of analysis. Even if there are attempts to analyze conversations with multiple participants using transacts [10], this approach is also based on a two interlocutors' model. For chats, TF-IDF [11, 12], Latent Semantic Analysis [12, 13, 14], Naïve Bayes [15], Social Network Analysis [13], WordNet (wordnet.princeton.edu) [11, 13], Support Vector Machines and Collin's perceptron [10], and the TagHelper environment [16] are used for detection of topics and links [11], dialog acts [15], lexical chains [13] or other complex relations [16].

In phone and face-to-face-like dialogs only one person usually speaks at a given moment in time, determining a single thread of discussions. However, some chat environments, like the one used in the Virtual Math Teams (VMT) project [17] offer explicit referencing facilities, which means that users may indicate to which previous utterance(s) they refer to. This facility is extremely important in chat conversations with more than two participants because it allows the existence of several discussion threads or voices, in parallel. The co-occurrence of several voices gives birth to inter-animation and polyphony, phenomena identified in any text by Mikhail Bakhtin [18].

Voices may be considered as particular positions, which may be taken by one or more persons when they emit an utterance, which may have both explicit (like those provided by the VMT chat environment [17]) and implicit links (for example, lexical

chains, co-references or argumentation links) and influence other voices. Each utterance is filled with 'overtones' of other utterances [1]. Moreover, by the simple fact that they co-occur, voices are permanently *inter-animating*, entering in competition, generating multivocality in any conversation and even in any text (in Bakhtin's dialogic theory everything is a dialog [18]) or, as Bakhtin calls it, a "*heteroglossia,* which grows as long as language is alive" [1].

In order to detect overtones and inter-animation in chats, in our system we start from the explicit and implicit links among utterances. Thus, a graph is constructed connecting utterances and, in some cases, words. In this graph, threads may be identified. Each thread may be considered as a voice which becomes less or more powerful than the others. Among chat voices, both sequential and counterpointal, transversal relations similar to polyphonic music may be identified [18, 2]. From these data, several measures of contributing to the conversation may be computed, for each participant and for the group as a whole.

3 Automatic Analysis of Chats with Multiple Participants

The input of the system for analysis and giving feedback is a chat log similar to the one presented in Figure 1. An XML schema was designed for encoding chat conversations and discussion forums. Each utterance has an unique identifier, ('genid') and the existing explicit references ('ref') to previous utterances, which were specified by the participants using the facility provided by the VMT environment. In addition to annotating the elements of a chat, the schema also includes at the end data generated by the system.

The input data may be in different formats besides the above XML schema. A preprocessing module transforms these formats to respect the XML schema. The supported formats are: saved chats from Yahoo Messenger in text format, other text format chats, VMT format.

Figure 2 presents an overview of the architecture of the system and specifies the communication between the modules. Some of these modules – the ones that are heavily based on NLP technologies – are presented in detail in the following subsections. Others, like the one used for Social Network Analysis were presented in other papers [4].

3.1 The NLP Pipe

The processing starts with a NLP pipe containing the following: spelling correction, stemmer, tokenizer, Named Entity Recognizer, POS tagger and parser, and NP-chunker. The components of the NLP pipe are mainly those provided by the Stanford NLP software group (http://nlp.stanford.edu/software), with the exception of the spell checker (which uses Jazzy, http://www.ibm.com/developerworks/java/library/j-jazzy/ and http://jazzy.sourceforge.net/). Two alternative NLP pipes are under development, integrating modules from GATE (http://gate.ac.uk) and LingPipe (http://alias-i.com/lingpipe/).

```
<?xml version="1.0" encoding="UTF-8" ?>
 <Dialog time="2005-01-11 09:26:11" description="this is an assignment for the NLP
       course"  file="chat_input_1.xml"  id="Social  networks13_6_200610_57_10"
       language="en|fr|ro" name="chat-12-A" subject="about pragmatics" team="12">
 <Participants>
   <Person nickname="Alex" realname="Bibi Ionescu" />
   <Person nickname="vvalcea" realname="" />
   <Person nickname="Adrian" />
 </Participants>
 <Topics>
   <Itemset description="NLP - pragmatics">
     <Item>speech act</Item>
    <Item description="cnf. Grice's theory">implicature</Item>
  </Itemset>
      <Itemset>.....</Itemset>
 </Topics>
 <Body>
   <Turn nickname="Alex">
     <Utterance genid="1" ref="0" time="2005-01-11 09:26:03"> hello all </Utterance>
   </Turn>
   <Turn nickname="Adrian">
     <Utterance genid="2" ref="0" time="2005-01-11 09:27:18">hi</Utterance>
   </Turn>
    <Turn nickname="vvalcea">
      <Utterance genid="3" ref="1" time="2005-01-11-09:29:29"> Hello Alex </Utterance>
   </Turn>
.....
  </Body>
 </Dialog>
```

Fig. 1. A fragment of a chat log encoding

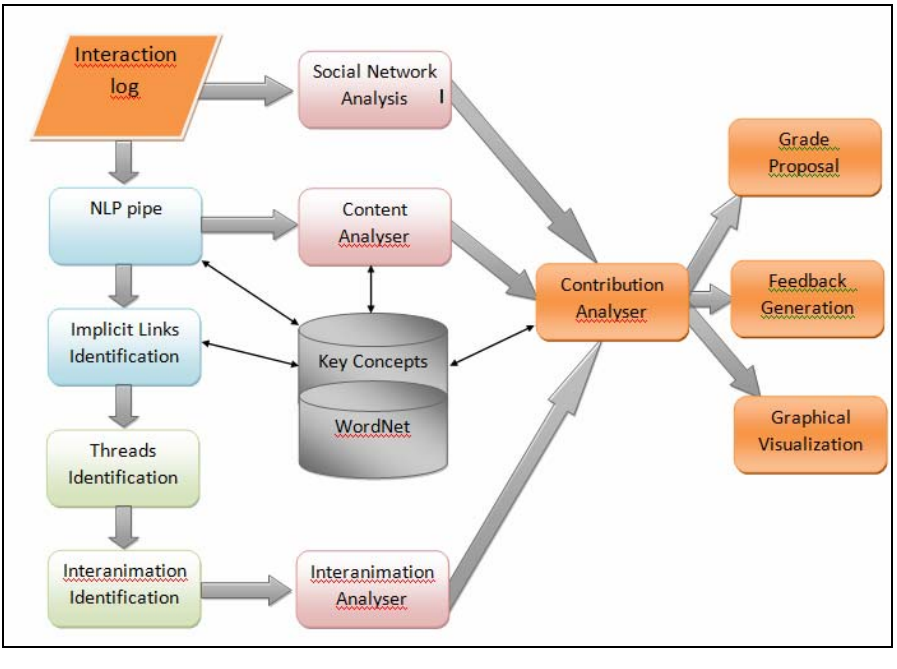


Fig. 2. Main modules of the analysis and feedback system

3.2 Pattern Language

Because important parts of the processing in the system are based on patterns identified by cue phrases, a module, called 'PatternSearch' was implemented for searching occurrences that match expressions specified by the user in a log of a chat or a forum. In addition to a simple regular expression search, the module allows considering not only words, but also synonyms, hypernyms and hyponyms via WordNet, words' stems and their part of speech (POS). Another novel facility is the consideration of utterances as a search unit, for example, specifying that a word should be searched in the previous *n* utterances and that two expressions should be in two utterances.

For example, the expression `<S "convergence"> #[*] cube` searches pairs of utterances that have a synonym of "convergence" in the first utterance and "cube" in the second. One result from a particular chat is the pair of utterances 1103 and 1107:

```
1103 # 1107. overlap # cube [that would stil have to acount for the
overlap that way] # [an idea: Each cube is assigned to 3 edges. Then add
the edges on the diagonalish face.]
```

The search is made at utterance level - the program checks the utterances one by one (and if there is a match between a part of the utterance and the searched expression, both the utterance and the specific text that matched are indicated).

PatternSearch is used in several other modules: cue-phrases identification, implicit links identification and adjacency pairs identification.

3.3 Content Analysis

The content analysis identifies the main concepts of the chat or forum using the NLP pipe, cue-phrases and graph algorithms [2]. It also identifies speech acts (a set derived from DAMSL [19]) and argumentation types in utterances (as in Toulmin's theory [20]: Warrant, Concession, Rebuttal and Qualifiers). Concepts and their synonyms are searched in the lexical database WordNet (wordnet.priceton.edu) and in a collection of key concepts and their inter-relations for the subject, provided by the teacher.

Advanced NLP and specific discourse analysis identify various types of implicit links:

- Repetitions (of ordinary words or of Named Entities), which were identified by Tannen as very important for detecting the involvement of the participants in a conversation [21];
- Lexical chains, which identify relations among the words in the same post / utterance or in different ones, by using semantic similarity measures based on WordNet;
- Adjacency pairs [6] – pairs of specific speech acts – e.g. answers to a single question in a limited window of time (in which the echo of the "voice" of the question remains), greeting-greeting;
- Co-references (the BART system [22] is used – see also http://bart-coref.org/)

3.4 Words, Key Concepts, Voices, and Threads

In the implementation of our analysis tool, we start from the key concepts and associated features that have to be discussed and that are provided by the teacher. Each

participant is assigned to support a position which corresponds to a key concept. That corresponds to a kind of implicit voice emitting that concept and the associated features. We may identify other, additional voices in the conversation by detecting recurrent themes, new concepts. Therefore, a first, simple perspective is to have a word-based approach on voices: We consider that a repeated word (that is a noun, verb, adjective or adverb) becomes a voice [21, 2]. The number of repetitions and some additional factors (e.g. presence in some specific patterns) may be used to compute the strength of that voice (word).

We use voices to keep track of the position that each participant has to support, in order to identify divergences and conjunctions. This position is, as mentioned above, an implicit voice. For a given small period of time, the last utterances are echo-like voices. For example, answers may be associated to questions that are present in a given time window.

Voices continue and influence each other through explicit or implicit links. In this perspective, voices correspond to chains or threads of utterances. They may be a reasoning or argumentation chain [20], a chain of rhetorical schemas, chains of co-references, lexical chains and even only chains of repeated words, in the idea of Tannen [21]. The identification of argumentation chains, rhetorical schemas or co-references in texts and conversations are very difficult tasks for Natural Language Processing. Chains of repeated words, however, are very easy to detect, the sole problem being the elimination of irrelevant repeated words. Lexical chains can also be detected, but their construction is more difficult and the resulted lexical chains are greatly influenced by the choice of the ontology and similarity measures.

3.5 Polyphony, Inter-animation and Collaboration

In polyphony, the most advanced kind of music compositions, a number of melodic lines (or "voices," in an extended, non-acoustical perspective) jointly construct a harmonious musical piece, generating variations on one or several themes. Dissonances should be resolved, even if several themes (melodies) or theme variations are played simultaneously, and even if sometimes the voices situate themselves in opposing positions.

Voices in polyphonic music have two dimensions, the sequential threading of utterances or words and the transversal one implicitly generated by the coincidence of multiple voices. In addition, another dichotomy, the unity-difference (or centrifugal-centripetal [1]) opposition may also be observed.

The evaluation of the contributions of each learner considers several features like the coverage of the expected concepts, readability measures, the degree to which they have influenced the conversation or contributed to the inter-animation. In terms of our polyphonic model, we evaluate to what degree they have emitted *sound* and *strong* utterances that influenced the following discussion, or, in other words, to what degree the utterance became a strong voice [2].

The automatic analysis considers the inter-animation patterns in the chat [2]. It uses several criteria such as the presence in the chat of questions, agreement, disagreement or explicit and implicit referencing. In addition, the strength of a voice (of an utterance) depends on the strength of the utterances that refer to it. If an utterance

is referenced by other utterances that are considered important, obviously that utterance also becomes important [2].

By using this method of computing their importance, the utterances that have started an important conversation within the chat, as well as those that began new topics or marked the passage between topics, are more easily emphasized. If the explicit relationships were always used and the implicit ones could be correctly determined in as high a number as possible, then this method of calculating the contribution of a participant would be considered [2, 4].

The implemented system supports the analysis of collaboration among learners: It produces different kinds of information about discussions in chat and forum discussions, both quantitative and qualitative, such as various metrics, statistics and content analysis results such as the coverage of the key concepts related to executing a task and the understanding of the course topics or the inter-threaded structure of the discussion [2, 23]. In addition, the system provides feedback about the involvement of each learner, generates a preliminary assessment and visualizes the interactions and the social participation. Finally, the system identifies the most important chat utterances or forum posts (that express different opinions, missing topics/concepts, misleading posts, misconceptions or wrong relations between concepts).

The results of the contribution analyzer are annotated in the XML file of the chat or forum. The annotations are on utterances:

```
<UtteranceFeedback genid="53">
        <Grade type="overall">8.15</Grade>
        <SpeechAct>Continuation</SpeechAct>
        <SpeechAct>Info Request</SpeechAct>
        <SpeechAct>Statement</SpeechAct>
        <Argumentation>Claim</Argumentation>
</UtteranceFeedback>
```

and on participants:

```
<GeneralGrade nickname="AlexI">
      <Grade type="diction">20.07</Grade>
      <Grade type="spelling">13.16</Grade>
      <Grade type="fluency">25.63</Grade>
      <Grade type="pageRanking">20.44</Grade>
      <Grade type="utteranceStructure">22.89</Grade>
      <Grade type="nbrWordsProc">27.37</Grade>
      <Grade type="nbrDiffWordsProc">24.98</Grade>
      <Grade type="nbrUtterancesProc">25.06</Grade>
      <Grade type="nbrUtterancesProc">23.19</Grade>
      <Grade type="meanUtteranceWords">13.21</Grade>
      <Grade type="correctWordsProc">13.16</Grade>
      <Grade type="flesch">55.18</Grade>
      <Grade type="kincaid">8.23</Grade>
      <Grade type="fog">10.35</Grade>
      <Grade type="inDegree">25.51</Grade>
      <Grade type="outDegree">23.19</Grade>
      <Grade type="rank">22.09</Grade>
      <Grade type="eigen">100.01</Grade>
      <Grade type="closeness">16.79</Grade>
      <Grade type="centrality">17.46</Grade>
</GeneralGrade>
```

For the values describing the activity of the participants, Social Network Analysis, Latent Semantic Analyses and other techniques were used [4, 23]. These values are used for generating textual feedback, which include, besides the above numerical values:

- the list of most important (used, discussed) concepts in a chat / forum;
- the coverage of the important concepts specified by the tutor;
- the most important utterances of each participant (the ones with the largest scores)
- the score for an utterance (which uses a complex formula that takes into account the concepts used, dialog acts, the links between utterances and SNA factors [2, 23]);
- a score for each participant in the conversation;
- areas of the conversations with important collaboration (inter-animation, argumentation, convergence and divergence);
- other indicators and statistics that are going to be added with the development of the system.

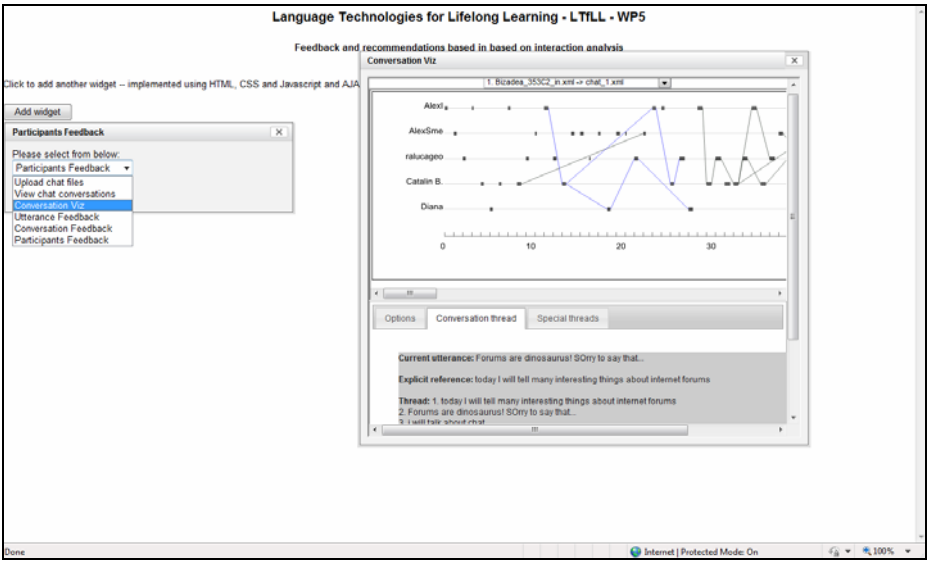


Fig. 3. A screenshot illustrating the graphical feedback and analysis system

As graphical feedback, the service provides interactive visualization and analysis of the conversations graph with filtering enabled. The graphical representation of chats was designed to facilitate an analysis based on the polyphony theory of Bakhtin and to permit the best visualization of the conversation. For each participant in the chat, there is a separate horizontal line in the representation and each utterance is placed in the line corresponding to the issuer of that utterance, taking into account its positioning in the original chat file – using the timeline as an horizontal axis (see Figure 3). Each utterance is represented as a rectangular node having a horizontal

length proportional with the textual length of the utterance. The distance between two different utterances is proportional to the time between the utterances [2].

An image of the facilities of graphical and textual visualization is presented in Figure 3.

4 Conclusions

A new theory, inspired from Bakhtin's ideas was proposed for explaining and evaluating collaboration and inter-animation in chats. Its main idea is the consideration of intertwining of discussion threads similarly with counterpoint in polyphonic music. Graphical visualization and various metrics are computed using a wide range of NLP techniques for the lexical, semantic and discourse analysis levels.

The first experiments with the implemented system showed that the polyphony model eases the development of algorithms and implementation of a system that analyses and gives feedback to participants in chats.

Acknowledgements

We would like to thank the students of "Politehnica" University of Bucharest, Department of Computer Science, which participated to our experiments. The research presented in this paper was partially performed under the FP7 EU STREP project LTfLL and the national CNCSIS grant K-Teams.

References

1. Bakhtin, M.: The Dialogic Imagination: Four Essays. University of Texas Press, Austin (1981)
2. Trausan-Matu, S., Rebedea, T.: Polyphonic Inter-Animation of Voices in VMT. In: Stahl, G. (ed.) Studying Virtual Math Teams, pp. 451–473. Springer, Heidelberg (2009)
3. Berlanga, A.J., Rosmalen, P.V., Trausan-Matu, S., Monachesi, P., Burek, G.: The Language Technologies for Lifelong Learning Project. In: Aedo, D.S.I., Chen, N., Kinshuk (eds.) Proceedings of the 9th IEEE International Conference on Advanced Learning Technologies, Riga, pp. 624–625 (2009)
4. Dascalu, M., Chioasca, E.V., Trausan-Matu, S.: ASAP-An Advanced System for Assessing Chat Participants. In: Dochev, D., Pistore, M., Traverso, P. (eds.) AIMSA 2008. LNCS (LNAI), vol. 5253, pp. 58–68. Springer, Heidelberg (2008)
5. Stahl, G.: Group Cognition: Computer Support for Building Collaborative Knowledge. MIT Press, Cambridge (2006)
6. Jurafsky, D., Martin, J.H.: Speech and Language Processing. In: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, 2nd edn. Pearson Prentice Hall, London (2009)
7. Mann, W.C., Thompson, S.A.: Rhetorical structure theory: A theory of text organization. Tech. rep. RS-87-190, Information Sciences Institute (1987)
8. Grosz, B.J., Joshi, A.K., Weinstein, S.: Centering: A framework for modeling the local coherence of discourse. Computational Linguistics 21(2), 203–225 (1995)

9. Dysthe, O.: The Multivoiced Classroom: Interactions of Writing and Classroom Discourse. Written Communication 13.3, 385–425 (1996)
10. Joshi, M., Rosé, C.P.: Using Transactivity in Conversation Summarization in Educational Dialog. In: Proceedings of the SLaTE Workshop on Speech and Language Technology in Education (2007)
11. Adams, P.H., Martell, C.H.: Topic detection and extraction in chat. In: Proceedings of the 2008 IEEE International Conference on Semantic Computing, pp. 581–588 (2008)
12. Schmidt, A.P., Stone, T.K.M.: Detection of topic change in IRC chat logs, http://www.trevorstone.org/school/ircsegmentation.pdf
13. Dong, A.: Concept formation as knowledge accumulation: A computational linguistics study. Artif. Intell. Eng. Des. Anal. Manuf. 20(1), 35–53 (2006)
14. Manning, C., Schütze, H.: Foundations of statistical Natural Language Processing. MIT Press, Cambridge (1999)
15. Kontostathis, A., Edwards, L., Bayzick, J., McGhee, I., Leatherman, A., Moore, K.: Comparison of Rule-based to Human Analysis of Chat Logs. In: 1st International Workshop on Mining Social Media Programme, Conferencia de la Asociación Española para la Inteligencia Artificial (2009)
16. Rose, C.P., Wang, Y.C., Cui, Y., Arguello, J., Stegmann, K., Weinberger, A., Fischer, F.: Analyzing Collaborative Learning Processes Automatically: Exploiting the Advances of Computational Linguistics in Computer-Supported Collaborative Learning. International Journal of Computer Supported Collaborative Learning (2007)
17. Stahl., G. (ed.): Studying Virtual Math Teams. Springer, Boston (2009)
18. Bakhtin, M.M.: Problems of Dostoevsky's poetics. University of Minnesota Press, Minneapolis (1993)
19. DAMSL, http://www.cs.rochester.edu/research/cisd/resources/damsl/RevisedManual/ (downloaded on November 22, 2009)
20. Toulmin, S.: The Uses of Arguments. Cambridge Univ. Press, Cambridge (1958)
21. Tannen, D.: Talking Voices: Repetition, Dialogue, and Imagery in Conversational Discourse. Cambridge University Press, Cambridge (1989)
22. Versley, Y., Ponzetto, S.P., Poesio, M., Eidelman, V., Jern, A., Smith, J., Yang, X., Moschitti, A.: BART: A Modular Toolkit for Coreference Resolution. In: Companion Volume of the Proceedings of the 46th Annual Meeting of the Association for Compuatational Linguistics (2008)
23. Dascalu, M., Trausan-Matu, S., Dessus, P.: Utterances Assessment and Summarization in Chat Conversations. In: CICLING 2010 Complementary Proceedings Volumes (to appear, 2010)

Computational Models for Incongruity Detection in Humour

Rada Mihalcea[1,3], Carlo Strapparava[2], and Stephen Pulman[3]

[1] Computer Science Department, University of North Texas
rada@cs.unt.edu
[2] FBK-IRST
strappa@fbk.eu
[3] Computational Linguistics Group, Oxford University
sgp@clg.ox.ac.uk

Abstract. Incongruity resolution is one of the most widely accepted theories of humour, suggesting that humour is due to the mixing of two disparate interpretation frames in one statement. In this paper, we explore several computational models for incongruity resolution. We introduce a new data set, consisting of a series of 'set-ups' (preparations for a punch line), each of them followed by four possible coherent continuations out of which only one has a comic effect. Using this data set, we redefine the task as the automatic identification of the humorous punch line among all the plausible endings. We explore several measures of semantic relatedness, along with a number of joke-specific features, and try to understand their appropriateness as computational models for incongruity detection.

1 Introduction

Humour is one of the most interesting and puzzling aspects of human behaviour, and it is rightfully believed to play an important role in an individual's development, as well as in interpersonal communication. Research on this topic has received a significant amount of attention from fields as diverse as linguistics, philosophy, psychology and sociology, and recent years have also seen attempts to build computational models for humour generation and recognition.

One of the most widely accepted theories of humour is the incongruity theory, which suggests that humour is due to the mixing of two disparate interpretation frames in one statement. One of the earliest references to an incongruity theory of humour is due to Aristotle [1] who found that the contrast between expectation and actual outcome is often a source of humour. The theory also found a supporter in Schopenhauer [20], who emphasized the element of surprise by suggesting that "the greater and more unexpected [...] the incongruity is, the more violent will be [the] laughter."

In more recent work in the field of linguistics, the incongruity theory has been formalized as a necessary condition for humour and used as a basis for the Semantic Script-based Theory of Humour (SSTH) [16] and the General Theory of Verbal Humour (GTVH) [2].

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 364–374, 2010.

The incongruity theory (also referred to as "incongruity resolution" theory) is a theory of comprehension. When a joke narration evolves, some "latent" terms are gradually introduced, which set the joke itself against a rigid and selective linear train of thought. In this way, a short circuit occurs: the available information does not become distorted in its content, but the starting point of the initial sequence suddenly changes. Because of these latent terms, the humorous input advances on two or more interpretation paths, consisting usually of a principal path of semantic integration that the listener is more aware of, and a secondary one, which is weak and latent but existent. This latter path gains more importance as elements are added to the current interpretation of the reader, and eventually ends up forming the punch line of the joke.

For instance, the following example (taken from [18]) illustrates this theory: "Why do birds fly south in winter? It's too far to walk." The first part of the joke (the set-up) has two possible interpretations, due to two possible foci of the question: "Why do birds go south?" (focus on "south") versus "Why do birds fly, when traveling south?" (focus on "fly"). The first interpretation is more obvious, also due to the phrase "in winter" which emphasizes this interpretation), and thus initially preferred. However, the punch line "it's too far to walk" changes the preference to the second interpretation, which is surprising and generates the humorous effect.

The goal of this paper is to develop and evaluate computational models for the identification of incongruity in humour. To this end, we build a data set consisting of short jokes (one-liners), each of them consisting of a set-up, followed by several possible coherent continuations out of which only one has a comic effect. The incongruity detection task is thus translated into the problem of automatically identifying the punch line among all the possible alternative interpretations. The task is challenging because all the continuations express some coherence with the set-up. We explore several measures of semantic relatedness, along with other joke-specific features, and try to understand their appropriateness as models of incongruity detection.

The paper is organized as follows: Section 2 introduces the data set we used in the experiments. In Section 3 we explore the identification of incongruity looking at two classes of models: models based on semantic relatedness (including knowledge-based and corpus-based metrics), and models based on joke-specific features. In Section 4 we report and discuss the results, and conclude the paper with final remarks.

2 Data

To evaluate the models of incongruity in humour, we construct a data set consisting of 150 set-ups, each of them followed by four possible continuations out of which only one had a comic effect. The task is therefore cast as an incongruity resolution task, and the accuracy of the models is defined as their ability to identify the humorous continuation among the four provided.

The data set was created in four steps. First, 150 one-liners were randomly selected from the humorous data set used in [13]. A one-liner is a short sentence with comic effects and an interesting linguistic structure: simple syntax, deliberate use of rhetoric devices (e.g. alliteration, rhyme), and frequent use of creative language constructions meant to attract the reader's attention. While longer jokes can have a relatively complex

narrative structure, a one-liner must produce the humorous effect "in one shot," with very few words. These characteristics make this type of humour particularly suitable for use in an automatic learning setting, as the humor-producing features are guaranteed to be present in the first (and only) sentence.

Each one-liner was then manually split into a set-up and a punch line. While there are several possible alternatives for doing this split, we tried to do it in a way that would result in a minimum-length punch line. The reason for this decision is the fact that we wanted to minimize the differences among the four alternative endings by keeping them short, thus making the task more difficult (and more realistic).

Next, we provided the set-up to 10 human annotators and asked them to complete the sentence. The annotators were required to write the continuations so that the sentences make sense and are complete. We also provided a range for the number of words to be used, which was determined as a function of the number of words in the punch line. Again, the reason for providing this range was to maximize the similarity between the punch line and the other continuations.

Table 1. Sample joke set-ups, with comic (a) and serious (b, c, d) continuations

Don't drink and drive. You might hit a bump and
a) spill your drink.
b) get a flat tire.
c) have an accident.
d) hit your head.
I took an IQ test and the results
a) were negative.
b) were average.
c) confused me.
d) said I'm dumb.
I couldn't repair your brakes, so I made
a) your horn louder.
b) a phone call.
c) a special stopping device.
d) some new ones.

Finally, the continuations were manually filtered, and three alternative continuations were kept for each one-liner. The filtering was done to make sure that the alternatives had no grammatical or spelling errors, were coherent, and did not have a comic effect. Table 1 shows three entries from our data set, each entry including one punch line (a) and three alternative continuations (b, c, d).

3 Models for Incongruity Detection

Humour recognition is a difficult task. In fact, the identification of incongruity in humour has to satisfy two apparently opposite requirements: jokes have to be coherent (and thus the requirement for coherence between the set-up and the punch line), but at

the same time they have to produce a surprising effect (and thus the requirement of an unexpected punch line interpretation based on the set-up).

In our experiments, since we assume that jokes already satisfy the first requirement (jokes are coherent since they are written by people), we emphasize the second requirement and try to find models able to identify the surprising effect generated by the punch line.

Specifically, we look at two classes of models: (1) models based on semantic relatedness, including knowledge-based metrics, corpus-based metrics and domain fitness, where we seek to minimize the relatedness between the set-up and the punch line; and (2) models based on joke-specific features, including polysemy and latent semantic analysis trained on joke data, where we seek to maximize the connection between the set-up and the punch line.

3.1 Knowledge-Based Semantic Relatedness

We use several knowledge-based metrics to measure the relatedness between the set-up and each candidate punch line. The intuition is that the correct punch line, which generates the surprise, will have a minimum relatedness with respect to the set-up.

Given a metric for word-to-word relatedness, similar to [12], we define the semantic relatedness of two text segments T_1 and T_2 using a metric that combines the semantic relatedness of each text segment in turn with respect to the other text segment. First, for each word w in the segment T_1 we try to identify the word in the segment T_2 that has the highest semantic relatedness, according to one of the word-to-word measures described below. Next, the same process is applied to determine the most similar word in T_1 starting with words in T_2. The word similarities are then weighted, summed up, and normalized with the length of each text segment. Finally the resulting relatedness scores are combined using a simple average.

There are a number of measures that were developed to quantify the degree to which two words are semantically related using information drawn from semantic networks – see e.g. [4] for an overview. We present below several measures found to work well on the WordNet hierarchy. All these measures assume as input a pair of concepts, and return a value indicating their semantic relatedness. The six measures below were selected based on their observed performance in other language processing applications, and for their relatively high computational efficiency.[1]

The **Leacock & Chodorow** [8] similarity is determined as:

$$Sim_{lch} = -\log \frac{length}{2 * D} \tag{1}$$

where $length$ is the length of the shortest path between two concepts using node-counting, and D is the maximum depth of the taxonomy.

The **Lesk** similarity of two concepts is defined as a function of the overlap between the corresponding definitions, as provided by a dictionary. It is based on an algorithm proposed by Lesk [9] as a solution for word sense disambiguation. The application of

[1] We use the WordNet-based implementation of these metrics, as available in the WordNet::Similarity package [15].

the Lesk similarity measure is not limited to semantic networks, and it can be used in conjunction with any dictionary that provides word definitions.

The **Wu & Palmer** [23] similarity metric measures the depth of two given concepts in the WordNet taxonomy, and the depth of the least common subsumer (LCS), and combines these figures into a similarity score:

$$Sim_{wup} = \frac{2 * depth(LCS)}{depth(concept_1) + depth(concept_2)} \tag{2}$$

The measure introduced by **Resnik** [17] returns the information content (IC) of the LCS of two concepts:

$$Sim_{res} = IC(LCS) \tag{3}$$

where IC is defined as:

$$IC(c) = -\log P(c) \tag{4}$$

and $P(c)$ is the probability of encountering an instance of concept c in a large corpus.

The next measure we use in our experiments is the metric introduced by **Lin** [10], which builds on Resnik's measure of similarity, and adds a normalization factor consisting of the information content of the two input concepts:

$$Sim_{lin} = \frac{2 * IC(LCS)}{IC(concept_1) + IC(concept_2)} \tag{5}$$

Finally, the last similarity metric considered is **Jiang & Conrath** [6]:

$$Sim_{jnc} = \frac{1}{IC(concept_1) + IC(concept_2) - 2 * IC(LCS)} \tag{6}$$

Note that all the word similarity measures are normalized so that they fall within a 0–1 range. The normalization is done by dividing the similarity score provided by a given measure with the maximum possible score for that measure.

3.2 Corpus-Based Semantic Relatedness

Corpus-based measures of semantic similarity try to identify the degree of relatedness of words using information exclusively derived from large corpora. In the experiments reported here, we considered two metrics, namely: (1) pointwise mutual information [22], and (2) latent semantic analysis [7].

The simplest corpus-based measure of relatedness is based on the **vector space model** [19], which uses a *tf.idf* weighting scheme and a cosine similarity to measure the relatedness of two text segments.

The **pointwise mutual information** using data collected by information retrieval (**PMI**) was suggested by [22] as an unsupervised measure for the evaluation of the semantic similarity of words. It is based on word co-occurrence using counts collected over very large corpora (e.g. the Web). Given two words w_1 and w_2, their PMI is measured as:

$$PMI(w_1, w_2) = log_2 \frac{p(w_1 \& w_2)}{p(w_1) * p(w_2)} \tag{7}$$

which indicates the degree of statistical dependence between w_1 and w_2, and can be used as a measure of the semantic similarity of w_1 and w_2. From the four different types of queries suggested by Turney [22], we are using the AND query. Specifically, the following query is used to collect counts from the AltaVista search engine.

$$p_{AND}(w_1 \& w_2) \simeq \frac{hits(w_1\ AND\ w_2)}{WebSize} \tag{8}$$

With $p(w_i)$ approximated as $hits(w_1)/WebSize$, the following PMI measure is obtained:[2]

$$log_2 \frac{hits(w_1\ AND\ w_2) * WebSize}{hits(w_1) * hits(w_2)} \tag{9}$$

Another corpus-based measure of semantic similarity is the **latent semantic analysis (LSA)** proposed by Landauer [7]. In LSA, term co-occurrences in a corpus are captured by means of a dimensionality reduction operated by a singular value decomposition (SVD) on the term-by-document matrix $\mathbf{T}$ representing the corpus.

For the experiments reported here, we run the SVD operation on two different corpora. One model (**LSA on BNC**) is trained on the British National Corpus (BNC) – a balanced corpus covering different styles, genres and domains. A second model (**LSA on jokes**) is trained on a corpus of 16,000 one-liner jokes, which was automatically mined from the Web [13].

SVD is a well-known operation in linear algebra, which can be applied to any rectangular matrix in order to find correlations among its rows and columns. In our case, SVD decomposes the term-by-document matrix $\mathbf{T}$ into three matrices $\mathbf{T} = \mathbf{U}\mathbf{\Sigma_k}\mathbf{V}^T$ where $\mathbf{\Sigma_k}$ is the diagonal $k \times k$ matrix containing the k singular values of $\mathbf{T}$, $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_k$, and $\mathbf{U}$ and $\mathbf{V}$ are column-orthogonal matrices. When the three matrices are multiplied together the original term-by-document matrix is re-composed. Typically we can choose $k' \ll k$ obtaining the approximation $\mathbf{T} \simeq \mathbf{U}\mathbf{\Sigma_{k'}}\mathbf{V}^T$.

LSA can be viewed as a way to overcome some of the drawbacks of the standard vector space model (sparseness and high dimensionality). In fact, the LSA similarity is computed in a lower dimensional space, in which second-order relations among terms and texts are exploited. The similarity in the resulting vector space is then measured with the standard cosine similarity. Note also that LSA yields a vector space model that allows for a *homogeneous* representation (and hence comparison) of words, word sets, and texts.

The application of the LSA word similarity measure to text semantic relatedness is done using the *pseudo-document* text representation for LSA computation, as described by Berry [3]. In practice, each text segment is represented in the LSA space by summing up the normalized LSA vectors of all the constituent words, using also a *tf.idf* weighting scheme.

3.3 Domain Fitness

It is well-known that semantic domains (such as MEDICINE, ARCHITECTURE and SPORTS) provide an effective way to establish semantic relations among word senses. This domain relatedness (or lack thereof) was successfully used in the past for word

[2] We approximate the value of $WebSize$ to $5\text{x}10^8$.

sense disambiguation [5,11] and also for the generation of jokes [21]. We thus conduct experiments to check whether domain similarity and/or opposition can constitute a feature to discriminate the humorous punch line.

As a resource, we exploit WORDNET DOMAINS, an extension developed at FBK-IRST starting with the English WORDNET. In WORDNET DOMAINS, synsets are annotated with subject field codes (or domain labels), e.g. MEDICINE, RELIGION, LITERATURE. WORDNET DOMAINS organizes about 250 domain labels in a hierarchy, exploiting Dewey Decimal Classification. Following [11], we consider an intermediate level of the domain hierarchy, consisting of 42 disjoint labels (i.e. we use SPORT instead of VOLLEY or BASKETBALL, which are subsumed by SPORT). This set allows for a good level of abstraction without losing relevant information.

In our experiments, we extract the domains from the set-up and the continuations in the following way. First, for each word we consider the domain of the most frequent sense. Then, considering the LSA space acquired from the BNC, we build the pseudo document representations of the domains from the set-up and the continuations respectively. Finally, we measure the domain (dis)similarity among the set-up and the candidate punch lines by using a cosine similarity applied on the pseudo document representations.

3.4 Other Features

Polysemy. The incongruity resolution theory suggests that humour exploits the interference of many different interpretation paths, for example by keeping alive multiple readings or double senses. Thus, we run a simple experiment where we check the mean polysemy among all the possible punch lines. In particular, given a set-up, from all the candidate continuations we choose the one that has the higher ambiguity.

Alliteration. Previous work in automatic humour recognition has shown that structural and phonetic properties of jokes constitute an important feature, especially in one-liners [14]. Moreover, linguistic theories of humour based on incongruity resolution, such as [2,16], account for the importance of meaning-to-sound theories of how sentences are being formed. Although alliteration is mainly a stylistic feature, it also has the effect of inducing expectation, and thus it can prepare and enforce incongruity effects.

To extract this feature, we identify and count the number of alliteration/rhyme chains in each example in our data set. The chains are automatically extracted using an index created on top of the CMU pronunciation dictionary.[3] The underlying algorithm is basically a matching device that tries to find the largest and longest string matching chains using the transcriptions obtained from the pronunciation dictionary. The algorithm avoids matching non-interesting chains such as e.g. series of definite/indefinite articles, by using a stopword list of functional words that cannot be part of an alliteration chain.

We conduct experiments checking for the presence of alliteration in our data set. Specifically, we select as humorous the continuations that maximize the alliteration chains linking the punch line with the set-up.

[3] Available at *http://www.speech.cs.cmu.edu/cgi-bin/cmudict*

4 Results

Table 2 shows the results of the experiments. For each model, we measure the precision, recall and F-measure for the identification of the punch line, as well as the overall accuracy for the correct labeling of the four continuations (as punch line or neutral). The performance of the models is compared against a simple baseline that identifies a punch line through random selection.

When using the knowledge-based measures, even if the F-measure exceeds the random-choice baseline, the overall performance is rather low. This suggests that the typical relatedness measures based on the WordNet hierarchy, although effective for the detection of related words, are not very successful for the identification of incongruous concepts. A possible explanation of this low performance is the fact that knowledge-based semantic relatedness also captures coherence, which contradicts the requirement for a low semantic relatedness as needed by a surprising punch line effect. In other words, the measures are probably misled by the high coherence between the set-up and the punch line, and thereby fail to identify their low relatedness.

A similar behaviour is observed for the corpus-based measures: the F-measure is higher than the baseline (with the exception of the LSA model trained on BNC), but the overall accuracy is low. Somehow surprising is the fact that contrary to the observations made in previous work, where LSA was found to significantly improve over the vector

Table 2. Precision, recall, F-measure and accuracy for finding the correct punch line

Model	Precision	Recall	F-measure	Accuracy
SEMANTIC RELATEDNESS				
Knowledge-based measures				
Leacock & Chodorow	0.28	0.34	0.31	0.61
Lesk	0.24	0.36	0.29	0.56
Resnik	0.24	0.35	0.28	0.56
Wu & Palmer	0.28	0.34	0.31	0.62
Lin	0.25	0.34	0.29	0.58
Jiang & Conrath	0.25	0.31	0.27	0.59
Corpus-based measures				
PMI	0.27	0.29	0.28	0.63
Vector space	0.26	0.61	0.37	0.48
LSA on BNC	0.20	0.25	0.22	0.56
Domain fitness				
Domain fitness	0.28	0.37	0.32	0.60
JOKE-SPECIFIC FEATURES				
Polysemy	0.32	0.33	0.32	0.66
Alliteration	0.29	0.75	0.42	0.48
LSA on joke corpus	0.75	0.75	**0.75**	**0.87**
COMBINED MODEL				
SVM	**0.84**	0.50	0.63	0.85
BASELINE				
Random choice	0.25	0.25	0.25	0.62

space model, here the opposite holds, with a much higher F-measure obtained using a simple measure of vector space similarity.

The models that perform best are those that rely on joke-specific features. The best results are obtained with the LSA model trained on the corpus of jokes, which exceeds by a large margin the baseline as well as the other models. This is perhaps due to the fact that this LSA model captures the "surprise" word associations that are frequently encountered in jokes.

The other joke-specific features also perform well. The simple verification of the amount of polysemy in a candidate punch line leads to a noticeable improvement above the random baseline, which confirms the hypothesis that humour is often relying on a large number of possible interpretations, corresponding to an increased word polysemy. The alliteration feature leads to a high recall, even if at the cost of low precision.

Finally, in line with the results obtained using the semantic relatedness of the set-up and the punch line, the fitness of domains is also resulting in an F-measure higher than the baseline, but a low overall accuracy.

Overall, perhaps not surprisingly, the highest precision is due to a combined model consisting of an SVM learning system trained on a combination of knowledge-based, corpus-based, and joke-specific features. During a ten-fold cross-validation run, the combined system leads to a precision of 84%, which is higher than the precision of any individual system, thus demonstrating the synergistic effect of the feature combination.

5 Conclusions

In this paper, we proposed and evaluated several computational models for incongruity detection in humour.

The paper made two important contributions. First, we introduced a new data set consisting of joke set-ups followed by several possible coherent continuations out of which only one had a comic effect. The data set helped us map the incongruity detection problem into a computational framework, and define the task as the automatic identification of the punch line among all the possible alternative interpretations. Moreover, the data set also enabled a principled evaluation of various computational models for incongruity detection.

Second, we explored and evaluated several measures of semantic relatedness, including knowledge-based and corpus-based measures, as well as other joke-specific features. The experiments suggested that the best results are obtained with models that rely on joke-specific features, and in particular with an LSA model trained on a corpus of jokes. Additionally, although the individual semantic relatedness measures brought only small improvements over a random-choice baseline, when combined with the joke-specific features, they lead to a model that has the highest overall precision of 84%, several order of magnitude better than the random baseline of 25%.

Acknowledgments

Rada Mihalcea's work was partially supported by the National Science Foundation under award #0917170. Carlo Strapparava was partially supported by the MUR

FIRB-project number RBIN045PXH. Stephen Pulman's work was partially supported by the Companions project (http://www.companions-project.org) sponsored by the European Commission as part of the Information Society Technologies programme under EC grant number IST-FP6-034434. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

References

1. Aristotle. Rhetoric. 350 BC
2. Attardo, S., Raskin, V.: Script theory revis(it)ed: Joke similarity and joke representation model. Humor: International Journal of Humor Research 4(3-4) (1991)
3. Berry, M.: Large-scale sparse singular value computations. International Journal of Supercomputer Applications 6(1) (1992)
4. Budanitsky, A., Hirst, G.: Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures. In: Proceedings of the NAACL Workshop on WordNet and Other Lexical Resources, Pittsburgh (2001)
5. Buitelaar, P., Magnini, B., Strapparava, C., Vossen, P.: Domain specific sense disambiguation. In: Edmonds, P., Agirre, E. (eds.) Word Sense Disambiguation: Algorithms, Applications, and Trends. Text, Speech and Language Technology, vol. 33, pp. 277–301. Springer, Heidelberg (2006)
6. Jiang, J., Conrath, D.: Semantic similarity based on corpus statistics and lexical taxonomy. In: Proceedings of the International Conference on Research in Computational Linguistics, Taiwan (1997)
7. Landauer, T.K., Foltz, P., Laham, D.: Introduction to latent semantic analysis. Discourse Processes 25 (1998)
8. Leacock, C., Chodorow, M.: Combining local context and WordNet sense similarity for word sense identification. In: WordNet, An Electronic Lexical Database. The MIT Press, Cambridge (1998)
9. Lesk, M.: Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In: Proceedings of the SIGDOC Conference 1986, Toronto (June 1986)
10. Lin, D.: An information-theoretic definition of similarity. In: Proceedings of the 15th International Conference on Machine Learning, Madison, WI (1998)
11. Magnini, B., Strapparava, C., Pezzulo, G., Gliozzo, A.: The role of domain information in word sense disambiguation. Natural Language Engineering 8(4), 359–373 (2002)
12. Mihalcea, R., Corley, C., Strapparava, C.: Corpus-based and knowledge-based approaches to text semantic similarity. In: Proceedings of the American Association for Artificial Intelligence, Boston, MA, pp. 775–780 (2006)
13. Mihalcea, R., Strapparava, C.: Making computers laugh: Investigations in automatic humor recognition. In: Proceedings of the Human Language Technology / Empirical Methods in Natural Language Processing conference, Vancouver (2005)
14. Mihalcea, R., Strapparava, C.: Learning to laugh (automatically): Computational models for humor recognition. Computational Intelligence 22(2), 126–142 (2006)
15. Patwardhan, S., Banerjee, S., Pedersen, T.: Using measures of semantic relatedness for word sense disambiguation. In: Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics, Mexico City (February 2003)

16. Raskin, V.: Semantic Mechanisms of Humor. Kluwer Academic Publications, Dordrecht (1985)
17. Resnik, P.: Using information content to evaluate semantic similarity. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence, Montreal, Canada (1995)
18. Ritchie, G.: Developing the incongruity-resolution theory. In: Proceedings of the AISB Symposium on Creative Language: Stories and Humour (1999)
19. Salton, G., Lesk, M.: Computer evaluation of indexing and text processing, pp. 143–180. Prentice Hall, Inc., Englewood Cliffs (1971)
20. Schopenhauer, A.: The World as Will and Idea. Kessinger Publishing Company (1819)
21. Stock, O., Strapparava, C.: Getting serious about the development of computational humor. In: Proceedings of the International Conference on Artificial Intelligence, IJCAI 2003 (2003)
22. Turney, P.: Mining the web for synonyms: PMI-IR versus LSA on TOEFL. In: Flach, P.A., De Raedt, L. (eds.) ECML 2001. LNCS (LNAI), vol. 2167, p. 491. Springer, Heidelberg (2001)
23. Wu, Z., Palmer, M.: Verb semantics and lexical selection. In: Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, Las Cruces, New Mexico (1994)

Emotions in Words: Developing a Multilingual WordNet-Affect

Victoria Bobicev, Victoria Maxim, Tatiana Prodan,
Natalia Burciu, and Victoria Angheluş

Technical University of Moldova,
168, Stefan cel Mare bd., Chisinau, Republic of Moldova
vika@rol.md, maxivica@yahoo.com, tatiana.ursulenco@gmail.com,
natusicb@yahoo.com, lazu_vic@yahoo.com

Abstract. In this paper we describe the process of Russian and Romanian WordNet-Affect creation. WordNet-Affect is a lexical resource created on the basis of the Princeton WordNet which contains information about the emotions that the words convey. It is organized in six basic emotions: *anger, disgust, fear, joy, sadness, surprise*.

We translated the WordNet-Affect synsets into Russian and Romanian and created an aligned English – Romanian – Russian lexical resource. The resource is freely available for research purposes.

Keywords: sentiment analysis, lexical representation of affects, multilingual lexical resources.

1 Introduction

Currently, the researchers in the field of the natural language processing drew their attention to the fact that texts contain not only objective information but also the emotional attitude of the author towards this information.

These days, the booming growth of Web 2.0 technologies allows every user to participate actively in web content creation (blogs, social networks, chats). The volumes of texts with emotionally-rich content grow in geometrical progression. This makes the subjective analysis of texts especially topical.

So far, the sentiment analysis and studies of the word affect were concentrated on English. The example is the SemEval-2007 task of Affective Text classification [6]. Most lexical resources have been created for English, as well. For example, SentiWordNet is a lexical resource for opinion mining which assigns to each synset of the WordNet three sentiment scores: positiveness, negativity, objectivity [11].

Recently, most of the Internet use growth was supported by non-native English speakers: starting 2000, for non-English speaking regions, the growth has surpassed 3,000% to compare with 342 % of the over-all growth.[1]

[1] http://www.internetworldstats.com/stats.htm

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 375–384, 2010.

Consequently, the amount of the text data written in languages other than English rapidly grows [3]. This raise increases the demand for automatic text analysis tools and linguistic resources for languages other than English. The tool development has progressed for Western European (French, German) and Asian (Japanese, Chinese, Arabic) [4].

At the moment, resources for Eastern European languages are not easily available. In order to fill this gap, we developed a linguistic resource, starting from WordNet-Affect, through the translation in Russian and Romanian languages, editing of the translated synsets and aligning them to English source.

2 WordNet-Affect

WordNet-Affect[2] is a well-used lexical resource which contains information about the emotions that the words convey. Compared with the complete WordNet, WordNet-Affect is a small lexical resource but valuable for its affective annotation.

WordNet-Affect [7] was created starting from WordNet DOMAINS [12]. WordNet-Affect produces an additional hierarchy of the *affective domain labels*, independent from the domain hierarchy, wherewith the synsets that represent affective concepts are annotated. The "affective words" are considered to be words that have "emotional connotation" [13]. There are words that not only describe directly some emotions (for example, *joy*, *sad* or *scare*) but also are related to emotions like words describing *mental state*s, *physical* or *bodily states, personality traits*, *behaviours*, *attitudes*, and *feelings* (such as *pleasure* or *pain*).

The collection of the WordNet-Affect synsets used in our work was provided as a resource for the SemEval-2007 "Affective Text". This task was focused on text annotation by affective tags [6]. There is not the whole WordNet-Affect but a part of it being more fine-grain re-annotated using six emotional category labels: *joy, fear, anger, sadness, disgust, surprise* [8]. This choice of the six emotions comes from psychological research into human non-verbally expressed emotions [5].

```
a#01943022 awed awestruck awestricken in_awe_of
```

Fig. 1. A synset of WordNet-Affect

The data is described in Table 1. The whole data is provided in six files named by the six emotions. Each file contains a list of synsets; one synset per one line. An example of a synset is shown in figure 1.

The first letter in the line indicates the part of speech; it is followed by the number of the synset and then all synset words are listed. The representation was simple and easy for further processing. There were a large number of word combinations, collocations and idioms. One of them can be seen in the example. These parts of synsets presented a problem during translation.

[2] For research purposes, WordNet-Affect is available upon request at http://wndomains.itc.it

Table 1. Data sets of affective words

Classes	#synsets	%synsets	#words	%words
anger	128	21.0	318	20.7
disgust	20	3.3	72	4.7
fear	83	13.5	208	13.5
joy	228	37.2	539	35.1
sadness	124	20.3	309	20.1
surprise	29	4.7	90	5.9
Total	612	100.0	1536	100.0

3 Development of Romanian and Russian WordNet

Romanian WordNet has been created by the Alexandru Ioan Cuza University of Iaşi during European project BalkaNet [9]. After the BalkaNet project ended, the Research Institute for Artificial Intelligence, at the Romanian Academy continued to update the Romanian WordNet and currently it contains 33151 noun synsets, 8929 verb synsets, 851 adjective synsets and 834 adverb synsets [10]. It can be accessed through the online MultiWordNet[3] interface where WordNets for several languages are aligned to Princeton WordNet.

First of all in our work we checked WordNet-Affect synsets using online interface of MultiWordNet. We just copied to our set all the synsets which already are in the Romanian WordNet and did not process these synset further. As result, 166 synsets were found in the Romanian WordNet, the majority of them being available for nouns and verbs. The adjectives and adverbs are less represented. The statistics of the already existing Romanian synsets is presented in table 2.

Table 2. Data sets of the already existing Romanian WordNet synsets

Classes	# synsets in WordNet-Affect	# synsets from Romanian WordNet	% synsets from Romanian WordNet
Anger	116	35	30.1
disgust	17	7	41.1
Fear	76	25	32.8
Joy	210	63	30.0
sadness	97	24	24.7
surprise	26	12	46.1
Total	542	166	30.6

There is completely different situation for Russian. Several attempts were taken to develop the Russian WordNet. RussNet is a project of computer thesaurus of Russian vocabulary [1]. An alternative project of Russian version of WordNet is Russian WordNet [2]. Both projects are non-commercial. Two commercial projects aimed to develop WordNets in Russian: RuThes is informational thesaurus used in UIS

[3] http://multiwordnet.itc.it/english/home.php

RUSSIA[4] and Russian WordNet project by the Novosoft company group[5]. Unfortunately, little information is available and even less freely available resources.

4 Development of Romanian and Russian WordNet-Affect

In order to create the two data sets, we applied the three-step approach: (1) automatic translation; (2) removing irrelevant translations; (3) generating Romanian and Russian synsets.

4.1 Automatic Translation

The translation was done automatically using bilingual dictionaries. We used Electronic Romanian-English Dictionary ROMEN from PRIMASOFT[6]. It consists of English-Romanian, Romanian-English, English-Russian and Russian-English parts, each containing more than 200 000 entries. In our work we used only the parts with English as a source language. There were a number of word combinations, collocations and idioms in the dictionaries which we have used in target languages. For the automatic translation, the dictionary was organized in a list of source words followed by the target translations. An example of the dictionary entry is presented in figure 2.

```
Joy
Dicţionar general:
noun: bucurie; confort; fericire; plăcere; tihnă;
veselie; voioşie;
verb: a bucura; a înveseli;
```

Fig. 2. An example of dictionary entry

At this stage, our goal was to obtain as many affective words as possible for the analysis. For this purpose we translated every word in the WordNet-Affect synsets. We decided to exclude from the English synsets all the word combinations, collocations and idioms as they could not be translated automatically. The figure 3 presents an example of the translated synset obtained after this step. As it is seen in the example, for the Romanian translation, we also obtained word combinations which were in the dictionary: "`cuprins de veneraţie`", "`cuprins de teamă`".

Some synset elements were not translated. These can be divided into four groups. (1) Word combinations, collocations and idioms which we intentionally removed from English synsets before the translation. (2) Spelling variations of the same word; for example, "`jubilance`", "`jubilancy`" – the first word was translated, the second one was not found in the dictionary. (3) Words which were formed using suffixes like "`ness`", "`less`", "`ful`" (for example "`heartlessness`"); these are

[4] http://www.cir.ru
[5] http://research-and-development.novosoft-us.com
[6] http://www.primasoft.biz/romen_eng.php

unlikely to appear in dictionaries as well as adverbs formed using suffix "ly". (4) Words which were not translated because of the limitedness of our dictionary. While WordNet can reasonably be mentioned as one of the largest English dictionary, our bilingual dictionary is fairly modest. Table 3 shows the percentage of words which were not translated. Average percentage of not translated words was 21%.

Table 3. Number and percentage of not translated words

Classes	# of English words	# of translated words	# of not translated words	% of not translated words
anger	318	248	70	22.0
disgust	72	60	13	18.0
fear	208	162	47	22.5
joy	539	420	119	22.0
sadness	309	246	63	20,5
surprise	90	72	18	21.0
Total	**1536**	**1208**	**330**	**21.0**

The second group of words did not present a problem but the first, third and the fourth ones had to be translated manually. It was done during the third step.

```
01943022 a:
awed =     speriat
awestruck =
           cuprins de venerație
           cuprins de teamă
awestricken = înspăimântat
```

Fig. 3. An example of English synset translation

4.2 Removing Irrelevant Translations

Many words in English synsets had several meanings. It was obvious that the automatic translation provided all possible translations for all their senses. We were interested in only one translation which was relevant to the synset meaning. The relevant translation was selected manually. We removed all translations whose meaning was not related to the emotion. For example, the word "taste" in the synset with the meaning "preference" had several meanings but only the last one in the list of possible translations was related to the synset common meaning. The example is demonstrated in figure 4. Thereby, we removed all translations except the last one.

As we translated every word separately, we obtained a lot of duplicates which had to be removed as well. We also watched over the part-of-speech correspondence. In many cases, it was rather difficult, especially for the already mentioned nouns formed using suffixes, for example, "plaintiveness" or "uncheerfulness".

```
05573914 n:
 preference =
           preferinţă
 penchant =
           înclinaţie
           slabiciune
 predilection =
           predilecţie
 taste =
           a avea gust
           a gusta, a cunoaşte
           a gusta; a degusta (un aliment)
           degustare
           fărâmă, bucăţică, îmbucătură (de)
           gust
           înclinaţie, preferinţă
```

Fig. 4. An example of one synset translation

4.3 Generating Romanian and Russian Synsets

All words in the synset represent one concept, one meaning. The aim of the third step was to find the adequate translation of exactly this meaning. At this step, we firstly had to attach English glosses to every synset. It made clearer the meaning of the synsets for translators. After the glosses were added to the synsets, the whole set was given to three translators which worked independently. Their task was twofold: (1) to remove the translations which, from their point of view, were irrelevant to the synset meaning described by the gloss; (2) to add as many relevant synonyms as possible to the Romanian and Russian synsets. Thereby, their task was to verify the equivalence of the English, Romanian and Russian synset meanings. They also had to translate the words which remained without translation from the first step. For translation they mostly used online dictionaries.

Bilingual English-Romanian dictionaries used:

- http://hallo.ro,
- http://dictionar.netflash.ro,
- http://www.ectaco.co.uk/English-Romanian-Dictionary;

Romanian thesaurus: http://dexonline.ro/.

Bilingual Russian dictionaries used:

- http://en.bab.la,
- http://dictionary.babylon.com,
- http://russianlessons.net/dictionary/dictionary.php;

Russian thesauri:

- http://slovo.freecopy.ru/,
- http://slovari.yandex.ru/dict/ushakov.

This step was the most laborious and difficult. Many English synsets have quite similar meaning with some nuances. In some cases, the synsets contained obsolete words, which were not found in the dictionary. As it was mentioned above, we tended to avoid word combinations, collocations and idioms. However, in some cases, the exact sense of the English synset could be represented only by some combination of Romanian or Russian words. In some cases even the English synset was presented by word combination. For example, `n#05591681 stage_fright.` Another example contains a German word: `n#05600844 world-weariness Weltschmerz`. In such cases, we did not obtain the proper translation. In some cases, several English synsets have got the same Romanian or Russian words as translations because we could not reflect the nuances of the source language senses in the target languages.

Referring to the problem with suffixes, for instance, the words "`weepiness`", "`dysphoria`", "`plaintiveness`", "`mournfulness`", "`ruthulness`" can hardly be found in dictionaries either in Romanian or English. In order to solve this problem, we searched the lemmas of the mentioned words in the available dictionaries. In this way, we could find the meaning of the words and, by adding the necessary affixes, the Romanian and Russian equivalents were created. For example, to find the adequate translation for the word "`mournfulness`", we searched in the dictionary the word "`mournful`". The result for Romanian is "`îndoliat`" and for Russian "`траурный`". As the word "`mournfulness`" is a noun, we transformed the obtained adjectives into nouns. Likewise, the Romanian equivalent is "`doliu`" and the Russian one is "`траур`".

However, most difficulties appeared with the alignment of adjectives. For example, for the emotional label "`sadness`", many of adjectival synsets translated in Russian contain the words "`грустный`" and "`печальный`". For different adjectival synsets we obtain quite similar translations as well.

4.4 Inter-translator Agreement

In our case, we could not use standard metrics for inter-translator agreement as we had the output as a set of synonyms. Therefore the agreement was calculated as follows. If **A** was a set of words selected by the first translator for the synset and **B** was a set of words selected by the second translator for the same synset, inter-annotator agreement **IntAgr** was equal to quotient of number of words in **A** and **B** intersection divided by number of words in **A** and **B** union:

$$\mathbf{IntAgr = A \cap B / A \cup B .} \quad (1)$$

For example, if one translator formed a synset from three words w_k, w_l and w_m and the second translator formed this synset from four words w_k, w_l, w_m and w_n and the first three words are the same, then A=(w_k w_l w_m), B=(w_k w_l w_m w_n), $A \cap B$ = (w_k w_l w_m), $\mathbf{A \cup B}$ = (w_k w_l w_m w_n), number of words in **A** and **B** intersection would be 3, number of words in **A** and **B** union would be 4 and therefore inter-translator agreement would be 3/4 = 0.75.

For example the synset "`a#01195320 friendly`" was translated by the first translator as "`prietenesc prietenos amical`", by the second translator as "`amical prietenos binevoitor`", and by the third as "`prietenesc`

prietenos binevoitor". For the first and the second translators the intersection of translations was two words: "prietenos amical" and translation's union were four words "prietenesc prietenos amical binevoitor". Inter-translator agreement in this case was 2/4=0.5. For the second and third translators the intersection of translations was two words: "prietenos binevoitor" and translation's union were four words "prietenesc prietenos amical binevoitor". Therefore, the agreement is the same: 0.5. For the first and third translators inter-translator agreement is again the same: 0.5. All three translators shared only one word "prietenos" and union of translations consisted from four words. Thus, the agreement was 1/4=0.25.

Table 4 presents the average values of the inter-translator agreement. The three translators are presented as T1, T2 and T3.

Table 4. Inter-translator agreement

Pair of translators	Inter-translator agreement
Russian data	
T1 – T2	0.57
T2 – T3	0.61
T1 – T3	0.59
All	0.29
Romanian data	
T1 – T2	0.58
T2 – T3	0.57
T1 – T3	0.67
All	0.32

As it is seen in the table, the agreement is low. There were some synsets with agreement equal to one as for example in the synset "a#00863650 euphoriant", all three translators translated it as "euforizant". However, for the majority of the synsets, the translators provided more different translations but not many of these translations were common for all translators. In some translated synsets, there was not any single word shared between all three translators. For example, for the synset "a#00670851 gladdened exhilarated", the three translations were "bucurat înveselit înviorat bine_dsipus", "bucuros vesel voios încântat bine_dispus" and "bucurat voios bucuros înveselit". There was no common word for all three translations.

Thus, we decided to form the synsets from words which were in at least two variants among the three translations. In such way, we formed the final synsets. For example, the synset "a#01195320 friendly" was translated as "prietenesc prietenos amical binevoitor" because all these words appeared at least twice in translations. The synset "a#00670851 gladdened exhilarated" was translated as "bucurat înveselit bine_dsipus bucuros voios".

Table 5 contains data on the final number of words in translations for each of the six WordNet-Affect emotions.

Table 5. Data sets of affective words for Russian and Romanian

Classes	#synsets	# Russian words	# Romanian words
anger	117	393	330
disgust	17	73	60
fear	80	327	248
joy	209	765	641
sadness	98	437	364
surprise	27	129	87
Total	548	2199	1869

It should be mentioned that in the source WordNet-Affect set there were some duplicated synsets. We removed all these repetitions and the number of synsets in our source is smaller. Besides, there were small differences in WordNet-Affect, MultiWordNet and online version of Wordnet because the MultiWordNet uses version 2.0 of WordNet and online version of WordNet is 3.0. It is seen that, despite of smaller number of synsets, the number of words in Romanian and Russian set is bigger than in English. This is due to our tendency to collect in our resource as many words as possible. We aim to use it in statistical methods of emotion recognition in text.

5 Conclusion and Future Work

This paper describes the process of the Russian and Romanian WordNet-Affect creation. WordNet-Affect is a lexical resource created on the basis of Princeton WordNet, which contains information about the emotions that the words convey. It is organized in six basic emotions: *anger, disgust, fear, joy, sadness, surprise*. WordNet-Affect is a small lexical resource but valuable for its affective annotation.

We translated the WordNet-Affect synsets into Russian and Romanian and, afterwards, created English – Romanian – Russian aligned WordNet-Affect. The resource can be used for the automatic recognition of emotions and affects in text. It is freely available for research purposes at http://lilu.fcim.utm.md.

The resource is still under development. The first version based on WordNet-Affect was released in August 2009; the second one, released in October 2009, is already aligned with the Romanian WordNet. Further, we are going to refine the Russian part and to create 'bag-of-words' resource for immediate use in emotion and affect recognition tasks. The resource has already been used in [14] and it is only one among many possible uses of the word sets.

References

1. 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 Standardization and How this affect Wordnet Applications and Evaluation, Las Palmas, pp. 60–64 (2002)

2. Balkova, V., Suhonogov, A., Yablonsky, S.A.: Russian WordNet. From UML-notation to Internet/Intranet Database Implementation. In: Second International WordNet Conference, GWC 2004, Brno, Czech Republic, pp. 31–38 (2004)
3. Crystal, D.: Language and The Internet. Cambridge University Press, Cambridge (2001)
4. Edmonds, P.: Introduction to Senseval. ELRA Newsletters 7(3), 337–344 (2002)
5. Ekman, P.: An argument for basic emotions. Cognition and Emotion 6(3-4), 169–200 (1992)
6. Strapparava, C., Mihalcea, R.: Learning to identify emotions in text. In: ACM Symposium on Applied Computing, Fortaleza, Brazil, pp. 1556–1560 (2008)
7. Strapparava, C., Valitutti, A.: Wordnet-affect: an affective extension of wordnet. In: 4th International Conference on Language Resources and Evaluation, pp. 1083–1086 (2004)
8. Strapparava, C., Valitutti, A., Stock, O.: The affective weight of the lexicon. In: 5th International Conference on Language Resources and Evaluation (LREC 2006), Genoa, Italy, pp. 474–481 (2006)
9. Tufis, D., Mititelu, B., Bozianu, L., Mihaila, C.: Romanian wordnet: New developments and applications. In: 3rd Conference of the Global WordNet Association, Korea, pp. 337–344 (2006)
10. Tufiş, D., Ion, R., Bozianu, L., Ceauşu, A., Ştefănescu, D.: Romanian Wordnet: Current State, New Applications and Prospects. In: 4th Global WordNet Conference, GWC 2008, pp. 441–452. University of Szeged, Hungary (2008)
11. Esuli, A., Sebastiani, F.: SentiWordNet: A Publicly Available Lexical Resource for Opinion Mining. In: 5th International Conference on Language Resources and Evaluation (LREC 2006), Genoa, Italy, pp. 417–422 (2006)
12. Magnini, B., Cavaglia, G.: Integrating subject field codes into wordnet. In: Second International Conference on Language Resources and Evaluation (LREC 2002), Athens, Greece, pp. 1413–1418 (2002)
13. Ortony, A., Clore, G.L., Foss, M.A.: The psychological foundations of the affective lexicon. Journal of Personality and Social Psychology 53, 751–766 (1987)
14. Sokolova, M., Bobicev, V.: Classification of Emotion Words in Russian and Romanian Languages. In: RANLP 2009 conference, Borovets, Bulgaria, pp. 415–419 (2009)

Emotion Holder for Emotional Verbs – The Role of Subject and Syntax

Dipankar Das and Sivaji Bandyopadhyay

Department of Computer Science and Engineering,
Jadavpur University, Kolkata, India
dipankar.dipnil2005@gmail.com, sivaji_cse_ju@yahoo.com

Abstract. Human-like holder plays an important role in identifying actual emotion expressed in text. This paper presents a baseline followed by syntactic approach for capturing emotion holders in the emotional sentences. The emotional verbs collected from WordNet Affect List (WAL) have been used in extracting the holder annotated emotional sentences from VerbNet. The baseline model is developed based on the *subject* information of the dependency-parsed emotional sentences. The unsupervised syntax based model is based on the relationship of the emotional verbs with their argument structure extracted from the *head* information of the chunks in the parsed sentences. Comparing the system extracted argument structure with available VerbNet frames' syntax for 942 emotional verbs, it has been observed that the model based on syntax outperforms the baseline model. The precision, recall and F-Score values for the baseline model are 63.21%, 66.54% and 64.83% and for the syntax based model are 68.11%, 65.89% and 66.98% respectively on a collection of 4,112 emotional sentences.

Keywords: Emotion Holder, VerbNet, Emotional Verb, Subject, Syntax, Arguments.

1 Introduction

In psychology and common use, emotion is an aspect of a person's mental state of being, normally based in or tied to the person's internal (physical) and external (social) sensory feeling [1]. The determination of emotion expressed in the text with respect to reader or writer is itself a challenging issue. Emotion holder extraction research is important for discriminating between emotions that are viewed from different perspectives [2]. A wide range of Natural Language Processing (NLP) tasks such as tracking users' emotion about products or events or about politics as expressed in online forums or news, to customer relationship management are using emotional information. So, the determination of emotion holder from the text invokes a challenge and helps us track and distinguish user's emotion separately.

In linguistics, a grammatical agent or holder is the participant of a situation that carries out the action and also, *agent* or *holder* is the name of the *thematic role*. The basic clue for identifying the emotion holder is the presence of any emotional verb or the appearance of non-emotional verb with any emotional phrases. The argument-based

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 385–393, 2010.

relationship of the emotional verbs with other component phrases in an emotional sentence gives the information to tag emotion holder both syntactically and semantically.

The present work aims to identify the emotion holder using two different approaches. A baseline system is developed based on the *subject* information of the emotional sentences parsed using Stanford Dependency Parser [3]. The *precision*, *recall* and *F-Score* values of the holder identification system are 63.21%, 66.54% and 64.83% respectively for the baseline approach.

Another way to identify emotion holder is based on the syntactical argument structure of the emotional sentences corresponding to the emotional verbs. Emotional verbs corresponding to Ekman's six different emotion types are retrieved from the WordNet Affect Lists (WAL) [15]. A total of 4,112 emotional sentences for these 942 emotional verbs have been extracted from the English VerbNet [4]. The holder related information as specified in the VerbNet such as *Experiencer, Agent, Actor, Beneficiary* etc. are properly tagged in the correct position of the syntactical frames for each sentence. All possible subcategorization frames and their corresponding syntaxes, available in the VerbNet are retrieved for each emotional verb. To achieve the objective, the *head* of each chunk is extracted from the dependency-parsed output. This chunk level information helps in constructing the syntactic argument structure with respect to the key emotional verb. The acquired syntactic argument structure is mapped to all the possible syntax structures present for each emotional verb in the VerbNet. If the syntactic argument structure of a sentence matches with any of the syntax structures extracted from the VerbNet for each emotional verb, the holder role associated with the VerbNet syntax is then assigned the holder tag in the appropriate component position of the syntactical arguments. Two separate techniques have been adopted for extracting the argument structure. One is from parsed result directly and another is from the corpus that has been POS tagged and chunked separately. The *precision* (P) and *recall* (R) values of these two techniques are 68.11 % (P), 63.04% (P) and 65.89 % (R), 64.34% (R) respectively on a collection of 4,112 emotional sentences. But, it has to be mentioned that the first technique gives significantly better *F-Score* value (66.98%) than the second one (62.39%) as the second one fails to disambiguate mostly the arguments from adjuncts. So, the dependency parser based method has been selected for emotion holder identification task. It has been observed that the baseline model suffers from the inability to identify emotion holder from the sentences containing passive senses. Although the *recall* value has been decreased in the syntactic model, it outperforms over the baseline model significantly in terms of *precision* and *F-Score*.

The rest of the paper is organized as follows. Section 2 describes the related works done in this area. The baseline system based on parsed data is described in Section 3. Two methods for developing syntax based model for holder identification is discussed in Section 4. Evaluation mechanism along with associated results is specified in Section 5. Finally Section 6 concludes the paper.

2 Related Work

Identification of opinion with its holder and topic from online media text using semantic role labeling is described in [6]. Importance of verb classes and linguistic features in classifying *polarity* and *subjectivity* are explained in [20]. Other related works are [7, 8] where they use the named entities to identify opinion holders.

An anaphor resolution based opinion holder identification method exploiting lexical and syntactic information from online news documents is narrated in [9]. Using generated features for each named entity and sentence pair, the machine learning based classification task for "not holder", "weak holder", "medium holder", or "strong holder" from the MPQA corpus is carried out in [10]. Identifying opinion holders for Question Answering in opinion text and the supporting annotation task are reported in [12].

The work on labeling the arguments of the verbs with their semantic roles using a novel frame matching technique is described in [21]. The present work is mostly related to the work described in [21]. But, irrespective of assignment of semantic roles, a technique has been designed to acquire argument structure of a sentence corresponding to the emotional verbs and map them on the frame syntax available in VerbNet for those verbs.

Based on the traditional perspectives, a new emotion holder model [11] is generated containing an emotion knowledge base for emotion expression followed by performing emotion reasoning algorithm and finally implementing the emotions treatment. The identification of the opinion propositions and their holders mostly for verbs is described in [13]. This work is similar to the present approach. But, the application of argument structure to identify emotion holder with respect to emotional verb in this present task signifies the difference from this approach. The comparative study of *subject* based holder identification task with syntax-based technique adopted in this present task is contributory to the platform of emotion holder identification.

3 Subject Based Baseline Model

The emotion holder present for an emotional verb in a sentence is crucial from the perspective of active and passive forms of the sentence. Before going into the detail exploration of the systems, the preparation of the holder annotated gold standard emotional corpus is first described. This is followed by the baseline methodology to extract holder information from parsed sentences.

3.1 Corpus Preparation

The sentiment lexicon, *SentiWordNet* [14] and emotion word lists like *WordNet Affect lists* (WALs) [15] are available in English. The English WALs, based on Ekman's [16] six emotion types are updated with the synsets retrieved from the English *SentiWordNet* to make adequate number of emotion word entries [17]. The list of verbs that have been collected from these six modified WALs, are termed as emotional verbs.

The enlisted emotional verbs are searched through the VerbNet classes. As the member verbs in any VerbNet class share the same syntactic and semantic information, the sentences described in that class is common for all members in that class. The sentences present in a VerbNet class and shared by different members are similar for an emotional verb if it is a member of that class and it can be considered that these sentences carry emotion due to the presence of such emotional verb(s).

If an emotional verb is found in any of the member verbs of any VerbNet class, the sentences corresponding to that class have been retrieved to construct the emotion corpus. The holder related tags (e.g. *Agent, Experiencer, Beneficiary* and *Actor*) are used to tag the retrieved sentences accordingly to prepare the gold standard holder annotated corpus. Table 1 shows the detailed statistics of the Emotion Corpus. Out of total 5,432 retrieved emotional sentences, 4,112 sentences are tagged with their holder related tags accordingly.

Table 1. Statistics of the Emotion Corpus

Information	# Total items found
Emotional Verbs in WordNet Affect list	1,278
Emotional verbs present in VerbNet	942
Retrieved Emotional Sentences	5,432
Annotated emotional sentences with *Holder* tag	3,156
Annotated sentences with other tag (*Experiencer*/ *Beneficiary* etc.)	956
Distinct VerbNet classes	523

3.2 Dependency Parsing and Subject Extraction

Stanford Parser [3], a probabilistic lexicalized parser containing 45 different part of speech (POS) tags of Pen Tree bank has been used to get the parsed sentences with dependency relations. The input emotional sentences are passed through the parser. The dependency relationships extracted from the parsed data are checked for predicates "*nsubj*" so that the *subject* related information in the "*nsubj*" predicate is considered as the probable candidate for identifying the emotion holder. Other dependency relations are filtered out from the parsed output. The present baseline system is developed based on the filtered subject information only. An example sentence is noted below whose parsed output and dependency relations are shown in Table 2. Here, the "*nsubj*" relations containing the emotional verb "grieve" tags "I" as an emotional holder.

*"I **grieve** for my departed Juliet."*

Table 2. Parsed Results

Parsed Output	*Dependency Relation*
(ROOT (S (**NP** (PRP I)) (**VP** (VBP grieve) (**PP** (IN **for**) (**NP** (PRP$ my) (JJ departed)(NN Juliet)))) (. .)))	***nsubj***(**grieve**-2, **I**-1) poss(Juliet-6, my-4) amod(Juliet-6, departed-5) prep_for(grieve-2, Juliet-6)

This baseline model is evaluated on the gold standard holder annotated emotional sentences that has been extracted from VerbNet. Total 4,112 sentences are evaluated and evaluation results are presented in Table 3 in terms of *precision*, *recall* and *F-Score*. It has been observed that the grammatical holders of the emotional sentences containing passive sense are often confused with the *subject* information. So, the next step is to explore the syntactical way for identifying argument structure of the sentences for their corresponding emotional verbs and to capture the emotion holder as a *thematic role* respectively.

4 Syntax Based Model

The syntax of a sentence is an important clue to capture the holder inscribed in text. More specifically, the argument structure or subcategorization information for a verb plays an essential role to identify the emotion holder from an emotional sentence. A subcategorization frame is a statement of what types of syntactic arguments a verb (or an adjective) takes, such as objects, infinitives, that-clauses, participial clauses, and subcategorized prepositional phrases [18]. VerbNet (VN) [4] is the largest online verb lexicon with explicitly stated syntactic and semantic information based on Levin's verb classification [23]. It is a hierarchical domain-independent, broad-coverage verb lexicon with mappings to other lexical resources such as WordNet [24], XTAG [25] and FrameNet [22]. Irrespective of other well-known lexical resources, VerbNet is used throughout this experiment as the main thrust for identifying the emotion holders is based on the characteristics of the emotional verbs only.

The existing syntax for each emotional verb is extracted from VerbNet and a separate rule based argument structure acquisition system is developed in the present task for identifying the emotion holder. The acquired argument structures are compared against the extracted VerbNet frame syntaxes. If the acquired argument structure matches with any of the extracted frame syntaxes, the emotion holder corresponding to each emotional verb is tagged with the holder information in the appropriate slot in the sentence.

4.1 Syntax Acquisition from VerbNet

VerbNet associates the semantics of a verb with its syntactic frames and combines traditional lexical semantic information such as thematic roles and semantic predicates, with syntactic frames and selectional restrictions. Verb entries in the same VerbNet class share common syntactic frames, and thus they are believed to have the same syntactic behavior. The VerbNet files containing the verbs with their possible subcategorization frames and membership information are stored in XML file format. E.g. the emotional verbs "*love*" and "*enjoy*" are member of the *admire-31.2-1* class and "*enjoy*" also belongs to the class *want-32.1-1*. A snapshot of the XML file for the *admire-31.2-1* class is given below.

…<VNCLASSID="admire-31.2"
… <SUBCLASSES>….
<VNSUBCLASS ID="admire-31.2-1">
<MEMBERS>….
<MEMBER name="**love**" wn="love%2:37:00 love%2:37:02 love%2:37:01"/>
<MEMBER name="**enjoy**" wn="enjoy%2:37:00 enjoy%2:37:01 enjoy%2:34:00"/>…..
<THEMROLES/> <FRAMES>
<FRAME> <DESCRIPTION descriptionNumber="8.1" primary="TO-INF-SC" secondary="" xtag="0.1"/> …. <EXAMPLE>I loved to write.</EXAMPLE>
<SYNTAX> <NP value="Experiencer"> <SYNRESTRS/> </NP> <VERB/> <NP value="Theme">
<SEMANTICS> <PRED value="**emotional_state**">
<ARGS> <ARG type="Event" value="E"/> <ARG type="**VerbSpecific**" value="**Emotion**"/> <ARG type="ThemRole" value="Experiencer"/> …..
</ARGS> </PRED> </SEMANTICS> </FRAME>…..

The XML files of VerbNet are preprocessed to build up a general list that contains all member verbs and their available syntax information retrieved from VerbNet. This preprocessed list is searched to acquire the syntactical frames for each emotional verb. One of the main criteria considered for selecting the frames is the presence of "*emotional_state*" type predicate associated with the frame semantics.

4.2 Argument Structure Acquisition Framework

To acquire the argument structure for a sentence, two separate approaches, Methods A and B, have been used, one (Method A) is from the parsed result directly and another (Method B) is from the POS tagged and chunked sentences accordingly.

The parsed emotional sentences are passed through a rule based *phrasal-head* extraction process to identify the phrase level argument structure of the sentences corresponding to the emotional verbs. The extracted *head part* of every phrase from the well-structured bracketed parsed data is considered as the component of the argument structure. For example, the *head* parts of the phrases are extracted to make the phrase level pattern or argument structures of the following sentences.

Sentence1: "*Caesar killed Brutus with a knife.*"
Parsed Output:

(ROOT (S (**NP** (NNP Caesar)) (**VP** (VBD killed) (**NP** (NNS Brutus)) (**PP** (IN **with**) (NP (DT a) (NN knife)))) (. .)))

Acquired Argument Structure: [NP VP NP PP-with]
Simplified Extracted VerbNet Frame Syntax: [<NP value="*Holder*"> <VERB/> <NP-*patient*> <PREP value="*with*">]

Sentence2: "***I love*** *everybody*."
Parsed Output:

```
(ROOT (S (NP (PRP I))(VP (VBP love))(NP (NN everybody))) (. .))
```

Acquired Argument Structure: [NP VP NP]
Simplified Extracted VerbNet Frame Syntax: [<NP value="*Experiencer*" ></VERB><NP-*theme*>]

Sentence3: "*The children liked that the clown had a red nose.*"
Parsed Output:

(ROOT (S (**NP** (DT The) (NNS children)) (**VP** (VBD liked) (**SBAR** (IN **that**)
(S (NP (DT the) (NN clown)) (VP (VBD had) (NP (DT a) (JJ red) (NN nose)))))) (. .)))

Acquired Argument Structure: [NP VP SBAR-that]
Simplified Extracted VerbNet Frame Syntax: [<NP value="*Experiencer*"><VERB/> <NP-theme><SYNRESTR type="*that*_comp"/>

It is to be mentioned that, the phrases headed by "S" (sentential complement), "PP" (Preposition Phrase), "NP" (Noun Phrase) and followed by the emotional verb phrase contribute in structuring the syntactical argument. One tag conversion routine has been developed to transform the POS information of the system-generated argument structure for comparison with the POS categories of the VerbNet syntax. It has been observed that the phrases that start with ADJP, ADVP (adjective, adverbial phrases) tags generally do not contribute towards valid argument selection strategy. But, the entities in the slots of active frame elements are added if they construct a frame that matches with any of the extracted frames from VerbNet. The *head* part of each phrase with its component attributes (e.g. "*with*" component attribute for "PP" phrase) in the parsed result helps in identifying the maximum matching possibilities.

Another alternative way to identify the argument structure from a sentence is carried out based on the POS tagged and chunked data. The emotional sentences are tagged with an open source Stanford Maximum Entropy based POS tagger [5]. The best reported accuracy for the POS tagger on the Penn Treebank is 96.86% overall and 86.91% on previously unseen words. The POS tagged sentences are passed through a Conditional Random Field (CRF) based chunker [19] to acquire chunked data where each component of the chunk is marked with *beginning* or *intermediate* or *end* corresponding to the elements slot in that chunk. The POS of the *beginning* part of every chunk has been extracted and frames have been developed to construct the argument structure of the sentence corresponding to the emotional verb. The acquired argument structure of a sentence is mapped to all of the extracted VerbNet frames. If a single match is found, the slot devoted for the holder in VerbNet frame is used to tag in the appropriate slot in the acquired frame. For example, the argument structure acquired from the following chunked sentence is "NP-VP-NP".

```
I/PRP/B-NP love/VBP/B-VP them/PRP/B-NP ././O
```

But, it has been observed that this second system suffers from the inability to recognize arguments from adjuncts as the system blindly captures *beginning* parts as arguments whereas they are adjuncts in real. So, this system is biased to the *beginning* chunk.

5 Evaluation

The evaluation of the baseline system is straightforward. The emotion holder annotated sentences are extracted from the VerbNet and the sentences are passed through the baseline system to annotate the sentences with their *subject* based holder tag accordingly. A total of 4,112 sentences are evaluated and the *precision*, *recall* and *F-Score* values are shown in Table 3. It is observed that the *subject* information helps in identifying emotion holder with high *recall*. But, the holder identification task for passive sentences fails in this baseline method and hence there is a fall in *precision* value. Two types of unsupervised rule based methods have been adopted to acquire the argument structure from the emotional sentences. It has been observed that, the Method-A that acquires argument structure from parsed result directly outperforms the Method-B that acquires these structures from POS tagged and chunked data. The *recall* value has decreased in Method-B as it fails to distinguish the arguments from the adjuncts. The emotion holder identification system based on argument structure directly from parsed output gives satisfactory performance.

Table 3. Precision, Recall and F-Score values of the Baseline and Syntactic model

Type	*Baseline Model* (in %)	*Syntactic Model* (in %)	
		Method-A	Method-B
Precision	63.21	68.11	63.05
Recall	66.54	65.89	64.34
F-Score	64.83	66.98	62.39

6 Conclusion

In this work, the emotion holder identification task is carried out based on the roles associated to *subject* information. The syntactic way of developing the holder extraction module by focusing on the role of arguments of the emotional verbs improves the result significantly. Further works need to be on sentences with non-emotional verbs but emotional phrases. The holder-annotated corpus preparation from VerbNet especially for emotional verbs followed by the argument extraction module can be further explored through the help of machine learning approach.

References

1. Zhang, Y., Li, Z., Ren, F., Kuroiwa, S.: A Preliminary Research of Chinese Emotion Classification Model. IJCSNS International Journal of Computer Science and Network Security 8(11), 127–132 (2008)
2. Seki, Y.: Opinion Holder Extraction from Author and Authority Viewpoints. In: SIGIR 2007. ACM, New York (2007), 978-1-59593-597-7/07/0007
3. de Marneffe, M.-C., MacCartney, B., Manning, C.D.: Generating Typed Dependency Parses from Phrase Structure Parses. In: 5th International Conference on Language Resources and Evaluation (2006)

4. Kipper-Schuler, K.: VerbNet: A broad-coverage, comprehensive verb lexicon. Ph.D. thesis, Computer and Information Science Dept., University of Pennsylvania, Philadelphia, PA (2005)
5. Manning, C.D., Toutanova, K.: Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger. In: Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, EMNLP/VLC (2000)
6. Kim, S.-M., Hovy, E.: Extracting Opinions, Opinion Holders, and Topics Expressed in Online News Media Text. ACL (2006)
7. Kim, S.-M., Hovy, E.: Identifying Opinion Holders for Question Answering in Opinion Texts. In: Proceedings of AAAI 2005 Workshop on Question Answering in Restricted Domains (2005)
8. Choi, Y., Cardie, C., Riloff, E., Patwardhan, S.: Identifying Sources of Opinions with Conditional Random Fields and Extraction Patterns. In: Proceedings of HLT/EMNLP 2005 (2005)
9. Kim, Y., Jung, Y., Myaeng, S.-H.: Identifying Opinion Holders in Opinion Text from Online Newspapers. In: 2007 IEEE International Conference on Granular Computing, pp. 699–702 (2007), doi:10.1109/GrC.2007.45
10. Evans, D.K.: A low-resources approach to Opinion Analysis: Machine Learning and Simple Approaches. NTCIR (2007)
11. Hu, J., Guan, C., Wang, M., Lin, F.: Model of Emotional Holder. In: Shi, Z.-Z., Sadananda, R. (eds.) PRIMA 2006. LNCS (LNAI), vol. 4088, pp. 534–539. Springer, Heidelberg (2006)
12. Wiebe, J., Wilson, T., Cardie, C.: Annotating expressions of opinions and emotions in language. Language Resources and Evaluation 1(2) (2005)
13. Bethard, S., Yu, H., Thornton, A., Hatzivassiloglou, V., Jurafsky, D.: Automatic Extraction of Opinion Propositions and their Holders. In: AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications (2004)
14. Esuli, A., Sebastiani, F.: SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining. In: LREC 2006 (2006)
15. Strapparava, C., Valitutti, A.: WordNet-Affect: an affective extension of WordNet. In: Proceedings of the 4th International Conference on Language Resources and Evaluation, pp. 1083–1086 (2004)
16. Ekman, P.: Facial expression and emotion. American Psychologist 48(4), 384–392 (1993)
17. Das, D., Bandyopadhyay, S.: Sentence Level Emotion Tagging. In: ACII 2009. IEEE, Los Alamitos (2009)
18. Manning, C.D.: Automatic Acquisition of a Large Subcategorization Dictionary from Corpora. In: 31st Meeting of the ACL, Columbus, Ohio, pp. 235–242 (1993)
19. Phan, X.-H.: CRFChunker: CRF English Phrase Chunker. In: PACLIC 2006 (2006)
20. Vincent, B., Xu, L., Chesley, P., Srhari, R.K.: Using verbs and adjectives to automatically classify blog sentiment. In: Proceedings of AAAI-CAAW 2006, the Spring Symposia (2006)
21. Swier, R.S., Stevenson, S.: Unsupervised Semantic Role Labelling. In: Proceedings of EMNLP (2004)
22. Baker, C.F., Fillmore, C.J., Lowe, J.B.: The Berkeley FrameNet project. In: COLING/ACL 1998, Montreal, pp. 86–90 (1998)
23. Levin, B.: English Verb Classes and Alternation: A Preliminary Investigation. The University of Chicago Press, Chicago (1993)
24. Miller, G.A.: WordNet: An on-line lexical database. International Journal of Lexicography 3(4), 235–312 (1990)
25. XTAG Research Group.: A lexicalized tree adjoining grammar for English. IRCS. University of Pennsylvania (2001)

A Chunk-Driven Bootstrapping Approach to Extracting Translation Patterns

Lieve Macken[1,2] and Walter Daelemans[3]

[1] LT3, University College Ghent,
Groot-Brittanniëlaan 45, Ghent, Belgium
[2] Dept. of Applied Mathematics and Computer Science, Ghent University,
Krijgslaan 281(S9), Ghent, Belgium
[3] CLiPS Computational Linguistics Group, University of Antwerp
Prinsstraat 13, 2000 Antwerpen, Belgium

Abstract. We present a linguistically-motivated sub-sentential alignment system that extends the intersected IBM Model 4 word alignments. The alignment system is chunk-driven and requires only shallow linguistic processing tools for the source and the target languages, i.e. part-of-speech taggers and chunkers.

We conceive the sub-sentential aligner as a cascaded model consisting of two phases. In the first phase, anchor chunks are linked based on the intersected word alignments and syntactic similarity. In the second phase, we use a bootstrapping approach to extract more complex translation patterns.

The results show an overall AER reduction and competitive F-Measures in comparison to the commonly used symmetrized IBM Model 4 predictions (intersection, union and grow-diag-final) on six different text types for English-Dutch. More in particular, in comparison with the intersected word alignments, the proposed method improves recall, without sacrificing precision. Moreover, the system is able to align discontiguous chunks, which frequently occur in Dutch.

Keywords: chunk alignment, word alignment, parallel corpora, computer-aided translation.

1 Introduction

Sub-sentential alignments are used among other things to create phrase tables for statistical phrase-based machine translation systems. A stand-alone sub-sentential alignment module however, is also useful for human translators if incorporated in CAT-tools, e.g. in sub-sentential translation memory systems [1], or for bilingual terminology extraction [2,3].

In the context of statistical machine translation, GIZA++ [4] is one of the most widely used word alignment toolkits. GIZA++ implements the IBM models [5] and is used in Moses [6] to generate the initial source-to-target and target-to-source word alignments after which a symmetrization heuristic combines the

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 394–405, 2010.

alignments of both translation directions. *Intersecting* the two alignments results in an overall alignment with a high precision, while taking the *union* of the alignments results in an overall alignment with a high recall. The default symmetrization heuristic applied in Moses (*grow-diag-final*) starts from the intersection points and gradually adds alignment points of the union to link unaligned words that neighbour established alignment points. The main problem with the *union* and the *grow-diag-final* heuristics is that the gain in recall causes a substantial loss in precision, which poses a problem for applications intended for human users.

A considerable amount of research has been devoted to the topic of improving the accuracy of word and phrase alignment models. Ganchev et al. [7] also favour the idea of using intersected word alignments *by encouraging the models to agree by training them concurrently*, rather than training the alignment models in two directions and combining their predictions. Zhang et al. [8] unify the training of the word and phrase alignment models. In their staged training procedure, they first train a word alignment model and use the confident word links to reduce the phrasal alignment space. We also use a staged training procedure starting from confident word links, but in our alignment system, we use linguistic constraints to align linguistically-motivated chunks.

Several researchers demonstrated that the addition of linguistic information can improve statistically-based word alignment systems. DeNero and Klein [9] use a syntax-aware distortion component to improve the word alignments. Tiedemann [10] combines association measures with additional linguistic heuristics based on part-of-speech, phrase type, and string similarity measures. While Tiedemann makes use of chunk information, the alignment process remains word-based. In our approach, the alignment process is primarily chunk-driven.

2 Architecture

The global architecture of our system is visualized in Figure 1. The sub-sentential alignment system takes as its input sentence-aligned texts, together with additional linguistic annotations (part-of-speech codes and chunk information) for the source and the target texts along with the intersected word alignments generated by the GIZA++ toolkit. The system stores all this information in a lexical link matrix.

The sub-sentential alignment system itself is conceived as a cascaded model consisting of two phases. The objective of the first phase is to link *anchor chunks*, i.e. chunks that can be linked with a very high precision. Those anchor chunks are linked based on the intersected word alignments and syntactic similarity. In the second phase, we use a bootstrapping approach to extract language-pair specific translation rules. The anchor chunks and the word alignments of the first phase are used to limit the search space in the second phase.

Although the global architecture of our sub-sentential alignment system is language-independent, some language-specific resources are used. The system

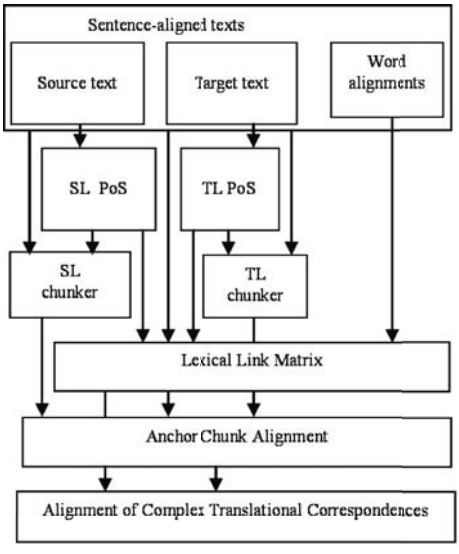


Fig. 1. Outline architecture

requires shallow linguistic processing tools for the source and the target languages, i.e. part-of-speech taggers and chunkers. We focus on the English-Dutch language pair.

2.1 Additional Linguistic Annotations

Part-of-speech tagging for English was performed by the memory-based PoS tagger MBT, which is part of the MBSP tools [11]. Part-of-speech tagging for Dutch was performed by TADPOLE [12].

We further enriched the corpora with chunk information. During text chunking, syntactically related words are combined into non-overlapping chunks based on PoS information [13]. We developed rule-based chunkers for English and Dutch. The rule-based chunkers contain distituency rules, i.e. the rules add a chunk boundary when two part-of-speech codes cannot occur in the same constituent. The following example shows an English-Dutch sentence pair divided in non-overlapping chunks:

En: It | is | a complicated | and | difficult problem
Nl: Het | is | een gecompliceerd | en | moeilijk probleem

2.2 Anchor Chunk Alignment

The basic idea behind our approach is that – at least for European languages – translations conveying the same meaning use to a certain extent the same building blocks from which this meaning is composed: i.e. we assume that to a

large extent noun and prepositional phrases, verb phrases and adverbial phrases in one language directly map to similar constituents in the other language. The extent to which our basic assumption holds depends on the translation strategy that was used. Text types that are typically translated in a more literal way (e.g. technical texts) will contain more direct correspondences than text types for which a more free translation strategy was adopted (e.g. journalistic texts).

In the first phase of this system, *anchor chunks* are linked, i.e. chunks that can be linked with a very high precision. Chunks are considered to be anchor chunks if all words of source and target chunk(s) are either linked by means of a lexical link or can be linked on the basis of corresponding part-of-speech codes.

2.3 Alignment of Complex Translational Correspondences

In the second phase, we use a bootstrapping approach to align more complex translational correspondences. We start from a sentence-aligned parallel corpus of short sentences in which anchor chunks have been aligned on the basis of the intersected GIZA++ alignments and syntactic similarity.

The bootstrapping process is a cyclic process which alternates between extracting candidate translation rules (extraction step) and scoring and filtering the extracted candidate translation rules (validation step). From the second bootstrapping cycle onwards, the validated translation rules are first applied to the corpus, after which the extraction process is launched again. The bootstrapping process is repeated four times.

2.4 Extraction Step

In the extraction step, candidate translation rules are extracted from unlinked source and target chunks. Different alignment types (1:1, 1:n, n:1 and n:m) are considered:

- From sentence pairs that only contain 1:1, 1:n and n:1 unlinked chunks, candidate translation rules that link 1:1, 1:n and n:1 chunks are extracted. In the left example of Figure 2, the source chunk *membership* and target chunk *het lidmaatschap* are selected because they are the only unlinked chunks in the sentence pair.
- From sentence pairs in which the only unlinked chunks in the source or target sentence are lexically interlinked, candidate translation rules that link n:m chunks are extracted. In the right example of Figure 2, the source chunks *not just | an old person's disease* and the target chunks *geen ziekte | die | alleen ouderen |treft* [En: *no disease that just elderly strikes*] are selected, as the source chunks *not just* and *an old person's disease* are lexically interlinked and are the only unlinked chunks in the source sentence.

From the selected source and target chunks two types of rules are extracted: *abstract* rules and *lexicalized* rules. The rules can be contiguous or non-contiguous.

– Abstract rules are coded as PoS sequences. Established word alignments within the extracted chunks are coded as indices. For example the rule DET+N-gen+N_1 → DET+N_1|PREP|DET+N captures the transformation of a genitive into a prepositional phrase as in *the public's right* → *het recht van de burgers* [En: *the right of the public*]
– Lexicalized rules are coded as token sequences, e.g. *to treat* → *ter behandeling van* [En: *for the treatment of*]

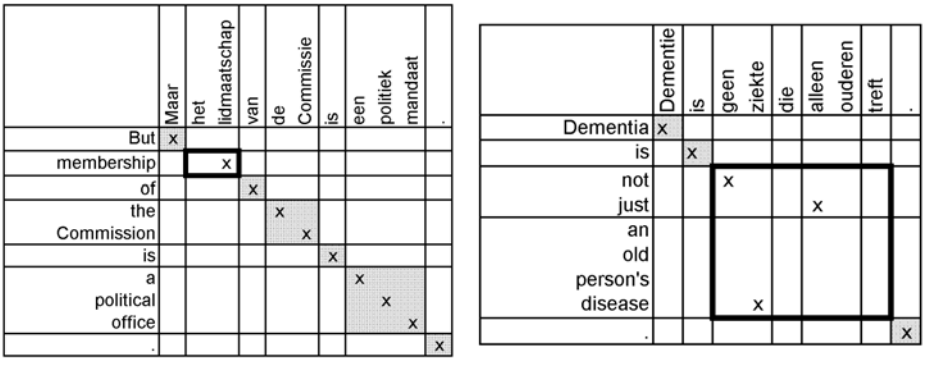


Fig. 2. Sentence pair with one unlinked source (*membership*) and target chunk (*het lidmaatschap [En: the membership]*) and sentence pair with unlinked source chunks that are grouped by means of lexical links. Chunk boundaries are indicated by horizontal and vertical lines, intersected IBM Model 4 word alignments by x's, and anchor chunks in light grey.

From the second bootstrapping cycle onwards, the validated rules are first applied to the whole training corpus, resulting in new translation pairs containing 1:1, 1:n and n:1 unlinked chunks, after which the extraction process is launched again.

The matching process considers all lexically interlinked groups of chunks (see the right example of Figure 2) and all unlinked source and target chunks with a neighbouring left or right anchor chunk and uses the word aligments and the anchor chunks to build up the target or source pattern.

2.5 Validation Step

The aim of the validation step is twofold. On the one hand, we want to extract a subset of *reliable* translation rules out of the set of candidate translation rules. On the other hand, we want to sort the translation rules to determine the order in which the rules are applied.

We use the Log-Likelihood Ratio as statistical association measure to compute an association score between each source and target pattern of all candidate translation rules. The Log-Likelihood ratio has been used before for building translation dictionaries [14] and for word alignment [15].

To compute the Log-Likelihood Ratio, we first count for each candidate translation rule how many times the source and target pattern co-occur in the corpus and calculate the Log-Likelihood on the basis of the observed frequencies and the expected frequencies under the null hypothesis of independence as follows:

$$-2log(\lambda) = 2\sum_{ij} O_{ij} log(\frac{O_{ij}}{E_{ij}}). \quad (1)$$

Dunning [16] showed that the Log-Likelihood ratio test allows comparisons to be made between the significance of the occurrences of both rare and common phenomena, which makes the test appropriate for our purposes. According to Manning and Schütze [17], $-2log(\lambda)$ has a distribution similar to that of chi-square can thus be used for hypothesis testing using the statistical tables for the distribution of chi-square. For a contingency table with two rows and two columns the critical value is 10.83 for the significance level of 0.001 [18].

Therefore, in the validation step, we only retain translation rules with a Log-Likelihood value higher than 10.8. To reduce the memory requirements of our system, we only validated candidate translation rules that co-occurred at least 5 times.

3 Experimental Results

3.1 Bootstrapping Corpus

For the extraction and the validation step of the bootstrapping process we extracted two subcorpora from the Dutch Parallel Corpus [19].

- The first subcorpus contains 36,406 sentence pairs (478,002 words) of short sentences (1-10 words).
- The second subcorpus contains 79,814 sentence pairs (1,892,233 words) of medium-length sentences (1-20 words).

The Dutch Parallel Corpus has a balanced composition and contains five text types: administrative texts, texts treating external communication, literary texts, journalistic texts and instructive texts. All text types are present in the selected subcorpora.

3.2 Reference Corpus

In order to evaluate the bootstrapping approach, a manual reference corpus was created that includes six different text types: journalistic texts, proceedings of plenary debates (selected from Europarl), financial newsletters, press releases, technical texts of the medical domain, and user manuals[1]. The formal characteristics of the reference corpus are presented in Table 1.

[1] The largest part of the manual reference corpus is publicly available as part of the Dutch Parallel Corpus, which is distributed by the Agency for Human Language Technologies (TST-centrale).

We assume that for each of the three text types another translation style was adopted, with the journalistic texts being the most free and the technical texts and user manuals being the most literal translations.

In the manual reference corpus, different units could be linked (words, word groups, paraphrased sections, punctuation). Three different types of links were used: regular links for straightforward correspondences, fuzzy links for translation-specific shifts of various kinds, and null links for words for which no correspondence could be indicated (deletions or additions).

To evaluate the system's performance, the links created by the system were compared with the links of the manual reference files.

Table 1. En-Nl Test data

Text type	# Words	# Sentences	# Texts
Journalistic texts	8,557	177	3
Proceedings EP	3,139	105	7
Newsletters	12,000	344	2
Press Releases	4,926	212	4
Technical texts	8,661	216	4
User Manuals	4,010	296	2
Total	41,293	1,350	22

To be able to compare the alignments of the system with the reference alignments, all phrase-to-phrase alignments were converted into word-to-word alignments by linking each word of the source phrase to each word of the target phrase (all-pairs heuristic).

3.3 Evaluation Metrics

The evaluation of word alignment systems is not a trivial task. Different evaluation metrics exist, and they mainly differ in the way divergent translational correspondences are treated. Given the controversy in the literature, we evaluated our system with two different metrics: Alignment Error Rate (AER) and a weighted version of F-Measure.

Alignment Error Rate. Alignment error rate was introduced by Och and Ney [4] to evaluate the performance of word alignment systems. They distinguished *sure* alignments (S) and *possible* alignments (P) and introduced the following redefined precision and recall measures (where A refers to the set of alignments):

$$precision = \frac{|A \cap P|}{|A|}, recall = \frac{|A \cap S|}{|S|}. \tag{2}$$

and the alignment error rate (AER):

$$AER(S, P; A) = 1 - \frac{|A \cap P| + |A \cap S|}{|A| + |S|}. \tag{3}$$

The distinction between *sure* and *possible* alignments approximately corresponds to the distinction between regular and fuzzy links in our annotation scheme. Therefore we consider all regular links of the manual reference as *sure* alignments and all fuzzy and null links as *possible* alignments to compare the output of our system with the manual reference.

Weighted F-Measure. F-Measure combines the traditional precision and recall metrics and can be calculated on all word-to-word links. However, Melamed [20] pointed out that F-Measure poses a problem. If precision and recall is calculated on all word-to-word links, all links would be equally important and would place undue importance on words that were linked more than once (e.g. all word-to-word links resulting from the phrasal alignments). Therefore, Melamed introduced a weighted version of precision and recall in which a weight is assigned to each word-to-word link.

We use the weighting method developed by Davis [21], which is a refinement of Melamed's weighting principles. In this weighting scheme, every word contributes 0.5 to the total weight. In case of interlinked word-to-word links from the phrasal alignments, each link is assigned the total weight of the phrasal alignment divided by the number of word-to-word links. Precision and recall are then calculated on the normalized weights.

3.4 Results

The results of all our experiments are summarized in Tables 2 and 3. In Table 2, we give per text type the alignment scores for the symmetrized IBM Model 4 predictions, using the three most commonly used symmetrization heuristics: intersection ($\cap$), union ($\cup$), and grow-diag-final (Gdf). As expected, the intersection heuristic generates the most precise overall alignment, while the union results in an alignment with the highest recall. The recall gain in the *union* and *grow-diag-final* heuristics causes a substantial loss in precision.

In Table 3, the results of our chunk-based extension to the intersected IBM Model 4 alignments are given for four different settings:

- **10Lex:** bootstrapping corpus of short sentences (1-10 words); only lexicalized translation rules or abstract rules containing lexical indices are retained in the validation step
- **10All:** bootstrapping corpus of short sentences (1-10 words), all lexicalized translation rules or abstract rules containing lexical indices are applied first; in a second step abstract rules without lexical clues are applied
- **20Lex:** identical to 10Lex but a bootstrapping corpus of medium-length sentences (1-20 words) is used
- **20All:** identical to 10All but a bootstrapping corpus of medium-length sentences (1-20 words) is used

The results reflect the different translation strategies of the different text types: the technical texts are the easiest to align; the journalistic and Europarl texts

Table 2. Results for the different symmetrized IBM Model 4 predictions: intersection (∩), union (∪), and grow-diag-final (Gdf) expressed in terms of AER and weighted F-measure

	Journalistic			Europarl			Newsletters		
	∩	∪	GDF	∩	∪	GDF	∩	∪	GDF
Prec	95.7	57.9	62.0	94.1	73.7	76.1	96.4	72.3	76.3
Rec	65.5	84.2	83.7	64.0	80.0	79.1	65.4	84.7	83.9
AER	21.8	32.4	29.6	22.9	23.6	22.6	21.8	22.3	20.3
WPrec	95.7	58.8	62.9	94.1	75.7	77.9	96.4	72.5	76.5
WRec	51.5	67.6	67.0	51.1	64.9	63.9	58.9	75.5	74.7
WF1	67.0	62.9	64.9	66.2	69.9	**70.2**	73.1	74.0	75.6
	Pressreleases			Technical			User Manuals		
Prec	98.6	76.2	80.7	97.8	78.0	81.3	97.8	73.2	77.8
Rec	63.3	76.3	75.5	73.2	88.0	87.4	64.1	83.4	82.5
AER	22.7	23.8	21.9	16.1	17.5	15.9	22.3	22.3	20.0
WPrec	98.6	77.3	81.7	97.8	78.5	81.8	97.8	74.1	78.7
WRec	64.4	76.3	75.8	68.1	80.9	80.3	61.0	78.1	77.5
WF1	77.9	76.8	78.6	80.3	79.7	81.1	75.1	76.1	78.1

Table 3. Results for the chunk-based extension to the intersected IBM Model 4 alignments for four different settings expressed in terms of AER and weighted F-measure

	Journalistic				Europarl				Newsletters			
	10Lex	10All	20Lex	20All	10Lex	10All	20Lex	20All	10Lex	10All	20Lex	20All
Prec	94.0	93.0	92.4	92.0	93.8	92.4	93.4	92.0	96.0	94.9	94.6	94.6
Rec	70.1	71.9	70.6	73.1	67.8	68.4	68.7	69.2	69.8	71.8	70.2	72.4
AER	19.3	18.6	19.7	**18.2**	20.4	20.5	**19.9**	20.1	8.9	18.0	19.1	**17.7**
WPrec	94.1	93.1	93.3	92.1	93.9	92.8	93.5	92.4	95.9	94.9	95.8	94.6
WRec	55.0	56.6	55.7	57.6	54.1	54.8	54.8	55.4	62.4	64.3	63.1	64.8
WF1	69.4	70.4	69.8	**70.9**	68.6	68.9	69.1	69.3	75.7	76.7	76.1	**76.9**
	Pressreleases				Technical				User Manuals			
Prec	98.2	97.6	97.7	96.8	97.2	96.4	96.1	96.3	96.6	96.4	96.3	95.7
Rec	65.3	66.4	65.8	66.9	76.3	77.7	77.3	78.4	68.0	69.9	68.8	70.8
AER	21.3	20.7	21.1	**20.6**	14.3	13.7	14.1	**13.3**	19.8	18.6	19.4	**18.3**
WPrec	98.3	97.8	97.7	97.0	97.3	96.6	96.9	96.4	96.7	96.5	96.4	95.8
WRec	66.5	67.6	67.0	68.1	70.9	72.3	72.1	73.0	65.0	66.6	65.7	67.2
WF1	79.3	79.9	79.5	**80.1**	82.0	82.7	82.6	**83.1**	77.7	78.8	78.1	**79.0**

the most difficult. In all settings, the results show an overall AER reduction over all symmetrized IBM Model 4 predictions. In terms of weighted F-Measure, the results show a higher F-score for all text types except for the Europarl texts.

For all text types and in all experimental settings, the proposed system improves the recall of the intersected IBM Model 4 word alignments without sacrificing precision.

Overall, enlarging the training set has a positive effect on the system's performance. More precise results can be obtained by only allowing translation rules that contain lexical clues (either abstract PoS rules with lexical indices or lexicalized rules).

Table 4 gives an overview of the total number of validated rules in the different experimental settings and gives details on the number of discontiguous (either abstract or lexicalized) and lexicalized validated rules. As expected, the number of validated rules increases if the corpus size is increased. If only translation rules that contain lexical clues are allowed, the number of validated translation rules is drastically reduced. The share of discontiguous rules ranges from 23 to 39%; the share of lexicalized rules from 31 to 48%.

On the right-handside of the table, the number of applied rules in the different test corpora is given. In order to process the 40,000 words of the test corpora, 14 to 24% of the rules are applied. The share of discontiguous rules accounts for 14 to 20%.

Some example rules are given below:

- N_1 → DET+N_1 (*History* → *de Geschiedenis*)
- DET_1+N_2+N → DET_1+N_2 (*a movie producer* → *een filmproducent*)
- PREP_1+V-prpa_2 → PREP_1 | DET+N_2 | PREP (*for managing* → *voor het management van*)
- DET_1+N_2;PREP | N_3 → DET_1+N_2+N_3 (*a number of events* → *een aantal evenementen*)
- V-fin_1+V-papa_2 → V-fin_1 ... V-papa_2 (*had written* → *had ... geschreven*)
- ADV_1;...;ADJ_2 → ADV_1+ADJ_2 (*not;...;longer* → *niet langer*)
- last → voor het laatst
- has → beschikt ... over
- agree → ben | het ... eens

The most frequently applied rules take care of the insertion of a determiner in Dutch (e.g. *History* → *de Geschiedenis*) or deal with Dutch compounds of which only a part has been aligned by the GIZA++ intersected word alignments (e.g. *filmproducent*). The most frequently applied discontiguous rules deal with verbal groups that are often split in Dutch (*had ... geschreven*). However, other discontiguous chunks are captured as well. The discontiguous lexicalized rules are able to deal with phrasal verbs (e.g. *beschikken ... over*).

Table 4. Total number of validated rules in the different settings and number of validated discontiguous and lexicalized rules; total number of applied rules in the test corpora and number of applied discontiguous and lexicalized rules

	Validated			Applied		
	Total	Discont.	Lexicalized	Total	Discont.	Lexicalized
10Lex	1526	344	724	303	46	70
10All	2135	574	744	508	104	70
20Lex	3790	1174	1828	530	108	153
20All	5826	2287	1828	872	249	153

4 Conclusion and Future Work

We developed a new chunk-based method to add language-pair specific knowledge – derived from shallow linguistic processing tools – to statistical word alignment models. The system is conceived as a cascaded model consisting of two phases. In the first phase *anchor chunks* are linked on the basis of the intersected IBM Model 4 word alignment and syntactic similarity. In the second phase, we use a bootstrapping approach to extract language-pair specific translation patterns.

We demonstrated that the proposed system improves the recall of the intersected IBM Model 4 word alignments without sacrificing precision, which makes the resulting alignments more useful for incorporation in CAT-tools or bilingual terminology extraction tools. Moreover, the system's ability to align discontiguous chunks makes the system useful for languages containing split verbal constructions and phrasal verbs.

As the chunk-based extension aligns chunks rather than words, we assume that incorporation of these precise chunks in the SMT phrase tables has a positive impact on Machine Translation quality as well. In future work, we would like to evaluate our approach in an existing phrase-based SMT system.

References

1. Planas, E.: SIMILIS Second-generation translation memory software. In: 27th International Conference on Translating and the Computer (TC27), London, United Kingdom, ASLIB (2005)
2. Itagaki, M., Aikawa, T., He, X.: Automatic Validation of Terminology Consistency with Statistical Method. In: Machine Translation Summit XI. European Associaton for Machine Translation, pp. 269–274 (2007)
3. Macken, L., Lefever, E., Hoste, V.: Linguistically-based Sub-sentential Alignment for Terminology Extraction from a Bilingual Automotive Corpus. In: Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), Manchester, United Kingdom (2008)
4. Och, F.J., Ney, H.: A systematic comparison of various statistical alignment models. Computational Linguistics 29(1), 19–51 (2003)
5. Brown, P.F., Della Pietra, V.J., Della Pietra, S.A., Mercer, R.L.: The Mathematics of Statistical Machine Translation: Parameter Estimation. Computational Linguistics 19(2), 263–311 (1993)
6. 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, Czech Republic, Prague. Association for Computational Linguistics, pp. 177–180 (2007)
7. Ganchev, K., Graça, J.V., Taskar, B.: Better Alignments = Better Translations? In: Proceedings of ACL 2008: HLT, Columbus, Ohio. Association for Computational Linguistics, pp. 986–993 (2008)

8. Zhang, H., Quirk, C., Moore, R.C., Gildea, D.: Bayesian Learning of Non-Compositional Phrases with Synchronous Parsing. In: Proceedings of ACL 2008: HLT, Columbus, Ohio. Association for Computational Linguistics, pp. 97–105 (2008)
9. DeNero, J., Klein, D.: Tailoring Word Alignments to Syntactic Machine Translation. In: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, Prague, Czech Republic. Association for Computational Linguistics, pp. 17–24 (2007)
10. Tiedemann, J.: Combining Clues for Word Alignment. In: Proceedings of the 10th Conference of the European Chapter of the ACL (EACL 2003), Budapest, Hungary (2003)
11. Daelemans, W., van den Bosch, A.: Memory-based language processing. Cambridge University Press, Cambridge (2005)
12. van den Bosch, A., Busser, B., Daelemans, W., Canisius, S.: An efficient memory-based morphosyntactic tagger and parser for Dutch. In: Selected Papers of the 17th Computational Linguistics in the Netherlands Meeting, Leuven, Belgium, pp. 191–206 (2007)
13. Abney, S.: Parsing by chunks. In: Berwick, R., Abney, S., Tenny, C. (eds.) Principle-Based Parsing. Kluwer Academic Publisher, Dordrecht (1991)
14. Melamed, D.I.: Models of translational equivalence among words. Computational Linguistics 26(2), 221–249 (2000)
15. Moore, R.C.: Association-Based Bilingual Word Alignment. In: ACL Workshop on Building and Using Parallel Texts, Ann Arbor, Michigan, United States, pp. 1–8 (2005)
16. Dunning, T.: Accurate Methods for the Statistics of Surprise and Coincidence. Computational Linguistics 19(1), 61–74 (1993)
17. Manning, C.D., Schütze, H.: Foundations of Statistical Natural Language Processing. Massachusetts Institute of Technology (2003)
18. McEnery, T., Xiao, R., Yukio, T.: Corpus-based Language Studies. An advanced resource book. Routledge, London (2006)
19. Macken, L., Trushkina, J., Rura, L.: Dutch Parallel Corpus: MT corpus and Translator's Aid. In: Machine Translation Summit XI, Copenhagen, Denmark, pp. 313–320 (2007)
20. Melamed, D.I.: Empirical Methods for Exploiting Parallel Texts. MIT Press, Cambridge (2001)
21. Davis, P.C.: Stone Soup Translation: The Linked Automata Model, Unpublished PhD, Ohio State University (2002)

Computing Transfer Score in Example-Based Machine Translation

Rafał Jaworski

Adam Mickiewicz University
Poznań, Poland
rjawor@amu.edu.pl

Abstract. This paper presents an idea in Example-Based Machine Translation - computing the transfer score for each produced translation. When an EBMT system finds an example in the translation memory, it tries to modify the sentence in order to produce the best possible translation of the input sentence. The user of the system, however, is unable to judge the quality of the translation. This problem can be solved by providing the user with a percentage score for each translated sentence.

The idea to base transfer score computation on the similarity between the input sentence and the example is not sufficient. Real-life examples show that the transfer process is as likely to go well with a bad translation memory example as to fail with a good example.

This paper describes a method of computing transfer score strictly associated with the transfer process. The transfer score is inversely proportional to the number of linguistic operations executed on the example target sentence. The paper ends with an evaluation of the suggested method.

1 Introduction

During the latest studies on EBMT (e.g. [1], [2], [3]), a serious problem with the development of usable EBMT systems emerged. The usability of a newly created EBMT system cannot be assessed based on automatic machine translation metrics, such as BLEU ([4]). There are two reasons behind it. Firstly, the automatic translation metrics were created rather for RBMT systems and the scores they produce do not reflect the nature of EBMT translations. Secondly, it is hard to say whether a system scoring well in BLEU (or any automatic metric) can be used in real-life situations.

As the issue of usability has been raised, a definition of usability of an EBMT system must be provided. This paper describes the implementation of an EBMT system as a translation-aide tool. CAT (Computer-Aided Translation) systems are not intended to replace a human translator at all stages of the translation process. Their role is limited merely to suggesting a possible translation of a sentence to a professional translator who performs post-editing. In that sense, a usable EBMT system is a system producing such translations that are easy to correct and based on which a good translation can be obtained with minimum

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 406–416, 2010.

effort. Therefore, the translation score computation process described in this paper serves as a measure of usability of translations produced by an EBMT system. The translation score metric is an answer to the drawbacks of automatic machine translation metrics like BLEU.

Section 2 of this paper describes the architecture of an EBMT system in which the transfer score computation mechanism was embedded. Section 3 outlines the transfer score computation mechanism itself. Section 4 compares the transfer score metric to other automatic machine translation metrics. It also discusses the experiment of comparing the transfer score metric with human judgment of usability of translations. The final section presents the conclusions.

2 EBMT System Architecture

The system basic architecture resembles the architectures of other EBMT implementations. Like the EBMT system designed at the Chinese Academy of Science ([5]), it consists of two basic modules: Example Matcher and Transferer. The former is responsible for finding an example best suited for the input sentence in the translation memory. The latter tries to modify the example's target sentence so that it can be returned as translation of the input sentence.

2.1 Word Substitution

The Transferer module performs operations to produce the translation of the input sentence. The crucial one is word substitution. The mechanism of this substitution can be explained using the following example:

INPUT SENTENCE (in Polish): "Uwzględniając Traktat ustanawiający Parlament Europejski". (in English: Having regard to the Treaty establishing the European Parliament).

Example from the translation memory:
SOURCE SENTENCE (in Polish): "Uwzględniając Traktat ustanawiający Wspólnotę Europejską".
TARGET SENTENCE (in English): "Having regard to the Treaty establishing the European Community."

The first operation is to check the resemblance of the input and source sentences. The result of this operation is presented in Figure 1. Each word of the input sentence is assigned to a word in the source sentence. The assignments are established using a monolingual dictionary. Solid lines represent equal words, while dotted lines represent different word forms within the same lexeme. The only word left unassigned in the input sentence is *"Parlament"* (Parliament), which does not have a match in the source sentence. Similarly, the word *"Wspólnotę"* (Community) in the source sentence does not have a match in the input sentence. At this point a significant decision is made. The word *"Wspólnotę"* in the example is substituted with the word *"Parlament"* from the input sentence.

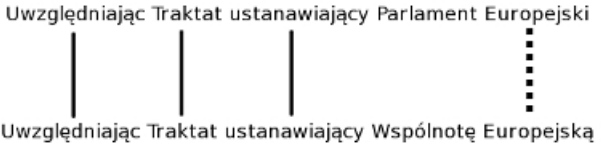


Fig. 1. The resemblance of the input and source sentences

The consequence of this decision is that the example target sentence, which will be used to produce the translation, must also be changed (in response to the change in the source sentence). In order to specify what modifications the target sentence must undergo, the resemblance of the source and target sentences must be checked. This operation corresponds to the operations in [6]. Here, however, it is performed during the transfer, not during the preparation of the translation memory. The operation is done using a bilingual dictionary. The results are presented in Figure 2. The dotted lines represent the alleged correspondence of Polish and English words (words which can be each other's translations are connected with these lines). The most significant information in the above diagram is the correspondence of the word *"Wspólnotę"* with the word "Community". It allows us to find the spot for the word *"Parlament"* from the input sentence. The Transferer module will put the word *"Parlament"* from the input sentence into the example's target sentence in the place of the word "Community". The result is as follows:

"Having regard to the Treaty establishing the European *Parlament*".

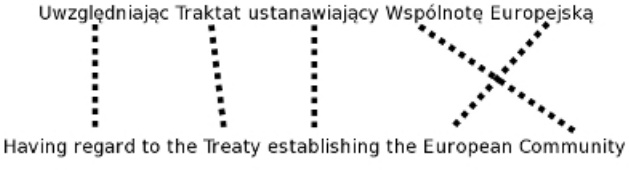


Fig. 2. The resemblance of the source and target sentences

The final step is translating the word *"Parlament"* into English using a dictionary. This operation produces the final translation output:

"Having regard to the Treaty establishing the European Parliament".

2.2 NE Recognition and Substitution

A Named Entity in this EBMT system is defined as one or more words of a sentence having special meaning and to which special translation rules must be applied. Named Entity recognition plays an important role in the process of

Machine Translation ([7]). Named Entities often carry the most important information in a sentence. At the same time, Named Entities are prone to translation errors because of their irregularity. Hence, dealing with Named Entities during the process of translation can considerably improve translation quality ([7]).

Here, Named Entities are used in a substitution mechanism similar to word substitution. There are, however, two main differences between the two mechanisms. First, words are extracted from the sentences by tokenization, while recognizing Named Entities is done by a special Named Entity recognition mechanism. The other difference lies in the process of translation - words are translated using dictionaries while Named Entity translation requires special Named Entity translation rules.

Named Entity recognition and translation is handled using rules written in a formalism called NERT (full specification of the formalism is available in [7]).

The following example illustrates the substitution of Named Entities during the process of transfer.

INPUT SENTENCE (in Polish): "Przepis Dz.U. z 12.03.2001 NR 4, poz. 17 brzmi następująco" (in English: "The regulation in Journal of Laws of 2001/03/12 No. 4, item 17 states the following").

Example from the translation memory:
SOURCE SENTENCE (in Polish): "Przepis Dz.U. z 17.01.1997 NR 8, poz. 2 brzmi następująco" (changed reference)
TARGET SENTENCE (in English): "The regulation in Journal of Laws of 1997/01/17 No. 8, item 2 states the following"

As with word substitution, the first step is to check the resemblance of the input and source sentences. During the process, Named Entities are recognised. The results of resemblance check are presented in Figure 3. The only difference between the two sentences is the target of the reference to the Journal of Laws. Therefore, the reference in the example will be substituted with the reference from input sentence. The information on resemblance of the source and input sentences is needed to complete the substitution. It is presented in Figure 4. The reference 'Dz.U. z 12.03.2001 NR 4, poz. 17' from the input sentence is to be put into the target sentence in place of the reference 'Journal of Laws of 1997/01/17 No. 8, item 2'. The reference is also translated into English. The resulting translation is:

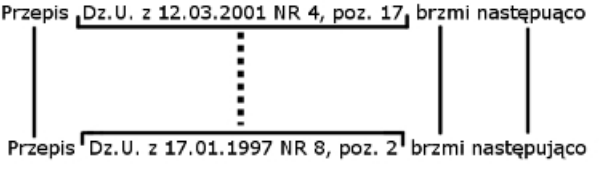


Fig. 3. The resemblance of the input and source sentences while substituting Named Entities

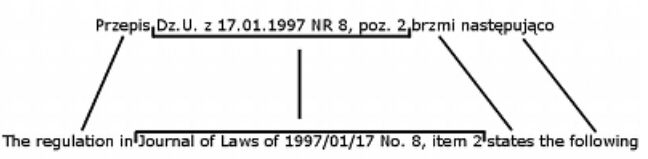


Fig. 4. The resemblance of the source and target sentences while substituting Named Entities

"The regulation in Journal of Laws of 2001/03/12 No. 4, item 17 states the following".

3 Transfer Score Computation Mechanism

3.1 General Idea

The idea of computing the transfer score in an EBMT system may be found in the ANTRA system [8]. However, the transfer score computation mechanism used in this system has proven not to reliable enough. Hence a different method of computing translation score has been developed.

3.2 Transfer Score Computation Algorithm

The transfer score computation is done by the Transferer module during the process of transfer. Hence, it is strictly connected with the transfer. The process consists of 3 steps:

1. Finding discrepancies between input and source sentences.
2. Imposing "penalties" for discrepancies.
3. Compensating "penalties" previously imposed for discrepancies (executed while transferring words and Named Entities from the input to the target sentence).

The "penalties" play a major role in score computation. They are real, positive numbers, representing discrepancies between the source and input sentences. These discrepancies are identified in step 1 of the score computation process. The more discrepancies, the less chance for the transfer process to produce a good translation, hence more penalties. The following table presents penalties for given discrepancies (the following abbreviations has been used: IS - input sentence, SS - source sentence):

The second step of computing the transfer score is taken during transferring words and Named Entities from the input to the target sentence. For each successful transfer, the penalty imposed previously is partially compensated for. The table below shows the compensations for each successful transfer operation: Note that when a transfer of words is possible, a 2.0 penalty must have been imposed during checking of resemblance between IS and SS (The tranferred word

Table 1. Penalties imposed for discrepancies between input and source sentences

Discrepancy	Penalty value
Dictionary correspondence (not identity) of words	0.5
Type correspondence (not identity) of NE	0.5
Word from IS missing in SS	1.0
Word from SS missing in IS	1.0
NE from IS missing in SS	1.5
NE from SS missing in IS	1.5
Inversion of pair of words	0.5
Missing punctuation mark	0.25

Table 2. Compensations for transfer operations

Transfer operation	Compensation
Transfer of a word	1.5
Transfer of a NE	0.4

from IS must have been missing in SS. Also, the word from SS, in whose place the tranferred word was put, must have been missing in IS). The compensation leaves a penalty of 0.5 (a measure of uncertainty, as to whether the transfer of the word produced a correct translation). As for NE substitution, the compensation leaves only 0.1 penalty, as transfer of NE's is more reliable (once a NE is recognized, its transfer is executed using manually created rules).

After computing the final penalty value (the sum of all penalties imposed in step 1, reduced by compensations from step 2), the transfer score is arrived at using the following formula:

$$score = 1 - \frac{p}{avgLength} \tag{1}$$

p - total penalty
avgLength - average number of words and Named Entities in the source and input sentences.

4 Evaluation of the Transfer Score Metric

The transfer score metric was evaluated using both automatic and human-aided techniques. Automatic evaluation involved the comparison of the transfer score metric with other automatic machine translation quality metrics - BLEU ([4]) and METEOR ([9]). Human-aided evaluation, on the other hand, was aimed at finding out whether the transfer score metric can serve as a measure of usability of produced translations.

Algorithm 1. Test translation procedure

```
select 10000 units from Polish-English JRC Corpus at random
for each selected unit u
    s := u.source
    r := u.target

    train EBMT sytem with JRC Corpus without unit u
    t := translation of sentence s by EBMT system
    transfer_score := score of translation of sentence s

    meteor_score := METEOR score for test translation t and reference r
    bleu_score := BLEU score for test translation t and reference r

    store the triple (transfer_score, meteor_score, bleu_score) in a file
```

Fig. 5. Algorithm for computing scores of test translations

4.1 Test Procedure

The JRC Corpus ([10]) was used for tests. The procedure of computing scores of test translations is presented in Figure 5. Following the preparation of the set of scores, two correlations were analyzed separately: of the transfer and METEOR scores and of transfer and BLEU scores. To compute the correlation of two scores, they were treated as random variables X and Y. The first step of finding their correlation was computing their covariance from the following formula:

$$cov(X,Y) = \sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}) \tag{2}$$

Where:
x_i, y_i - individual scores for translations
$\bar{x}, \bar{y}$ - average scores

The next step was computing standard deviations of the two variables from the formula:

$$\sigma(X) = \sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{3}$$

The final step was computing the Pearson product-moment correlation coefficient ([11]) from the formula:

$$r_{XY} = \frac{cov(X,Y)}{\sigma(X)\sigma(Y)} \tag{4}$$

4.2 Comparison with METEOR

Figure 6 shows the chart of correlation between the transfer score and the METEOR score (it also shows the regression line). The computed Pearson product-moment correlation coefficient of the two measures in this case was **0.25**.

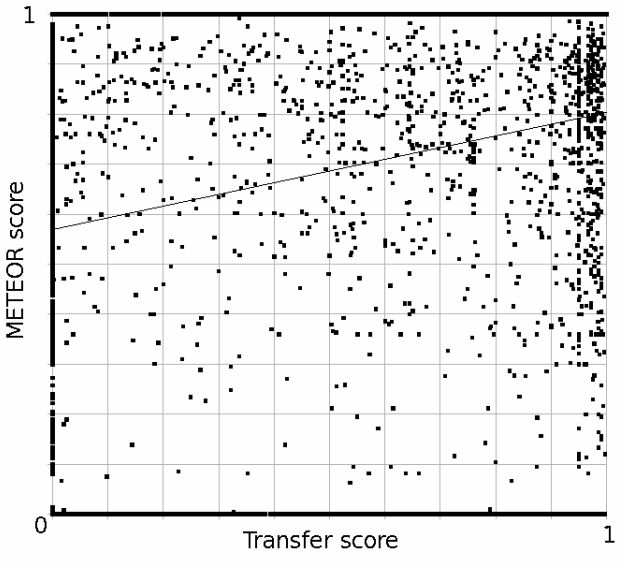


Fig. 6. The correlation between the transfer score and METEOR score

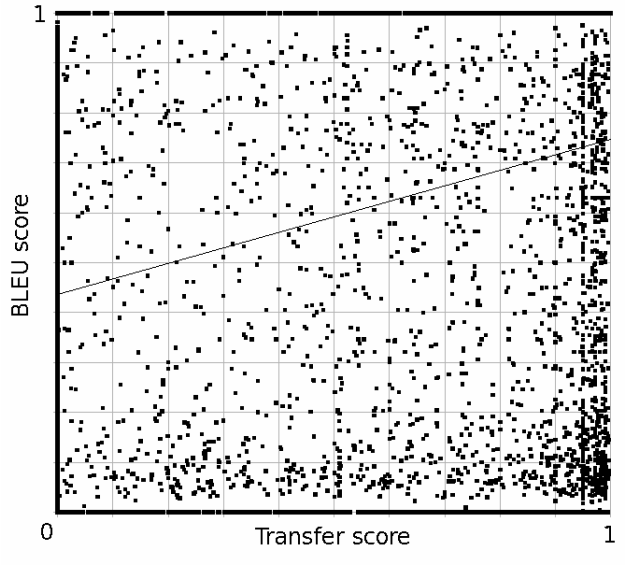


Fig. 7. The correlation between the transfer score and BLEU score

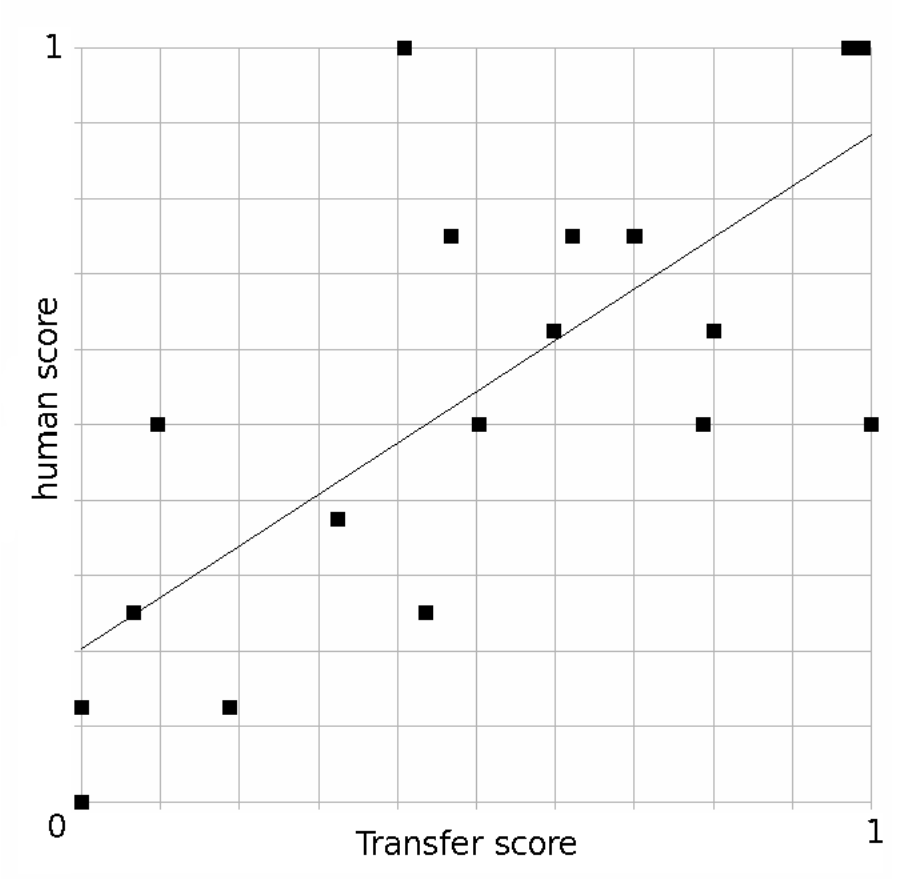


Fig. 8. The correlation between the transfer score and the human judgement

4.3 Comparison with BLEU

Figure 7 shows the chart of correlation between the transfer score and the BLEU score (as well as the regression line). The computed Pearson product-moment correlation coefficient of the two measures in this case was **0.32**.

4.4 Comparison with "Human Metric"

Apart from automatic evaluation, the transfer score metric was also evaluated by humans. Two Polish translators were asked to translate 20 Polish sentences from the JRC Corpus into English. For each sentence, they were given a suggestion for translation - a translation of the sentence performed by the EBMT system. The translators, however, were not provided with the translation scores for these translations. Instead, they were asked to judge the usability of suggestion using a 5 point scale (5 - no need to alter the suggestion, a good translation, 1 - suggestion to be

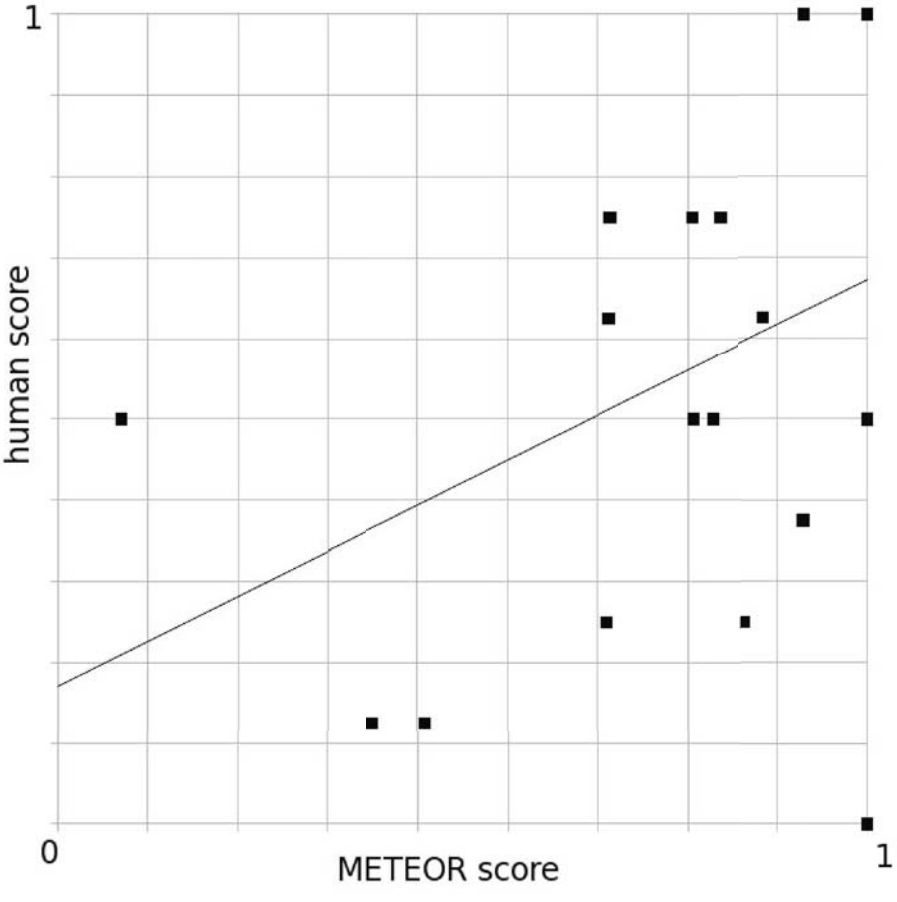


Fig. 9. The correlation between the METEOR score and the human judgement

ignored, need to translate the sentence from scratch). The score (marked as **p**) was then rescaled using the formula:

$$score = 0.25 * (p - 1) \tag{5}$$

The final human score for each translation was computed as an average of the two people's scores.

The correlation of the transfer score and the human scores was arrived at in a way similar to the previous evaluations. Figure 8 shows the correlation of the transfer score and the human scores. The computed Pearson product-moment correlation coefficient for the transfer score metric and human judgment was **0.74**.

For comparison, the correlation of the human scores and the METEOR score was also computed in a similar way. Figure 9 shows the correlation of

the METEOR score and the human scores. The computed Pearson product-moment correlation coefficient for the METEOR metric and human judgment was only **0.39**.

5 Conclusions

The evaluation showed that the transfer score metric used in the described EBMT system is not much correlated with the METEOR and BLEU metric. It is, however, unlike METEOR, well correlated with human judgment of usability of translation. Therefore, the transfer score can be used in a CAT system to help the user (translator) decide, to what extent can the translation suggestion of EBMT systems be relied on.

Evaluation of EBMT systems should not be carried out using the same techniques as with other machine translation systems, especially when the EBMT system serves as a CAT system.

References

1. Somers, H., Dandapat, S., Naskar, S.K.: A review of ebmt using proportional analogies. In: Proceedings of the 3rd International Workshop on Example-Based Machine Translation (2009)
2. Vandeghinste, V., Martens, S.: Top-down transfer in example-based mt. In: Proceedings of the 3rd International Workshop on Example-Based Machine Translation (2009)
3. Kurohashi, S.: Fully syntactic example-based machine translation. In: Proceedings of the 3rd International Workshop on Example-Based Machine Translation (2009)
4. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation (2002), http://www1.cs.columbia.edu/nlp/sgd/bleu.pdf
5. Hongxu, H., Dan, D., Gang, Z., Hongkui, Y., Yang, L., Deyi, X.: An EBMT system based on word alignment (2004), http://www.mt-archive.info/IWSLT-2004-Hou.pdf
6. Rapp, R., Vide, C.M.: Example-based machine translation using a dictionary of word pairs (2006), http://www.mt-archive.info/LREC-2006-Rapp.pdf
7. Jassem, K., Marcińczuk, M.: Semi-supervised learning rule acquisition for Named Entity recognition and translation (2008) (unpublished)
8. Gintrowicz, J.: Tłumaczenie automatyczne oparte na przykładach i jego rozszerzenia. Master thesis under the supervision of dr Krzysztof Jassem (2007)
9. Lavie, A., Agarwal, A., Denkowski, M.: The meteor metric for automatic evaluation of machine translation (2009), http://www.cs.cmu.edu/~alavie/METEOR/meteor-mtj-2009.pdf
10. Ralf, S., Pouliquen, B., Widiger, A., Ignat, C., Erjavec, T., Tufiş, 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 (2006)
11. Stigler, M.S.: Francis galton's account of the invention of correlation. Statistical Science (1989)

Systematic Processing of Long Sentences in Rule Based Portuguese-Chinese Machine Translation

Francisco Oliveira, Fai Wong, and Iok-Sai Hong

Faculty of Science and Technology, University of Macau.
Av. Padre Tomás Pereira, Taipa, Macao
{olifran,derekfw,ma66536}@umac.mo

Abstract. The translation quality and parsing efficiency are often disappointed when Rule based Machine Translation systems deal with long sentences. Due to the complicated syntactic structure of the language, many ambiguous parse trees can be generated during the translation process, and it is not easy to select the most suitable parse tree for generating the correct translation. This paper presents an approach to parse and translate long sentences efficiently in application to Rule based Portuguese-Chinese Machine Translation. A systematic approach to break down the length of the sentences based on patterns, clauses, conjunctions, and punctuation is considered to improve the performance of the parsing analysis. On the other hand, Constraint Synchronous Grammar is used to model both source and target languages simultaneously at the parsing stage to further reduce ambiguities and the parsing efficiency.

Keywords: Rule based Machine Translation, Sentence Partitioning, Constraint Synchronous Grammar.

1 Introduction

Most Rule based Machine Translation (MT) systems [1] can generate reasonable translations with short sentences. However, when sentences are long in length, the story is quite different. The parsing time is directly affected by the analysis required in determining the correct syntactic parse tree structure from several ambiguous trees. Moreover, MT systems have a higher probability to fail in the analysis, and produce poor translation results.

Based on an experiment conducted in studying over 2000 Portuguese sentences extracted online from a government department [2], we found that the average length is 19 words per sentence. Furthermore, many of them are very long in length. They don't have any punctuation except at the end with a full stop, or they have several fragments separated by too much punctuation although they are related to each other. This really shows that in most cases, MT systems need to deal with long sentences.

Recently, existing literature provides different approaches to overcome the problem of efficiency and to improve the translation quality. Researchers focused on breaking down complex and long sentences into several fragments based on a set of defined criteria.

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 417–426, 2010.

Some proposed the use of punctuations and conjunction words as the partition delimiter. Jin et al. [3], Xiong et al. [4] focused on partitioning Chinese long sentences based on comma. Li et al. [5] considered more types of punctuations in conjunction with a hierarchical parsing approach to tackle the problem. Although this concept is simple to implement, it is very easy to get wrong partitioning of fragments and lead to poor translation results.

Some proposed specific sequences of words that can be grouped together into grammatical constituents (noun, verb, adjective, clause, and other phrases) in splitting long sentences. Shallow parsing is then applied for each group of words identified, denoted as a chunk by Abney [6], instead of full parsing. The main purpose is to reduce the analysis in deciding the correct syntactic structure of a sentence, remove ambiguous cases in advance, and increase the efficiency as well as the translation quality. Different authors considered different types of chunks according to their interests. Garrido-Alenda et al. [7] proposed a MT system based on a partial transfer translation engine that relies on shallow parsing for structure transfer between the language pair. Yang [8] proposed a preprocessing module for chunking phrases in Chinese-Korean MT system.

Some defined syntactic patterns in the sentence partitioning. Kim et al. [9] defined a set of manually constructed pattern information to accomplish the task. To better acquire patterns automatically, Kim et al. [10] applied Support Vector Machines, and Kim et al. [11] used Maximum Entropy to learn and identify fragments of long sentences.

Each of these approaches has its strength and weakness in application to sentence partitioning. The combination of these methods seems the way to go in order to avoid the intrinsic obstacles of each approach. This paper presents different criteria defined for partitioning long sentences and their systematic execution in application to Rule based Portuguese-Chinese MT system. Our strategy divides sentence partitioning into three stages. Patterns including date, time, numbering, and phrases that have a specific sequence order are considered as the starting point to identify special fragments. In the second stage, partitioning is accomplished based on the punctuation, conjunction words and phrases delimiters. At last, all the fragments are shallow parsed in the identification of Noun Phrases (NP) before the full parsing, and generation of the target language. The parsing of the MT system is based on Constraint Synchronous Grammar (CSG) [12], which is used to model syntactic structures of two languages simultaneously. In order to perform necessary disambiguation during the parsing stage, feature constraints are defined for each CSG rule. Due to its characteristics, our MT system does not require another set of conversion rules to change the source parse tree into the target one. As a consequence, it can reduce errors during the transfer process, increase the parsing time, and strengthen the relationship between the parser and the generation modules.

This paper is organized as follows. Section 2 gives the details of each criterion considered in the identification of suitable fragments in long sentences. The whole process for partitioning long sentences in application to Rule based Portuguese-Chinese MT system is presented in Section 3. The evaluation is discussed in Section 4, and a conclusion is followed in Section 5.

2 Criteria in the Identification of Fragments in Long Sentences

In order to improve the quality of sentence partitioning for Portuguese language, three criteria have been studied and concluded.

2.1 Specific Pattern Rules

The first criterion is based on the identification of patterns. If there is an exact matching pattern, it is believed that high quality translation can be guaranteed for the fragment identified. Each pattern is written in Constraint Synchronous Grammar [12], a variation of synchronous grammar based on Context Free Grammar. Each production rule models both the source and the target sentential pattern for describing their relationships. An example of a CSG pattern is shown below.

$$\begin{aligned} \textit{Time} \rightarrow\ & \text{Number}_1\ \text{Symbol}_1\ \text{Number}_2\ \{ \\ & [\text{Number}^1\ \text{Symbol}^1\ \text{Number}^2]\ ;\ \text{Symbol}_1 = \text{“:”}\ \& \\ & \qquad 0 <= \text{Number}_1 < 24\ \& \\ & \qquad 0 <= \text{Number}_2 < 60\ \} \end{aligned} \tag{1}$$

The reduced syntactic symbol is *Time*, and the corresponding target sentential pattern is $Number^1\ Symbol^1\ Number^2$. The relationship between the languages is established by the given subscripts. This rule is only considered as success if the control conditions are satisfied. In this case, $Symbol_1$ has to be a colon, $Number_1$ should be a value from 0 to 23, and $Number_2$ should be a value from 0 to 59.

Different types of patterns are concluded, and they are summarized in Table 1. Many typical cases related to date, time, and specific patterns related to captions, numbers, currencies, article numbers, etc are defined to identify fragments that should not be partitioned.

Table 1. Types of Pattern Rules and Examples

Type of Pattern	Examples
Date	*25 de Abril, 2010* (25 of April, 2010); *25/04/2009*
Time	*15:30* ; *11:23:00* ; *5 em ponto* (5 o'clock on time)
Specific Patterns	*220/1000* ; *$1,222.00* ; *S.A.R.L.* ; *a)* ; *1)* ; *10.1*

We found that it is very common to have words linked together by punctuations and considered as a phrase. If the partition process starts with the punctuation delimiter criterion, in many cases, the partition will be incorrect and result in poor translation results. As a consequence, specific pattern rules identification must be considered before the punctuation delimiter criterion.

2.2 Punctuation and Conjunction

Punctuation and conjunction often have a high correlation with long sentences. An analysis was conducted in their distribution of the data collected in [2]. If we assume

that a long sentence contains more than 10 words, we found that 90% of these sentences contain either punctuation or conjunction. This reflects that they are widely used in linking fragments to form longer sentences.

Punctuation marks are essential in enabling a well organized written text. Since the main objective of punctuations is to add separators or pauses to organize logical groupings of sentences, they play an important role in sentence partitioning. Table 2 indicates the punctuations considered as the delimiter to separate fragments in long sentences, divided into four groups.

Table 2. Punctuation Delimiters and Conditions for the Partition

Group	Punctuation	Condition
1	Period, Exclamation Mark, Question Mark	Consider as Partition Delimiter
2	Single and Double Quotes, Parentheses, Brackets	Extract phrases bounded and mark them as one component
3	Hyphen, Dash	Split phrases only if the following are not met: Verb + Hyphen + (Reflexive or Object Pronoun)
4	Comma, Colon, Semicolon	Consider as Partition Delimiter

Group 1 is often marked as the end of a sentence. This group provides a good clue in identifying the end of a sentence, and each fragment should always contain a logical and complete thought.

Punctuations in Group 2 are used to highlight direct speech, important messages, or phrases with special meanings. Whenever there are fragments bounded by these punctuations in between, they are considered as one fragment of the sentence.

Group 3 is normally intended for separating complex phrases that relate to each other. They are only considered as a clue for splitting long sentences if some conditions are satisfied. In Portuguese, if the subject and the object are the same with reflexive verbs, then the reflexive pronoun is attached to the end of the verb preceded by a hyphen. The same case happens when object pronouns are used to describe something that is not the subject of an action. Under these circumstances, sentences are not partitioned. For example, “vestir-se” (to dress oneself) is composed of the verb to dress, hyphen, and the reflexive pronoun. In such a case, they are not separated into two fragments.

The last group is used to separate a compound sentence or fragments in a long list, and they are considered as a partition delimiter between sentences.

Conjunction words and phrases are also used to link words, phrases, and clauses together to form complex sentences. Based on the information given in a grammar book [13], 90 conjunction words and 37 conjunction phrases are considered as the partition delimiters.

As mentioned previously, this simple criterion can lead to wrong partition of sentences easily. In order to reduce the chance, conditions are defined for punctuation before the partition process. Even if there are some wrong partitions concluded after

the execution of this criterion, in the shallow parsing of NPs, they could be grouped together again. Since NPs are written in CSG, and there are a set of constraints defined for each rule, if there are fragments linked by punctuations or conjunctions satisfying the source sentential pattern and the constraints, they will be combined together to become a larger and correct fragment. As an example, it is very common to have two fragments with the same sense separated by comma or conjunctions words. In order to combine them together as one correct fragment, a CSG rule can be added by defining the source and target pattern, and stating that both fragments must have the same sense.

2.3 Noun Phrase Chunks

Shallow parsing of Noun Phrases is considered due to several reasons. First, the identification of NP chunks is often considered as a vital step for many Natural Language Processing tasks because NPs are arguably considered as the most important component in a sentence. In addition, this was also emphasized in the psycholinguistic studies of Gee and Grosjean [14] that NP chunks play an important role in Human language processing. Thus, the correct identification of NPs can not only disambiguate ambiguities in the generation of parse trees, but also improve the translation quality of the MT system.

Different categories of NP CSG rules are defined according to some of the analysis concluded from Costa [15]. Besides defining control conditions in the rules for restricting the selection of proper target language pattern, and the most suitable target translation, in order to prevent incorrect groupings of NPs, each category of NP is identified in a sequential order, as shown in Figure 1.

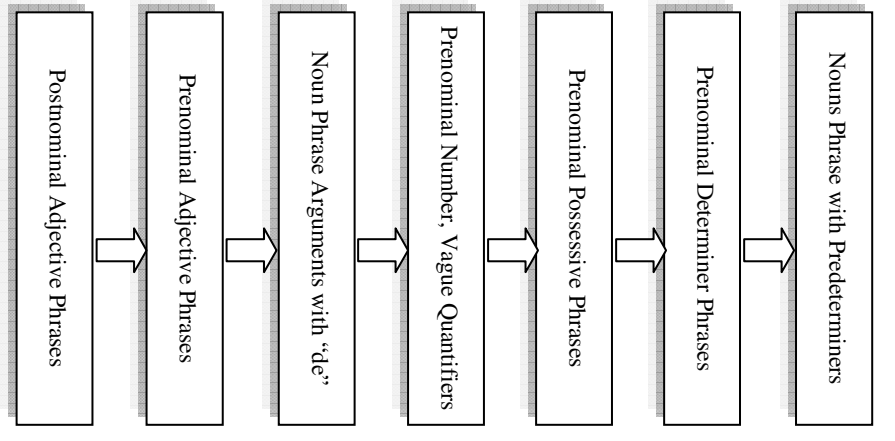


Fig. 1. Noun Phrase Chunking Process

As an example, below shows a simple NP CSG production rule executed in the Noun Phrase Arguments with "de":

$$\begin{aligned} NP \rightarrow\ & NP_1\ PP_1\ NP_2\ \{\ [NP^2\ PP^1\ NP^1]\ ;\ NP_{1sem} = NP_{2sem}\ \& \\ & PP_1 = \text{“de”} \\ & [Left\_NP^1\ NP^2\ Right\_NP^1]\ ; \\ & NP_1 = dynamic\_variable\ \& \\ & NP_{2sem} = SEM_{place}\ \& \\ & PP_1 = \text{“de”} \\ & \quad \ldots\ \} \end{aligned} \tag{2}$$

NP represents the reduced syntactic symbol, and there are two target language sentential patterns associated with the source language sentential pattern $NP_1\ PP_1\ NP_2$. The determination of the most suitable target pattern is based on a set of control conditions defined. These are not only used for inferring the structure of the source input, but also for inferring the structure of the target pattern during the generation phase. In the first condition, if the sense of NP_1 is the same as NP_2 and the preposition is "de", then the target $[NP^2\ PP^1\ NP^1]$ is associated with the source pattern. On the other hand, if the second condition is satisfied, then $[Left\_NP^1\ NP^2\ Right\_NP^1]$ is associated with the source. It is very common in the Chinese language that many words are discontinuous constituents. In the design of the dictionary, some entries are marked with dynamic variables, and the application of CSG can handle the case easily. As an example, suppose that the noun phrase "filhote de Lisboa" (a person who was born in Lisbon) is to be identified and parsed at this stage. Moreover, in the dictionary, the word "filhote" (a person who was born in) has the meaning of "出生於<N>的人". Since the second condition is satisfied, not only the NP chunk is identified, but also the corresponding translation is constructed at the same time by first splitting the left part of NP^1 "出生於" (was born in), combining it with NP^2 "里斯本" (Lisbon) and the right part of NP^1 "的人" (of people), to get "出生於里斯本的人" (a person who was born in Lisbon).

The remaining NPs that cannot be identified will be further considered in the last phase of full parsing, including complements of nouns, nouns followed by relative clauses and other preposition phrases, etc.

3 Long Sentence Partitioning and Translation Process

The overall architecture of the proposed long sentence partitioning module and its role in Rule based Portuguese-Chinese MT system is shown in Figure 2. In the preprocessing stage, the Portuguese sentence is first analyzed by the morphological module to restore the words into their original format, and then the corresponding Parts-of-Speech are assigned by the Tagging module. The analyzed result is then processed by the long sentence partitioning module. First, it checks if there are any exact patterns matching with the defined rules. In case of true, these are marked with a tag and will not be segmented further in the later stages of the process. Second, punctuations and conjunctions are considered as the delimiters to check if fragments should be segmented based on the conditions defined. Shallow parsing is then applied in identifying NP chunks. At last, all the fragments are combined together for a full CSG parsing, based on a modified version of generalized LR algorithm [16] that takes the features constraints and the inference of the target structure into consideration.

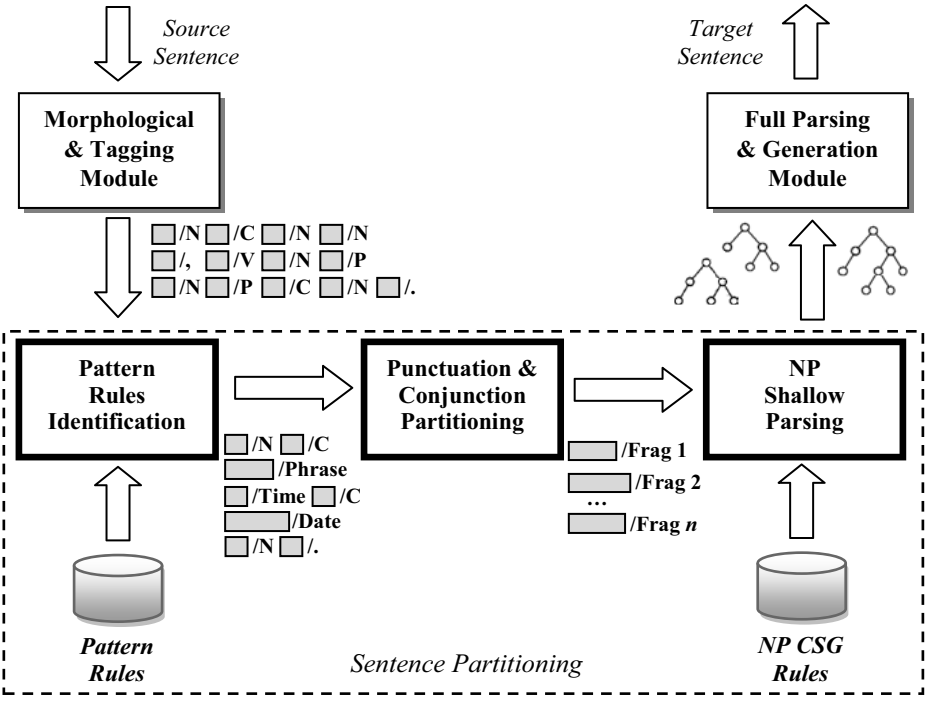


Fig. 2. Design Model of Long Sentences Partitioning Process

Since CSG can express the relationships between the source and the target languages, and allow multiple target productions in association with the same production rule based on different constraints, during the parsing, once the conditions are satisfied, the translation is actually being generated at the same time. This provides another solution in reducing the translation time effectively, especially for long sentences.

4 Evaluation and Discussion

In order to evaluate the performance of the Rule based MT system after taking consideration of the long sentence partitioning module, an experiment is conducted. The test suite includes 2070 sentences randomly extracted online from government pages [2], with an average of 19 words. The size of the MT's knowledge applied for the evaluation is shown in Table 3.

Figure 3 shows the average parsing time for sentences that have the same number of words. The horizontal axis represents the number of words in the sentence, and the vertical axis represents the average parsing time in seconds. The total average time for handling the translation with long sentence partition module only requires 46.72% of the time required when the translation is done without it. Since the computation time in the original approach is very high when dealing with long sentences, the help of the proposed module in splitting fragments and identifying chunks before the final parsing can

effectively reduce the work load of the whole system. On the other hand, we found that many short fragments identified by the long sentence partitioning module can be parsed successfully, guaranteeing the translation quality in advance before the full parsing.

Table 3. Resources used for evaluation purpose

Resources	Size
Bilingual Dictionary	110000 entries
Morphological and POS correction Rules	330 entries
Specific patterns	76 rules
Punctuation delimiters for partition	11
Conjunction delimiters for partition	127 words
CSG rules for NP identification	155 rules
CSG rules for Full parsing	630 rules

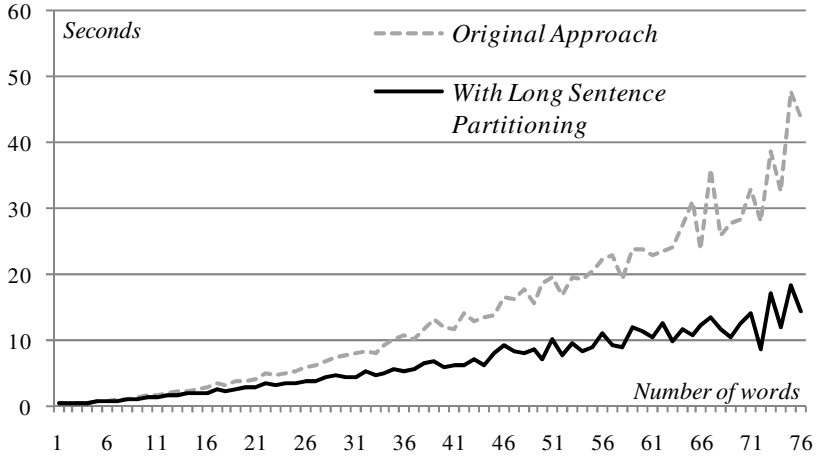


Fig. 3. Average Parsing Time (seconds) Comparison for sentences with the same number of words

Although the addition of the long sentence partition module can reduce the parsing time and produce better translation quality, it is impossible for Rule based MT systems to have enough rules in covering all the cases for any domain. These can either be defined manually or through machine learning approaches in the future to compensate this issue.

5 Conclusion

This paper presents a systematic approach in the partitioning of long sentences systematically in application to Rule based Portuguese-Chinese Machine Translation.

Three criteria are considered to accomplish the task before the full parsing and generation: specific pattern rules are identified from the phrases, sentence partitioning based on punctuation and conjunction delimiters, and shallow parsing in the identification of Noun Phrases. Each criterion is executed systematically in order to avoid improper partition of sentences. CSG is applied for modeling the relationship between the source and the target language in reducing the time during the generation of the target language. Based on the evaluation results, the system's performance with the proposed module has been improved.

Acknowledgments. This research work was supported by the Research Committee of University of Macau under Ref. UL019/09-Y1/EEE/DMC01/FST Cativo: 5868.

References

1. Bennett, W.S., Slocum, J.: The LRC Machine Translation System. Computational Linguistics 11(2-3), 111–121 (1985)
2. Macao Special Administrative Region Government Portal, http://www.gov.mo
3. Jin, M.X., Kim, M.Y., Kim, D., Lee, J.H.: Segmentation of Chinese Long Sentences Using Commas. In: SIGHAN Workshop on Chinese Language Processing, pp. 1–8 (2004)
4. Xiong, H., Xu, W., Mi, H., Liu, Y., Liu, Q.: Sub-Sentence Division for Tree-Based Machine Translation. In: Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP, Short Papers, Singapore, pp. 137–140 (2009)
5. Li, X., Zong, C., Hu, R.: A Hierarchical Parsing Approach with Punctuation Processing for Long Chinese Sentences. In: Proceedings of the Second International Joint Conference on Natural Language Processing, Companion Volume including Posters/Demos and tutorial abstracts, Jeju Island, Republic of Korea, pp. 7–12 (2005)
6. Abney, S.: Parsing by Chunks. Principle-Based Parsing, pp. 257–278. Kluwer Academic Publishers, Dordrecht (1991)
7. Garrido-Alenda, A., Gilabert-Zarco, P., Pérez-Ortiz, J., Pertusa-Ibáñez, A., Ramírez-Sánchez, G., Sánchez-Martínez, F., Scalco, M.A., Forcada, M.L.: Shallow Parsing for Portuguese-Spanish Machine Translation. In: Branco, A., Mendes, A., Ribeiro, R. (eds.) Language technology for Portuguese: shallow processing tools and resources, pp. 135–144 (2003)
8. Yang, J.: Phrase Chunking for Efficient Parsing in Machine Translation System. In: Monroy, R., Arroyo-Figueroa, G., Sucar, L.E., Sossa, H. (eds.) MICAI 2004. LNCS (LNAI), vol. 2972, pp. 478–487. Springer, Heidelberg (2004)
9. Kim, Y.B., Ehara, T.: A Method for Partitioning of Long Japanese Sentences with Subject Resolution in J/E Machine Translation. In: Proceedings of the 1994 International Conference on Computer Processing of Oriental Languages, Taejon, Korea, pp. 467–473 (1994)
10. Kim, Y.S., Oh, Y.J.: Intra-sentence segmentation based on support vector machines in English-Korean machine translation systems. Expert Systems with Applications: An International Journal 34, 2673–2682 (2008)
11. Kim, S.D., Zhang, B.T., Kim, Y.T.: Reducing parsing complexity by intra-sentence segmentation based on maximum entropy model. In: Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora, Hong Kong, pp. 164–171 (2000)

12. Wong, F., Hu, D.C., Mao, Y.H., Dong, M.C., Li, Y.P.: Machine Translation Based on Constraint-Based Synchronous Grammar. In: Proceedings of the 2nd International Joint Conference on Natural Language (IJCNLP 2005), Jeju Island, Republic of Korea, pp. 612–623 (2005)
13. Wang, S., Lu, Y.: Gramática da Língua Portuguesa. Shanghai Foreign Language Education Press (1999)
14. Gee, J., Grosjean, F.: Performance structures: A psycholinguistic and linguistic appraisal. Cognitive Psychology 15, 411–458 (1983)
15. Costa, F.N.Q.M.C.: Deep Linguistic Processing of Portuguese Noun Phrases. Master Thesis, University of Lisbon, Portugal (2007)
16. Tomita, M.: An efficient augmented-context-free parsing algorithm. Computational Linguistics 13(1-2), 31–46 (1987)

Syntax Augmented Inversion Transduction Grammars for Machine Translation

Guillem Gascó Mora and Joan Andreu Sánchez Peiró

Institut Tecnològic d'Informàtica,
Universitat Politècnica de València,
Camí de Vera s/n, València 46022, Spain
ggasco@iti.upv.es, jandreu@dsic.upv.es

Abstract. In this paper we propose a novel method for inferring an Inversion Transduction Grammar (ITG) from a bilingual parallel corpus with linguistic information from the source or target language. Our method combines bilingual ITG parse trees with monolingual linguistic trees in order to obtain a Syntax Augmented ITG (SAITG). The use of a modified bilingual parsing algorithm with bracketing information makes possible that each bilingual subtree has a correspondent subtree in the monolingual parsing. In addition, several binarization techniques have been tested for the resulting SAITG. In order to evaluate the effects of the use of SAITGs in Machine Translation tasks, we have used them in an ITG-based machine translation decoder. The results obtained using SAITGs with the decoder for the IWSLT-08 Chinese-English machine translation task produce significant improvements in BLEU.

1 Introduction

Phrase-Based Machine Translation (PBT) [1] systems split the input sentences into phrases, tranlate them and reorder the translated phrases in order to get a translation hypothesis. Although the PBT approach has been demonstrated to be one of the best approaches in translation of similar structured language pairs (Spanish-English, French-Italian...) [2], PBT systems usually have problems incorporating syntactic information in the translation process. For instance, most English sentences contain a subject and a verb, but it is difficult to include this information in a traditional phrase-based system. Syntactic motivated reorderings are usually also a problem for phrase-based systems.

There have been several attempts to incorporate syntactic information in PBT systems [3,4]. A new kind of machine translation decoder that uses Bracketing Translation Grammars (BTG) as a reordering model is presented in [5]. BTGs have only one non-terminal and model the probability of reordering, straight or inverse combination, of two neighbor blocks (partial translation hypotheses). A maximum entropy model is used in order to compute the probability of straight or inverse combination for each pair of blocks. The features used by the maximum entropy model are based on the boundary words of each of the blocks.

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 427–437, 2010.

In this work we extend the BTG framework presented in previous works to a more general framework of Inversion Transduction Grammars (ITG) [6]. BTG are ITG with only one non-terminal symbol. Thus, each of the non-terminals of the ITG can represent, for example, the syntactic constituents of one of the languages of interest.

We use a SITG-based decoder that allows us to retrieve the syntactic information, that is, the parse tree used during the decoding. This information can be used by rescoring systems or syntax correction post-processes. The use of a good grammar is very important for the performance of the decoder. For that reason, we present a new method for the inference of Syntax Augmented ITG from bilingual corpora using monolingual linguistic parse trees, either from the source or the target language.

The rest of the paper is structured as follows: In Section 2, we briefly explain the formalism of Stochastic Inversion Grammars (SITGs). In Section 3 we present the ITG-based decoder used in the experiments. In Section 4, we describe the method used for the inference of SITGs from a bilingual corpus and the association of linguistic information to the SITG. In Section 5, we show some experiments using the SAITG in a SITG machine translation decoder for the Chinese-English IWSLT2008 task. Finally, Section 6 explain the conclusions of the work.

2 Stochastic Inversion Transduction Grammars with Phrasal Productions

Inversion Transduction Grammars (ITGs) [6] are a restricted set of Synchronous grammars. Standard ITGs use only word-to-word transduction, however, in order to use the advantages of phrasal translation the original formalism has been extended to allow direct phrasal transductions.

An ITG with phrasal productions is a tuple $(N, \Sigma, \Delta, S, \mathcal{R})$ where N is the set of non-terminals, $S \in N$ is the root non-terminal, Σ is the source language alphabet, Δ is the target language alphabet, and $\mathcal{R}$ is a set of rules. Rules can be divided in two sets: syntactic rules (r^s) and lexical rules (r^l). Syntactic Rules have the form: $A \rightarrow [BC]$ or $A \rightarrow \langle BC \rangle$, where A, B and C are non-terminals and the brackets enclosing the right part of the rule means that the two non-terminals are expanded in the same order in the input and output languages, whereas the rules with pointed bracketing expand the left symbol into the right symbols in the straight order in the input language and in reverse order in the output language. Lexical Rules have the form $A \rightarrow x/y$ where $x \in \Sigma^*$ and $y \in \Delta^*$. It must be noted that x or y can be the empty string, denoted by ϵ, but it is not allow ϵ in both of the same production.

A derivation is a sequence of independent applications of lexical and syntactic rules ($D = \{r_1 ... r_n\}$). We say that a derivation is a complete derivation when the first rule starts from the root non-terminal (S) and after the application of the last rule, there are not non-terminals in the resulting bilingual string. We say that a complete derivation D yields a bilingual string (s, t), when (s, t) is

the result of the application of the productions of the derivation, and we denote it as $S \Rightarrow^D (s,t)$.

A Stochastic ITG (SITG) is the natural stochastic extension of an ITG where each one of the rules has been augmented with a probability. The probability of a derivation is the product of the probabilities of its rules.

3 A SITG-Based Decoder for Machine Translation

We use the SITG formalism described above as a translation model: given one source language sentence s we must find a target language sentence t^* that maximize the probability of a complete derivation that yields the bilingual sentence (s,t). We can obtain also the resulting derivation. That is

$$(t^*, D^*) = \underset{(t,D)}{\operatorname{argmax}} \Pr(S \Rightarrow^D (s,t)) \ . \tag{1}$$

In order to increase the performance of the decoder we added several additional models commonly used in SMT and we combine them using a log-linear combination of probability models [7]. The probability of a derivation is then:

$$\Pr(D) \propto \prod_i h_i(D)^{\lambda_i} \ . \tag{2}$$

where $h_i(D)$ is a set of features defined over the derivations and λ_i are feature weights. The features used in this work are:

SITG Probability: Probability of the SITG rules of the derivation: $h_1 = Pr_R(D)$.

N-gram Language Model: Probability of the target sentence using a n-gram language model: $h_2 = Pr_{LM}(t)$

Direct Translation Probability: Probability of the target sentence given the source sentence: $h_3 = Pr(t|s)$

Inverse Translation Probability: Probability of the source sentence given the target sentence: $h_4 = Pr(s|t)$

Lexical Direct Probability: Probability of translation of the source sentence words into the target sentence words using an IBM1 translation model: $h_5 = Pr_{IBM}(t|s)$

Lexical Inverse Probability: Probability of translation of the target sentence words into the source sentence words using an IBM1 translation model: $h_6 = Pr_{IBM}(s|t)$

Word Penalty Factor: This feature is used to model the length of the output: $h_7 = exp(|t|)$, being $|t|$ the number of words of the target sentence.

Phrase Penalty Factor: This feature is used to control the number of phrases used in the translation. $h_8 = e$

As it is made in [5], we grouped these features into three groups: reordering model $P_R = h_1^{\lambda_1}$, language model $P_L = h_2^{\lambda_4}$ and transduction model $P_T = \{h_3^{\lambda_3} ... h_8^{\lambda_8}\}$.

The language model and the transduction model are the usual for a PBT system. In order to obtain the transduction model, we followed the first method explained in [1]. As a reordering model, we used a Syntax Augmented ITG trained from a bilingual corpus. In Section 4 we explain the process of inference of such grammar.

During the decoding process, the source language sentence is split in phrases that are translated using the lexical rules of the SITG and then merged in a straight or inverted order using the syntactic rules. The search algorithm used in the decoder is similar to the CYK parsing algorithm for context-free grammars [8] but storing in each cell of the chart, not only the non-terminals, but also the partial translation hypotheses. The use of n-gram language models has been demonstrated to be very useful for PBT systems. However, in contrast to the other models, the n-gram language model probability of a derivation cannot be computed as a product of the language model probabilities of the rules used in the derivation (it depends on the context). The most likely translation of a sentence may use partial hypotheses that were not the most likely in their respective cells of the CYK chart. Hence, when including the n-gram language model, the optimality of the CYK algorithm is no longer guaranteed and its use is not enough to get the most likely translation.

In order to get the most likely translation, we need a translation hypotheses stack (from now on referred as Agenda) in each cell of the CYK-like chart instead of a single hypothesis. The hypotheses of two Agendas are combined in straigh or inverted order and the n-gram language model score of the new resulting hypothesis must be recomputed. Figure 1 shows an example of the combination of two different Agendas. First the decoder use the direct combination by means of the rule SN $\rightarrow$ [ADJ NN]. That means that the output strings of the two hypotheses, "verde" and "bruja", are concatenated in straight order creating a new hypothesis with the output string "verde bruja". The language model of this new hypothesis is computed and the hypotheses enters in the Agenda. The same process is used for the inverse combination with the rule SN $\rightarrow \langle$ADJ NN$\rangle$ and we obtain the hypothesis "bruja verde".

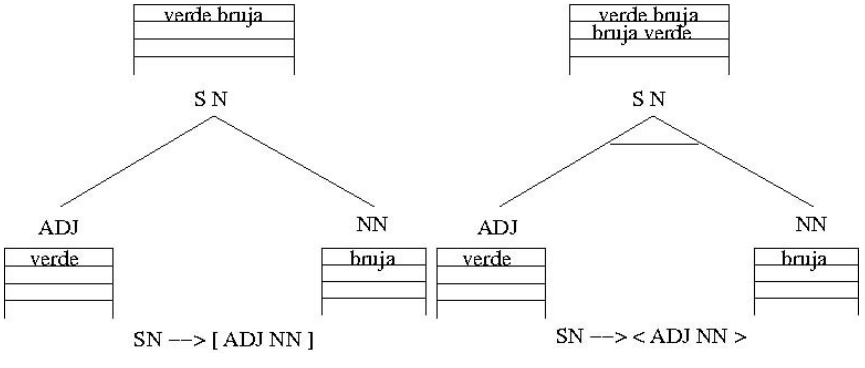


Fig. 1. Inverse and direct combination of two hypotheses

This algorithm could search exhaustively through the whole space of hypotheses. However, for big corpora and long sentences, we need to apply some kind of restriction over the search space. When two hypotheses in a Agenda have the same target language part, the less likely can be discarded without the risk of losing the optimality. We call this process hypotheses recombination.

As it happens with PBT systems, the free-risk hypotheses recombination is usually not enough. Hence, we use also two kinds of pruning that, in some of the cases, can make the decoder to lose the most likely translation:

1. Histogram Pruning: In each agenda only the n most likely hypotheses are stored.
2. Beam Pruning: We only store a hypothesis in an agenda if its probability is greater than $\gamma \cdot \Pr(h^*)$ where h^* is the hypothesis with the highest probability and γ is a real number between 0 and 1.

Both pruning strategies are parameterizable, so it can be chosen between a slow but precise search or a fast and more inaccurate one.

4 Inference of Syntax Augmented ITGs

In this section we describe the process of obtaining a SAITG from a bilingual corpus. This method consist of three basic steps: first we created an initial SITG, then we reestimated the probabilities of such SITG and finally we assigned linguistic information to the non-terminals of the grammar.

The two first steps are the same as explained in [9]: We assigned the probability of alignment of the words of the corpus $\Pr(s|t)$ (IBM models) to the lexical rules of the form $A \rightarrow s/t$. Then, we created all the possible syntactic rules (direct and inverse) using four non-terminal symbols (from NT1 to NT4) and assigning to them a low random probability. The grammar was smoothed by adding all the possible rules of the form $A \rightarrow s/\epsilon$ and $A \rightarrow \epsilon/t$ with a low probability.

With the initial SITG we applied several iterations of the Viterbi re-estimation algorithm. Hence, we parsed the corpus to get the most likely parse tree of each pair of sentences of the corpus and then we reestimated the probabilities of the productions of the SITG by counting and normalizing the occurrences of the rules in all the trees.

4.1 Linguistic Annotation of Non-terminals

The main objective of this process is to incorporate linguistic information to the SITG obtained in the previous steps. Figure 3 shows the process that was carried out over each of the sentence pairs of the bilingual corpus to get the Syntax Augmented ITG.

Suppose that we have the sentence pair $(L1, L2)$. First, using a monolingual parser, we got the parse tree of $L1$ and binarized it. Several binarization

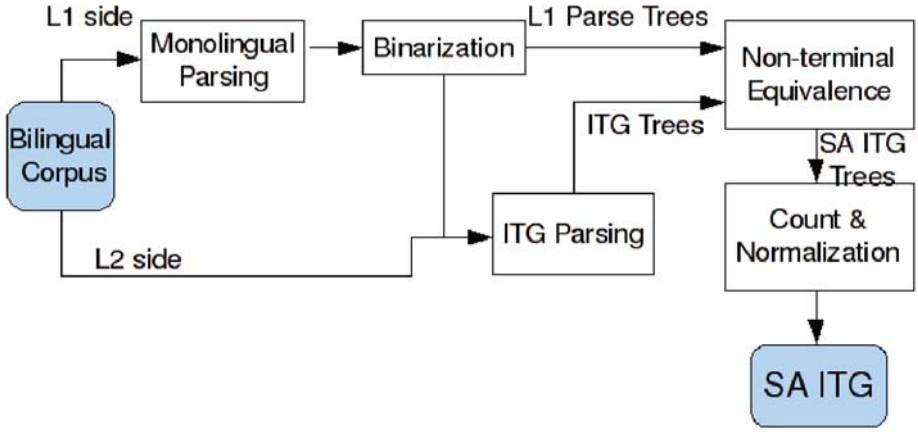


Fig. 2. Schema of the process of linguistic meaning association

strategies can be used but the preliminary experiments showed that the left binarization offered the best results. From now on, all the results reported have been using the left binarization.

Then we obtained the bracketed sentence resulting from the linguistic parse tree and together with $L2$ we use the bilingual parsing algorithm with bracketing information proposed in [9]. The bilingual parsing algorithm with bracketing information restricts the search of SITG trees to those that agree to the bracketing of the input sentence. Then an equivalence between the nodes of the linguistic parse tree and the nodes of the bilingual ITG tree must be defined. We say that one node of a source language monolingual parse tree that yields a phrase p is equivalent to one node of an ITG tree that yields a bilingual phrase (s, t) when $p = s$. A similar definition can be applied when the monolingual tree is from the target sentence. The use of the bracketing restriction guarantees the equivalence of the monolingual and bilingual trees. Figure 3 shows an example of equivalences between trees.

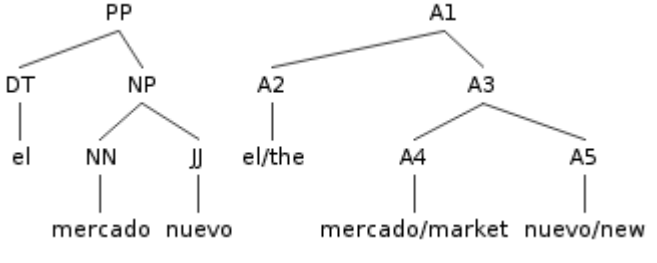


Fig. 3. Node equivalence between trees. For example nodes NP and A3 are equivalent

Once we have established the equivalence between nodes, we replaced the non-terminals of the ITG tree with the non-terminals of the monolingual tree that are equivalent. Hence, we obtained ITG trees with linguistic non-terminals: SAITG trees. Finally, we computed the probabilities for the rules by counting and normalizing the productions of the SAITG trees of the whole corpus.

The resulting SAITG was smoothed by adding all the production of the form $A \rightarrow a/\epsilon$, $A \rightarrow \epsilon/b$ and productions for the out of vocabulary words with a low probability.

The whole process can use either the source or the target language syntactic information. When the source language linguistic trees are used, the SAITG models the reorderings of the syntactic constituents of the source language. On the contrary, if we use the target language information, the SAITG models how the source language words are translated and reordered into target language syntactic structures. In addition, the decoder allows us to obtain the parse tree resulting from the decoding. Parse trees of the target language can be used in rescoring or syntax correction post-process systems.

5 Experiments

In this section we describe the experiments carried out on the Chinese-English task of the corpus IWSLT 2008. The results presented were computed over the lowercased and tokenized corpus. We used the training method of the software Moses [10] in order to obtain the phrase tables. The alignments for the initial SITG were computed using GIZA++ [11] and the weights of the log-linear combination of models were determined using the Minimum Error Rate training software ZMERT [12]. The linguistic parse trees for the SAITG inference were obtained using the Chinese version of the Stanford Parser [13]. As a baseline system, we used the phrase-based translation system Moses [10], from now on referred as PBT. The phrase table for both systems, PBT and SITG-decoder, are the same for all the experiments. Both decoders use the same 5-gram language model obtained from the corpus with the software SRILM [14].

5.1 Data

The experiments described in this section were carried out using the training and development sets provided for the Chinese-English BTEC task of the IWSLT 2008 evaluation campaign. There are 5 different development sets (devsets 1, 2, 3, 6, and 7). We added the devsets 1, 2 and 3 to the training set, devset6 was used for tuning the system and devset7 as a blind test set. The statistics of the resulting training, development and test sets are shown in Table 1. Since the development set is a multi-reference file, the number in the English side of the table is in fact, the number of words divided by number of references.

5.2 Results

Four different ITG have been tested for the decoder: The initial SITG; the re-estimated SITG (with Viterbi re-estimation algorithm); a SAITG with linguistic

Table 1. Statistics for the partitions used of the BTEC corpus

Corpus Set	Statistic	Chinese	English
Training	Sentences	42,655	
	Words	330,163	380,431
	Vocabulary Size	8,773	8,387
DevSet	Sentences	489	
	Words	3,169	3,861
	OOV Words	111	115
Test	Sentences	507	
	Words	3,357	-
	OOV Words	97	-

information from the source language and a SAITG with linguistic information from the target language.

The results presented were evaluated with respect to the BLEU machine translation evaluation metric [15]. The results obtained using the partitions described above are reported in Table 2. The initial SITG did not provide important syntactic information to the system, so the decoding process was almost completely driven by the phrase table and the language model. For that reason, the results of the PBT and the initial ITG are quite similar. The re-estimation of the SITG and the use of a SAITG improved significantly the performance of the system (improvement of 1.62 points in %BLEU score for the source SAITG and 1.79 for the target SAITG). The differences in the performance of the system for the source or target SAITG are not significant.

Table 2. Results of the experimentation in %BLEU score

System	%BLEU
Baseline PBT	41.1
Initial ITG	41.23
Re-estimated ITG	41.79
Source SAITG	42.85
Target SAITG	43.02

Figure 4 show the comparison between the output of the PBT decoder, the output of the SITG-based decoder with a target SAITG and one of the references. In the first sentence, it can be seen how the SITG-based system output is syntactically better formed than the PBT output sentence. The reordering of the PBT systems is usually guided by the language model and sometimes by lexical reordering tables. Such information is sometimes not enough. This fact can be seen in the second sentence. The PBT system changed the order of the numbers, while the SAITG has learned that numbers must not be combined in inverted order. The rules of the SITG involved in this reordering and their probabilities

PBT	this one and what 's the difference between ?
SAITG	what 's the difference between this with that ?
Reference	how is this one different from that one ?
PBT	call mr. three four one four five six seven .
SAITG	number of s. three six four five seven four one .
Reference	the number for s nicholas is three six four five seven four one .
PBT	can i go to the front row ?
SAITG	is it okay to the front row ?
Reference	can i go up to the front ?

Fig. 4. Comparison of reference, PBT and SAITG translation outputs for several sentences

$$\Pr(\text{QP} \rightarrow [\text{CD CD}]) = 0.161$$
$$\Pr(\text{QP} \rightarrow \langle\text{CD CD}\rangle) = 0.036$$
$$\Pr(\text{QP} \rightarrow [\text{CD QP}]) = 0.273$$
$$\Pr(\text{QP} \rightarrow \langle\text{QP CD}\rangle) = 0.058$$

Fig. 5. Rules involved in the reordering of numbers. QP and CN are non-terminal symbols that mean *quantified phrase and cardinal number, respectively.*

are shown in Figure 5. Note that the probability for the straigh combination is higher than for the inverse, in such rules.

Despite of not having significant better results, we can obtain the parse tree of the translated sentence and use it to improve the results. For example, we can obtain the n-best translations and their parse trees and rescore them to get a better translation. Another option is to use the target language parse tree with a syntax error correction post-process. Figure 6 shows the output of the system. A syntax correction post-process can detect the lack of a verb in the sentence and correct it.

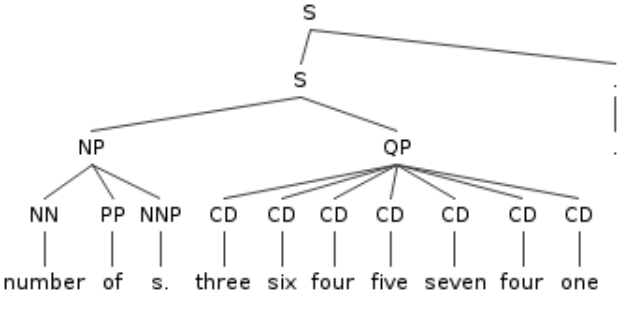


Fig. 6. Example of translated sentence with its target language linguistic parse tree

6 Conclusions

In this work we have presented a SITG-based machine translation decoder and a method to train a Syntax Augmented Inversion Transduction Grammar from a bilingual corpus. The resulting SAITG can use syntactic information either from the source or the target language. The experiment carried out over a Chinese-English corpus showed that the SITG-based decoder with a SAITG obtain better results than a state-of-the-art phrase-based decoder or the SITG-based decoder without syntactic information. There is no significant differences in the system performance when using a source SAITG or a target SAITG. However, when using a target SAITG we can obtain the target language parse trees of the translated sentences and use them in a syntactic rescoring or error correction system. We plan to test these kind of systems in further works.

Acknowledgements. Work supported by the European Comission (FEDER/FSE) and the Spanish MEC/MICINN under the MIPRCV Consolider Ingenio 2010 research programme (CSD2007-00018), the iTransDoc project (TIN2006-15694-CO2-01) and iTrans2 (TIN2009-14511). Also supported by the Generalitat Valenciana under grant Prometeo/2009/014 and the FPI scolarship BFPI/2007/117.

References

1. Koehn, P., Och, F.J., Marcu, D.: Statistical phrase-based translation. In: NAACL 2003: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology. Association for Computational Linguistics, pp. 48–54 (2003)
2. Callison-Burch, C., Fordyce, C., Koehn, P., Monz, C., Schroeder, J.: Meta-evaluation of machine translation. In: Proceedings of the Third Workshop on Statistical Machine Translation, Columbus, Ohio, June 2007. Association for Computational Linguistics, pp. 70–106 (2007)
3. Hassan, H., Sima'an, K., Way, A.: Supertagged phrase-based statistical machine translation. In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (2007)
4. Zollmann, A., Venugopal, A.: Syntax augmented machine translation via chart parsing. In: Proceedings on the Workshop on Statistical Machine Translation, New York City, June 2006. Association for Computational Linguistics, pp. 138–141 (2006)
5. Xiong, D., Liu, Q., Lin, S.: Maximum entropy based phrase reordering model for statistical machine translation. In: Proceedings of COLING-ACL 2006 (2006)
6. Wu, D.: Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. Computational Linguistics 23(3), 377–403 (1997)
7. Och, F.J., Ney, H.: Discriminative training and maximum entropy models for statistical machine translation. In: ACL, pp. 295–302 (2002)
8. Kasami, T.: An efficient recognition and syntax-analysis algorithm for context-free languages. Scientific report AFCRL-65-758, Air Force Cambridge Research Lab., Bedford, MA (1965)

9. Sánchez, J.A., Benedí, J.M.: Obtaining word phrases with stochastic inversion transduction grammars for phrase-based statistical machine translation. In: Proc. 11th Annual conference of the European Association for Machine Translation, Oslo, Norway, June 2006, pp. 179–186 (2006)
10. 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: Annual Meeting of the Assoc. for Computational Linguistics, Prague, Czech Republic, pp. 177–180 (2007)
11. Och, F.J., Ney, H.: A systematic comparison of various statistical alignment models. Computational Linguistics 29(1), 19–51 (2003)
12. Zaidan, O.F.: Z-MERT: A fully configurable open source tool for minimum error rate training of machine translation systems. The Prague Bulletin of Mathematical Linguistics 91, 79–88 (2009)
13. Levy, R., Manning, C.: Is it harder to parse chinese, or the chinese treebank? In: ACL 2003: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics, Morristown, NJ, USA, pp. 439–446. Association for Computational Linguistics (2003)
14. Stolcke, A.: Srilm – an extensible language modeling toolkit. In: International Conference on Spoken Language Processing (2002)
15. Papineni, K., Roukos, S., Ward, T., Zhu, W.-J.: Bleu: a method for automatic evaluation of machine translation. In: ACL 2002: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, Morristown, NJ, USA, pp. 311–318. Association for Computational Linguistics (2001)

Syntactic Structure Transfer in a Tamil to Hindi MT System – A Hybrid Approach

Sobha Lalitha Devi, Vijay Sundar Ram R, Pravin Pralayankar, and Bakiyavathi T

AU-KBC Research Centre
MIT Campus of Anna University, Chennai-600044
sobha@au-kbc.org

Abstract. We describe the syntactic structure transfer, a central design question in machine translation, between two languages Tamil (source) and Hindi (target), belonging to two different language families, Dravidian and Indo-Aryan respectively. Tamil and Hindi differ extensively at the clausal construction level and transferring the structure is difficult. The syntactic structure transfer described here is a hybrid approach where we use CRFs for identifying the clause boundaries in the source language, Transformation Based Learning (TBL) for extracting the rules and use semantic classification of Postpositions (PSP) for choosing semantically appropriate structure in constructions where there are one to many mapping in the target language. We have evaluated the system using web data and the results are encouraging.

1 Introduction

One of the central design questions in machine translation is the syntactic structural transfer, which is the conversion from a syntactic analysis structure of the source language to the structure of the target language. Here we describe the syntactic structure transfer in machine translation that uses machine learning and linguistic rules to arrive at the correct structure transfer. In transfer based machine translation, structural transfer plays a major role on translation output. The identification of structures present in a language is a difficult task and selecting syntactically and semantically appropriate structure in the target language for a given source language is much more complex.

There are several approaches to structural transfer – the interlingua, transfer grammar, and direct transfer [12]. Stat-XFER is a general search based and syntax- driven framework for developing MT system under a variety of data conditions [7]. It uses two language dependent resources: a grammar of weighted synchronous context free rules and a probabilistic bilingual lexicon of syntax-based word and phrase level translation. SMT systems in general use aligned parallel corpus for correct choice of lexical and structural transfers. The recent approaches to SMT [10,6] also attempts to improve some of its shortcomings by incorporating syntactic knowledge in the translation process. Statistical parsers can provide the syntactic information that is necessary for linguistic generalization and for the resolution of non local dependencies. This information source is deployed in recent work either for pre-ordering source

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 438–450, 2010.

sentences before they are input to a phrase-based system [14,3] or for re-ordering the output of translation models by statistical ordering models that access linguistic information on dependencies and part-of-speech [8,5,2].

In this paper we would like to step away from the lexical and phrase transfer and study how syntactic structure can be transferred with correct semantic interpretation and in doing so investigate the possible contributions of incorporating a statistical based syntactic structure into a rule based system. The statistical components of our system are modeled on shallow parsed structure at clause level. In contrast to phrase-based and dependency based SMT approach we use a transformation based learning algorithm to determine the target syntactic structure. The goal of this syntactic structure transfer is to improve the grammaticality of translation and to give the naturalness to the target language structures. The paper is designed as follows. In the next section we give our approach in detail. The third section describes about the clause boundary identification using CRFs, and also deals in detail the rule learning from the parallel clause aligned corpus using Transformational Based Learning (TBL). The semantic classification of Postpositions (PSP) and Case Markers (CM) and how it helps to improve the structure selection is discussed in section three and the fourth is the conclusion.

2 Our Approach

The fundamental principles behind the design of our transfer grammar are that it is possible to learn structures from parallel data which is clause identified, and that the correct rule in target language can be selected for a given source language rule using semantic classification of PSP and the classification of CM a noun takes. Here we take two languages Tamil a Dravidian language and Hindi an Indo-Aryan language. Though the two languages have similarity at the lexical level due to the influence of Sanskrit, structurally, they are very different and this difference is more in select clausal constructions such as relative participle, complement and conditional clauses. Both the languages are similar in the following features: verb final, relatively free word order, morphologically rich in inflections and dissimilar: agglutination where Tamil is agglutinative and Hindi is not.

The present MT system on Tamil to Hindi uses a combination of rule based and machine learning (ML) approach which comprises of eight modules and they are Morphological Analyzer, POS Tagger, NP and VP chunker, NER, WSD, Transfer Grammar and Word Generator. In the transfer grammar module the syntactic structure of the source language is mapped to the target language structure. Initially we had used a rule based approach and found that many constructions could not be handled since there is one to many mapping in relative participle and conditional clause constructions. Hence we came up with a new approach. The transfer grammar module which is dealt in this paper has three sub-modules, a) to identify the clause boundary in a sentence, b) to identify the parallel structures, and c) to choose the exact structure using semantic mapping rules from the structures identified by the previous sub-module. In clause boundary identification, the start and the end of a clause are identified using a ML algorithm with linguistic features. The parallel structure sub-module identifies whether there is a parallel structure between the source and target language

and the semantic mapping module capture the exact structure and transfer the sentence into target language. The translated final output is getting generated at the word generator level.

Our new method for mapping the correct syntactic structure transfer rules operate on paired sentences of a parallel bilingual corpus which is not aligned for words or phrases but for clause boundaries. Here we identify the clause boundaries in a sentence in both source and target sentences and are marked for the start and end of the clauses. We consider this marking as clause level alignment and use this corpus for the identification of the syntactic structure of the sentences. The equivalent syntactic structure is identified for each type of clause in source and target language using TBL. In the rest of the section we give a brief description about the complex nature of the clausal constructions in Tamil and Hindi. The structure of each clausal construction in Tamil and Hindi differs and it is illustrated by the following examples.

(1) Ta. netru [vantha] paiyan raaman aakum
Hi. [jo] ladka kal aaya dha [vo] ram hai
'The boy who came yesterday is Ram'

In sentence (1).Ta. the clause is identified by a non-finite verb "vantha" "come + past + relative participle (RP)" where as in 1.Hi. it is "jo–vo" that identify the clause. That is, in Tamil it is the non-finite verb and in Hindi it is the relative-correlative (RC) which flag the relative participle clause. The start boundary of the clause is identified by the adverb that precedes the non-finite verb in 1.Ta. and the end boundary is identified by the noun that follows the non-finite verb. In1.Hi. the constituents between the RC "jo-vo" along with RC form the start (jo) and end (vo) of the clause. The structure of conditional clauses is also like the RP clause construction where, Tamil has a non-finite verb with a conditional suffix and Hindi has a RC as in 2.Ta and 2.Hi.

(2) Ta. nii [vanthaal] naan viittukku pookuveen
Hi. [agar] thum avogi [tho] main ghar jaavogii
'If you came I will go home'

In 2.Ta. "vanthaal" "if comes" is the non-finite conditional verb and in 2.Hi "agar-tho" is the RC. Depending on the PSP or CM of the noun that follows the non-finite verb in Tamil, the RC is selected in Hindi. If the noun is nominative then the RC is "jo-vo" and if it is PSP then it is "jis-us". Coming to complement clause the structure is similar in both languages but in Tamil, clause inversion is possible, where the main clause and subordinate clause can be moved. This is not possible in Hindi and is explained below:

(3) Ta. raaman connaar sita varuvaal [enru]
Hi. ram ne kaha [ki] sita ayegii
'Ram said that sita will come'

"enru" and "ki" are the complementizer markers similar to "that" in English. In Tamil the complement clause moves infront of the main clause and also can be embedded between the subject of the main clause and the finite verb. The examples 3a, b and c illustrate the above syntactic variation in Tamil.

(3) a. Ta. sita varuvaal [enru] raaman connaar
b. Ta. raaman connaar sita varuvaal [enru]
c. Ta. raaman sita varuvaal [enru] connaar
'Ram said that Sita will come'

In the case of relative participle and conditional clause construction it is one to many mapping whereas in complement clause it is many to one mapping in Tamil to Hindi transfer. To identify the correct structure in these cases we use a mapping rule, which is triggered by linguistic information. For the one to many mapping we identify the triggering variable in Tamil which select the particular RC in Hindi. The triggering variables are the PSP and the CM. We have classified the PSP and CM according to the construction it takes in Hindi.

2.1 Corpus Selection for Training

The corpus required for the training of clause boundary identifier and for learning the rules is obtained through controlled elicitation method [11]. We have used this method because our requirement of corpus is for the syntactic structure learning and not the lexical learning. The design of the corpus is based on the elicitation principles of field linguistics and the variety of sentences cover a wide range of linguistic phenomena. Thus we have selected 3000 sentences in Tamil. A bilingual informant translated the sentences to Hindi and the clause marking was manually done. The sentences we elicited are designed to cover all possible constructions and multiple embeddings possible in the source language. In total we have taken 3000 sentences in Tamil and Hindi. These sentences are automatically morphologically anlaysed, POS tagged, phrase chunked and clause boundary marked. The tags we used for clause boundary marking are: relative participle (RP), conditional (COND), complement clause (COMP), infinitive clause (INF), non-finite clause (NF) and main clause (MCL).

3 Clause Identifier and Rule Learning

We identify the clause boundaries using CRFs and the syntactic structure rules are learned using TBL. The clause boundary identifier, the rule learner and the mapping algorithm are explained in detail in this section.

3.1 Clause Identifier

We have built the clause identifier using CRFs, a machine learning technique. Here we identify the clause boundary of the source language alone. In CRFs technique we have used grammatical rules as one of the major feature. The given sentence is preprocessed for part-of-speech and chunking information. The words in the sentence are analysed with a morphological analyzer. We replace the noun phrases in the sentence with a token "NP" after preprocessing the sentence, retaining the morphological information of the head noun. The clause identifier has to learn the sentence structures. How the CRFs is used for this task is explained below.

CRFs is undirected graphical model where the conditional probabilities of the output are maximized for a given input sequence [9] and it makes a first-order Markov independence assumption and thus it is a conditionally-trained finite state machine (FSMs). It has all the advantages of Maximum Entropy Markov Model (MEMMs) but solves label bias problem which is the disadvantage of MEMMs. The mathematical functions involved in CRFs is explained in [9].

To build the language model, CRFs requires the set of features to be defined. Using the features iteratively the model is built. The CRFs used is the CRF++ available in open source [13].

The features used are of two types, word level and structural level. At the word level we have considered the lexical word, its part-of-speech and chunk and have taken a window of size five. The structural level features are the grammatical rules. Sample grammatical rules and the way it is added in the column is described below.

Rule 1: To get the relative participle clause boundary end

```
-1 VM+RP = 1
 0  NP = 1        RP
 1   PSP = 0
```

If the current token is a NP, the previous is a relative participle verb (RP) and next word is not a PSP then the current NP is marked as probable RP clause end.

Rule 2: To get the relative participle clause boundary end

```
-1  VM+RP = 1
 0   PSP = 1        RP
```

If the current token is a PSP, the previous is a relative participle verb then the current PSP is marked as probable RP clause end.

Rule 3: To get the conditional clause boundary

```
0 VM+CON = 1 CON
```

If the current verb has a conditional marking suffix, then the current verb is marked for probable conditional clause end.

Once these rules are run, the probable clause start positions are marked based on the probable clause end marked with numbers. Consider the following sentence in example 4 as the input sentence to the clause identifier system. Here we show the output after the preprocessing stage.

(4) Ta. mazhaiyil cendrathaal naan nanaitheen.
rain+loc go+past+CONT I (became wet)
'Since I went in rain I was wet.'

The output is as shown below. Here '2 and -2' refers to the probable conditional clause start and end markers and '6,-6' refers to the main clause.

```
NP  NP  N_loc   2
ceVnYrYawAl VM_COND V_COND  -2
NP  NP  pn_nom  6
nanYEnwenY  VM_VGF  V_VGF   -6
.   SYM I-VGF   o
```

3.1.1 Evaluation and the Results

We have trained the system with 1800 sentences. The sentences are tagged with relative participle, conditional, non-finite, infinitive, complement and main clause. The distribution of the clauses is as follows. There were 1512 relative participle clauses, 732 non-finite clauses, 715 conditional clauses, 402 infinitive clauses and 172 complement clauses. These sentences are preprocessed for POS and chunking information and the words are morphological analyzed. In these sentences the lexical item of the noun phrases are replaced with "NP" symbol with grammatical features of the head noun. The system is tested with 1200 unseen sentences taken from a Tamil News paper corpus. The evaluation of the system is tabulated and given in Table1.

Table 1. Performance in percentage

Clause	Total Number of Clauses	Open Tag				Close Tag				Total Correct clause	Total Correct clause %
		Correct		Wrong		Correct		Wrong			
		No of Clause	%	No of Clause	%	No of Clause	%	No of Clause	%		
RP	453	373	82.34	80	17.66	349	77.04	104	22.96	312	69.03
NF	286	251	87.76	35	12.24	268	93.71	18	6.29	239	83.56
COND	139	122	87.77	17	12.23	121	87.05	18	12.95	115	82.73
INF	69	53	76.81	16	23.19	60	86.96	9	13.04	51	73.91
COMP	53	33	62.26	20	37.74	32	60.38	21	39.62	29	54.76
MCL	1200	976	81.33	224	18.67	1136	94.67	64	5.33	912	76
Total	2200	1808	79.71	392	20.29	1966	83.30	234	16.70	1658	73.34

3.1.2 Discussion

From Table 1 it is evident that the system has identified the conditional clause with high accuracy. In the case of relative clause, generally the clause end with the noun phrase or before the noun phrase. But there are cases where instead of noun phrase a possessive noun phrase follows with a PSP and then followed by a noun phrase as in 5.Ta. The finite verb, which forms the part of the main clause with the shared noun phrase in the relative clause, agglutinates with the noun phrase itself as in Ta.6. The agglutination of noun, verb and PSP occur commonly. This affects the proper identification of the clause boundaries. The complement clause is indicated by "enna" and "enru". Though these two words can occur in three different positions in a sentence as described in the earlier section, the sentence where it occurs as a complementizer, clause boundary can be easily identified by the preceding finite verb. The system fails in identifying the starting of the complement clause as the distance between the end of the clause and the starting of the clause is more. The CRFs is not efficiently learning the long distance between the start and end boundaries of clause.

(5) Ta. kovilil ulle amainthulla malaiyin mithulla itaththil
temple+loc in located +RP hill+poss on top of place+loc
katavulin cilai iruikirathu.
God+poss idol is present+finite verb

'The God's idol is present on top of the mountain, which is located inside the temple.'

(6) Ta. alakiya malaikalai konta inthiyavin oru ciriya nagaramaakum

beauty+adj hills+acc have+RP India+pos one small+adj city+ finite verb.
'A small city in India, which has beautiful hill.'

The clause opening tag is identified with 81.58% and the closing tag is identified with 87.37%. As the training set had more relative participle clause, occurring at the starting of the sentence, the correctness of the open RP clause tag is more than the closing RP clause tag. The overall performance of the system is with 74.74%.

3.2 Rule Learning Using Transformation Based Learning (TBL)

Here we use TBL technique for learning all possible structural rules from the clause boundary marked sentences. Transformation-based error-driven learning commonly referred as transformation-based learning, is an automatic machine learning technique, whose output is an ordered list of rules. This approach is error-driven because the transformations learned at each step of the iteration are those that lead to the greatest reduction in errors when compared with the training data.

This technique is used in many natural language processing applications; the best known application for TBL is part-of-speech tagger by Eric Brill. Other tasks, where TBL is applied are resolving syntactic attachment ambiguities, syntactic parser [1], text chunking, word sense disambiguation and in ellipsis resolution. In machine translation system the TBL is used at various stages of translation as it is used for learning word level transfer rules [4]. The central idea of TBL is to learn an ordered list of rules, which progressively improves upon the current state of the training set. To learn a model, baseline rules are applied on each sentence in the training corpus. From those sentences, where this baseline prediction is not correct, candidate rules are generated using the features defined. Those candidate rules are then tested against the rest of the corpus to identify how many negative changes they can cause. The scoring of the rules is based on the changes and those rules, whose score is maximal, are selected as learned rules. An ordered set of rules is learned by repeating this process. We choose TBL for the structure learning task mainly because:

a. An ordered set of structural rules, with no bias can be obtained using a hand crafted base rule.
b. The obtained rules are linguistically motivated and are understandable to human and machine.
c. Structural rules are not learned from the erroneous clause marked sentences.

As discussed earlier, TBL is used to generate all possible structural rules from a set of clause boundary marked sentences. The structures of the sentences, which vary from the baseline rule, are identified. Candidate rules are formed from those sentences using the specified features and to score those candidate rules, they are compared against the same set of clause boundary marked input sentences. Those candidate rules that are generated from erroneous clause marked sentences receive a very low score and they are filtered out. Thus we get an error free set of structural rules. In this task, the unique list of structural rules also include those structural rules, which matches with the baseline rule, to get a list of all possible rules. Here we have used TBL to learn rules from Tamil and Hindi sentences. The rule learning process for

Tamil and Hindi are done separately. The baseline rules and the features defined are different and are explained in detail.

3.2.1 Tamil TBL

The input sentences to the Tamil TBL have 3000 sentences containing three different sets. The first set contains 500 sentences having a subordinate and main clause, the second set with 1500 multiple embedded clause sentences and third set with 1000 sentences of simple and compound sentences. These sentences are pre-processed with morphanalyser, part-of-speech tagger, chunker and clause Identifier. The base line rule used by TBL for Tamil is as follows: a clause with a non-finite verb followed by main clause.

{clause} non-finite {/clause} {main clause} {/main clause}

Success of the TBL lies on the features used in generating the candidate rules. The features considered are clause boundary markers, suffix and the morphological analysis of the verbs. After the training phase of the TBL, it was observed that in the set of rules generated, a) the number of rules from the first set of sentences is very low and it also matches with the baseline rule, b) As the second set of sentences has multiple clauses in each sentence the number of rules learned is very high and c) The number of rules learned from the third set is low as sentences in the corpus are more structurally similar. The statistics of the rules learned is shown in Table 2. A sample set of rules learned by the Tamil TBL is shown in Table 3.

Table 2. Number of rules learned by Tamil TBL

Type	Number of sentences	Number of rules learned
Sentence with subordinate and main clause	500	8
Multiple embedded clause	1500	346
Simple and Compound	1000	5
Total	3000	358

Table 3. Sample list of rules from Tamil TBL

Rules satisfying the baseline rule	New rules
{RP} nw_a/v+past+rp {/RP} {MCL} {/MCL}	{NF}i/v+vbp{/NF}{NF}wwu/v+vbp{/NF}{NF}i/v+vbp{/NF}{NF}wu/v+vbp{/NF}{RP}ww_a/v+rp{/RP}{MCL}{/MCL}
{CON}Alum/v+cond+conc {/CON} {MCL} {/MCL}	{INF}a/v+inf{/INF}{RP}nw_a/v+rp{/RP}{MCL}{/MCL}

3.2.2 Hindi TBL

The Hindi rules are learned from the translated 3000 Tamil sentences. Clausal boundaries are marked to these translated sentences. In Hindi, as the presence of the clause is identified by the relative-corelative markers, it is used as the features for

generating the candidate rules. For learning the Hindi rules, the baseline rules used in the TBL are as follows

{clause} jo/RP vo/RP{/clause} {main clause} {/main clause}
{clause} agara/COND tho/COND {/clause} {main clause} {/main clause}

Hindi TBL learned 16 rules from the first 500 sentences. Similarly the number of rules learned from the second set of 1500 sentences (multiple embedded clausal sentences) is 682. And the number of rules learned from the last set of 1000 sentences is similar to Tamil TBL. The number of rules learned by the Hindi TBL is given in Table 4. A sample set of rules, learned from the Hindi tagged sentences are give in Table 5.

Table 4. Number of rules learned by Hindi TBL

Type	Number of sentences	Number of rules learned
Sentence with subordinate and main clause	500	16
Multiple embedded clause	1500	682
Simple and Compound	1000	5
Total	3000	710

Table 5. Sample list of rules from Hindi TBL

Rules satisfying the baseline rule	New rules
{COND} agara/COND wo/COND {/COND} {MCL} {/MCL}	{NF} {/NF} {NF} {/NF} {NF} {/NF} {MCL} {/MCL}
{RP} jo/RP vo/RP {/RP} {MCL} {/MCL}	{CON} cuMki/COND {/CON} {MCL} {/MCL}
	{INF} {/INF} {MCL} {/MCL}
	{RP} jo/RP una_ke_liya{/RP}{MCL} {/MCL}

As different RC markers are used to denote one type of clause in Hindi, the number of structural rules learned in Hindi is higher than that in Tamil. The number of rules, satisfying the baseline rule is more in Tamil than in Hindi, because the baseline rule used in Tamil is more generic and it is specific in Hindi. Since the clausal structures in Hindi are introduced by RC markers, the baseline rule cannot be generic.

3.3 Semantic Classification of Postpositions and Case Markers

The output from TBL gives a set of clausal rules for the source and the target language. Where there is one to one mapping between the rules, the assignment of the structure is straight forward, whereas, when rules have one to many mapping, the selection of the exact structure requires more information. The structure in Hindi has only one form where it uses a RC for a RP construction, but the selection of the correct RC depends on the semantic features of PSP and CM that follows the non-finite verb in Tamil. While analyzing the two languages, it is observed that the selection of

target language structure is triggered by the semantic feature of the constituent following the non-finite verb (clausal verb) within the clause boundary, in the source language sentence. The constituents that can follow the non-finite verb are the PSP, noun inflected by CM and it can also be an empty category. Depending on the semantic feature, such as time, space, and manner of the above constituents the structure selection is made. The PSP and CM are classified into 8 major semantic features and each feature is further classified. It was also observed that if there is a clausal construction in source language, it need not be a clausal construction in target language. The constructions which have one to many mapping are the relative participle and conditional clauses. For one structure of RP in Tamil there are 13 different forms of RC in Hindi. The list of PSP and CM in Tamil and their corresponding RC in Hindi is given in Tables 6 and 7.

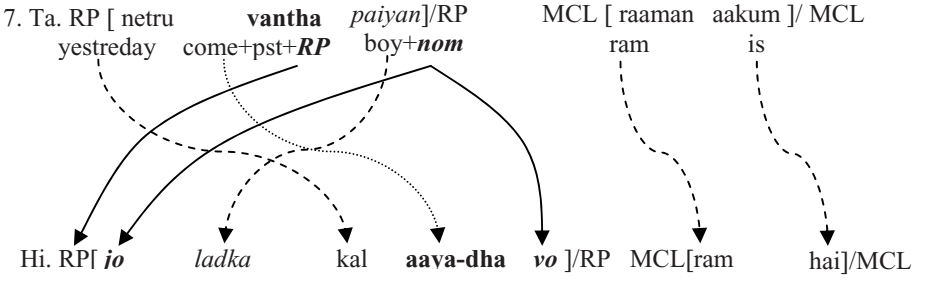
Table 6. Postpositions

PSP in Tamil	Meaning in English	RC in Hindi
poothu	When	jab…thab
pinner	After	simple sentence
munnar	Before	simple sentence
utane	As soon as	jaise hii.....vaise hii
utan	Once	jaise.....vaise
varai	Till	jabthak.....thabthak
varai (distance)	Till	jahaanthak….vahaanthak
poothilum, kuuta	Even though	agar (noun/pronoun) bhi-tho

Table 7. Case Markers

Case marker in Tamil	RC in Hindi
N+nom (singular)	jo….vo
N+nom (plural)	jo….ve
N+dative (singular)	jo….us
N+dative (plural)	jo….us
N+ablative (singular)	jis….us
N+ablative (plural)	jin….un

The following example illustrates how semantic features select the clause structure.



'The boy who came yesterday is Ram'

The presence of the non-finite verb with the RP suffix will trigger the RP clause construction and it is the nominative noun following the non-finite verb which selects the RC ie whether it is “jo-vo” or “jab-thab”. The semantic classification of PSP we have arrived at is given in (Fig 1) and the description of PSP in Table 8:

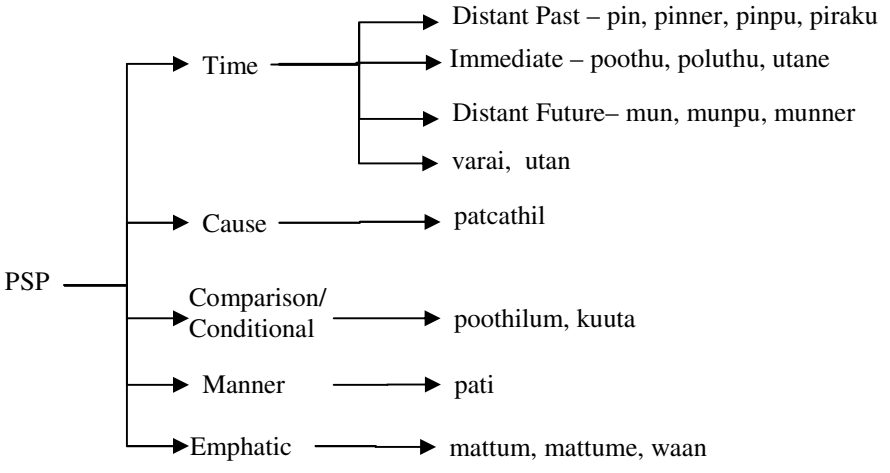


Fig 1. Semantic feature classification of Postpositions (PSP)

Table 8. Semantic Description of PSP

No	Semantic type of PSP	Examples	Description in English
1	Time	Distant Past – pin, pinner, pinpu, piraku	To describe the distant past time, such as ‘After’
		Immediate – poothu, poluthu, utane	To describe the immediate action such as ‘As soon as’, ‘When’
		Distant Future – mun, munpu, munner	To describe the distant future such as ‘Before’,
		varai, utan	To describe the time such as ‘Till’, ‘Once’
2	Cause	patcathil	Describes the causative such as ‘If at all’
3	Comparison/ Conditional	poothilum, kuuta	To describe the condition, such as ‘If’, ‘Though’
4	Manner	pati	Describes the manner such as ‘As’
5	Emphatic	mattum, mattume, waan	Describes the emphasis such as ‘Only if’

From the analysis we arrive at the following.

In structure transfer between two language families, wherein one has the clause marked by a non-finite verb (as in Tamil) and in other it is marked using a RC (as in Hindi), then the correct structure selection depends on

a. The immediate constituent following the non-finite verb within the clause boundary.
b. The immediate constituent following the non-finite verb can be a PSP, CM or empty category.
c. The semantic features of the PSP, CM decides the type of RC

We evaluated the semantic classification mapping with 3000 sentences and the results are given in Table 9.

Table 9. Evaluation of Semantic classification mapping

Semantic Class in RP and Conditional clause	**Number of Occurrence**	**Erroneous input (Errors introduced by prior modules)**	**Number of Correctly Triggered**	**Precision % Erroneous Input**	
				With Error	**Without Error**
Non-finite Verb + PSP	85	18	53	62.5	83.34
Non-finite Verb + Case marker	859	131	687	80	95.24
Non-finite Verb + Null	498	72	360	72.23	86.67
Total	1442	221	1100	76.28	90.09

Here we have identified the semantic features of PSP and CM which select the semantically correct clausal structures in target language. The results are encouraging and could get the correct target language construction. Our analysis also found that in certain constructions the semantic classification of verb is necessary for the correct selection of the structure. Semantic classification of verb is not with in the scope of this paper. Similarly another feature which influences the selection of RC is the empty category after the non-finite verb. We have not dealt with this feature in this paper.

We have checked whether this is applicable to translations involving other families of languages and found that the semantic classification of PSP and CM is needed in selecting semantically correct structure in translating from Tamil to English also. In general it is seen that in translation, the structure transfer from a language having the grammatical feature of non-finite verb form for clausal construction requires the semantic classification of PSP for selecting syntactically and semantically correct structure in target language. This is seen in translation from Dravidian languages (Tamil, Telugu, Malayalam, Kannada) to Indo-Aryan (Hindi, Bengali, Marathi) and Indo-European (Sanskrit, English).

4 Conclusion

The paper is about the syntactic structural transfer in a machine translation system. Here we have used a clause boundary identifier developed in CRFs, a rule learning system using TBL and appropriate rule choice using semantic classification of PSP and CM. The analysis of PSP and CM for identifying the correct structure is a new approach and the results are encouraging.

References

1. Brill, E.: Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part-of-Speech Tagging. Computational Linguistics 21(4), 543–566 (1995)
2. Chris, Q.: Arul Menezes, and Colin C.: Dependency tree let translation: Syntactically informed phrasal smt. In: Proceedings of the 43rd ACL
3. Collins, M., Koehn, P., Ivona, K.: Clause restructuring for statistical machine translation. In: ACL, Ann Arbor,MI, pp. 531–540
4. Dien, Z.D., Ngan, T., Quang, X., Nam, C.: A hybrid approach to word-order transfer in the english – vietnamese machine translation system. In: Proceedings of the MT Summit IX, Louisiana, USA, pp. 79–86 (2003)
5. Ding, Y., Palmer, M.: Machine translation using probablisitic synchronous dependency insertion grammars. In: Proceedings of the 43rd ACL
6. Koehn, P., Josef, O.F., Marcu, D.: Statistical Phrase-Based Translation. In: Proc of HLT/NAACL 2003, pp. 127–133 (2003)
7. Lavie, A.: Stat-XFER: A general search-based syntax-driven framework for machine translation. In: Gelbukh, A. (ed.) CICLing 2008. LNCS, vol. 4919, pp. 362–375. Springer, Heidelberg (2008)
8. Lin, D.: A path-based transfer model for machine translation. In: Proceedings of the 20th COLING 2004 (2004)
9. McCallum, A., Li, W.: Early results for named entity recognition with conditional random fields, feature induction and web enhanced lexicons. In: Proceedings of CoNLL 2003, Edmonton, Canada, pp. 188–191 (2003)
10. Och, F.J., Tillmann, C., Ney, H.: Improved Alignment Models for Statistical Machine Translation. In: EMNLP (1999)
11. Probst, K., Levin, L.: Challenges in Automated Elicitation of a Controlled Bilingual Corpus. In: Proceedings of TMI (2002)
12. Slocum, J.: Machine Translation: its history, current status, and future prospects. In: Proceedings of the 10th international conference on Computational linguistics, Stanford, California, July 02-06, pp. 546–561 (1984)
13. Kudo, T.: CRF++, an open source toolkit for CRF (2005), `http://crfpp.sourceforge.net`
14. Xia, F., Michael, M.: Improving a statistical MT system with automatically learned rewrite patterns. In: COLING 2004 (2004)

A Maximum Entropy Approach to Syntactic Translation Rule Filtering

Marcin Junczys-Dowmunt

Adam Mickiewicz University
Faculty of Mathematics and Computer Science
ul. Umultowska 87, 61-614 Poznań, Poland
junczys@amu.edu.pl

Abstract. In this paper we will present a maximum entropy filter for the translation rules of a statistical machine translation system based on tree transducers. This filter can be successfully used to reduce the number of translation rules by more than 70% without negatively affecting translation quality as measured by BLEU. For some filter configurations, translation quality is even improved.

Our investigations include a discussion of the relationship of *Alignment Error Rate* and *Consistent Translation Rule Score* with translation quality in the context of Syntactic Statistical Machine Translation.

1 Introduction

A crucial step when preparing a Syntactic Statistical Machine Translation system involves extracting a large set of translation rules from a bilingual word-aligned corpus. Even small errors in the alignment data may lead to the extraction of many wrong rules that can seriously affect translation quality. The majority of approaches designed to prevent "rogue rules" relies on methods that improve word alignments so they become more consistent with the given syntactic data, examples being [1,2]. As a result, the number of translation rules usually increases, but many of these rules are still incorrect or unlikely to be used in any translation. On the other hand, a reduction in the number of rules (e.g. by frequency thresholds or phrase length limitations) might cause a decrease in translation quality.[1] However, adhering to many possibly redundant translation rules results in greater requirements concerning resources and processing time.

Instead of tuning a single word alignment towards generating better rules, we extracted translation rules from several word alignments which have been created with different tools and combination methods. These rules were scored and discarded if they failed to achieve a predetermined threshold. This score is the probability that a rule represented by a set of features belongs to a class of correct rules as calculated by a Maximum Entropy (ME) model. This ME model learns to distinguish between correct and incorrect rules by being trained on a set

[1] This has been shown by [3] in the context of Phrase-Based, Hierarchical Phrased-Based and Syntax-Augmented SMT.

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 451–463, 2010.

of reference rules extracted from manually word-aligned sentences. Our decision to use many input word alignments instead of a chosen single word alignment is motivated by the increased coverage of correct rules that can be achieved this way. We show that it is possible to reduce the number of translation rules with this simple supervised machine-learning approach by 60–70% without sacrificing translation quality as measured by BLEU and NIST. Actually, for some filter settings, the translation quality is even higher than for the unfiltered rule sets.

Part of our investigations comprises a short discussion of the relationship of *Alignment Error Rate* (AER) and *Consistent Translation Rule Score* (CTRS) — a metric equivalent to *Consistent Phrase Error Rate* (CPER) [4] adapted to translation rules — with translation quality in the context of Syntactic SMT. Similar questions have been addressed by [4] in the context of Phrase-Based SMT, but only marginally for Syntactic SMT [1].

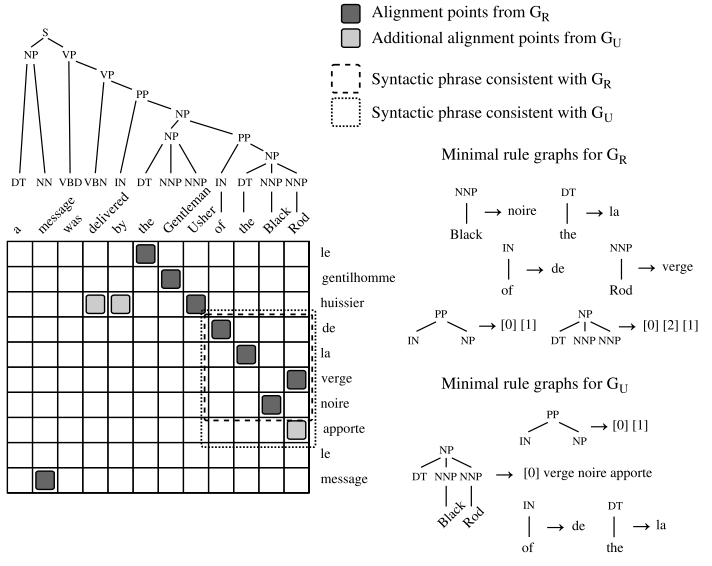
Section 2 reviews the process of translation rule extraction for Syntactic SMT from parallel corpora. Section 3 gives a short introduction to Maximum Entropy Models and details on the features used for the representation of translation rules. In Sect. 4 we compare automatically-created alignments as well as the rule sets generated from these alignments in terms of AER and CTRS. Section 5 gives the results of our filter measured in CTRS, BLEU and NIST. We finish the paper with a discussion of the presented findings.

2 Extraction of Translation Rules

The Syntactic SMT system used in our experiments — Bonsai — is described in [5] and is similar in function to the systems introduced by [6] and [7]. Formally, Bonsai is a tree-to-string transducer [8,9], which requires that the source language is syntactically parsed prior to translation. The parse tree is transformed by translation rules into a flat target language string. This process is guided by a set of probabilistic and heuristic rule features and one or more target language models.

For translation rule extraction, we applied the algorithm proposed by [10]. For a given word-aligned sentence pair and the parse tree of the source language sentence, this algorithm identifies syntactic constituents of the parse tree which are consistent with the word alignment and forms a set of minimal rule graphs. Complex rule graphs can be built from minimal graphs or smaller complex rule graphs by composing source tree fragments at shared nodes and concatenating the target sides of the composed rules. The number of minimal graphs used for the creation of a rule is denoted by k.

Figure 1 illustrates the rule extraction for a sentence pair from the Hansards parallel corpus [11] and two different alignments. These two alignments were created by training GIZA++ in both directions, after which the refined combination method from [12] (denoted by G_R) and union (denoted by G_U) were applied to the directed alignments. Dark gray alignment points belong to G_R and G_U, while light gray points appear only in G_U. The minimal rule graphs extracted from the marked phrases (dashed boxes) differ in number and form between both alignments, a fact which is caused by a single superfluous alignment point from G_U.



Composed rules from G_R	**Alignment**	**Phrases**	**k**
PP(of NP[0]) → de [0]	(1,1)	(8,12,3,7), (9,12,4,7)	2
PP(of the NNP[0] NNP[1]) → de la [1] [0]	(1,1), (2,2)	(8,12,3,7), (10,11,6,7), (11,12,5,6)	4
PP(IN[0] the Black Rod) → [0] la verge noire	(2,2), (3,4), (4,3)	(8,12,3,7), (8,9,3,4)	5
Composed rules from G_U	**Alignment**	**Phrases**	**k**
PP(of NP[0]) → de [0]	(1,1)	(8,12,3,8), (9,12,4,8)	2
PP(IN[0] the Black Rod) → [0] la verge noire apporte	(2,2), (3,4), (4,3), (4,5)	(8,12,3,8), (8,9,3,4)	3

Fig. 1. Rule extraction and composition

A small sample of more complex rules[2] that can be created by composing the minimal graphs is given together with three types of parameters: rule-internal alignments for terminal symbols, rectangles describing phrases-pairs consistent with root nodes and nonterminal symbols, and the composition factor k.

3 The Maximum Entropy Filter

3.1 Maximum Entropy Models

Maximum entropy models estimate the probability $p(c|x)$ of a class c in a context x. Given a set of facts or constraints, a model is computed that follows all of these constraints but otherwise makes as few assumptions as possible [13].

Constraints are represented as feature functions, in most cases binary functions, $f_i : \mathcal{C} \times \mathcal{X} \rightarrow \{0, 1\}$, where $\mathcal{C}$ is the set of all classes and $\mathcal{X}$ denotes the set

[2] The translation rules used in our syntactic MT system differ slightly from the rules proposed in the majority of similar systems [10,6] as we ignore internal nodes and preserve only information about root nodes and leaves.

of all facts. Each feature function f_i is associated with a model parameter λ_i, the feature weight. Given a set of N feature functions $f_1, \ldots, f_N$, the probability of class c given a context x is equal to:

$$p(c|x) = \frac{1}{Z_x} \exp\left(\sum_{i=1}^{N} \lambda_i f_i(c, x)\right) \tag{1}$$

where Z_x is a normalization constant. The contribution (i.e. the weight λ_i) of each feature function to the final outcome can be computed with the *Generalized Iterative Scaling* (GIS) algorithm [14].

When maximum entropy models are used for hard classification, the class $\hat{c}$ that has the highest probability is chosen, i.e.

$$\hat{c} = \arg\max_{c} p(c|x). \tag{2}$$

For our described binary classification problem, we found it more convenient to take advantage of the whole probability distribution over both classes, using the probability of a chosen class as a threshold.

3.2 Rule Features

Translation rules are processed sentence-wise. Quantitative information that go beyond the scope of a single sentence pair are not available. For an approach to filtering based on the quantitative distribution of phrase-pairs in Phrase-Based SMT see [15]. For each sentence pair $(\mathbf{e}, \mathbf{f})$ one or more rule sets R_m exist, where $r \in R_m$ is a single translation rule. Each set R_m has been generated from an automatically created word alignment A_m. We define $\mathcal{R} = \{R_1, \ldots, R_n\}$ as the set of rule sets available for one sentence pair $(\mathbf{e}, \mathbf{f})$. The rule set R_{H} denotes the set of reference translation rules generated from the human-created word alignment. The filter is supposed to select the rules from the rule sets in $\mathcal{R}$ in such a way that the resulting rule set is closer to R_{H} than any of the input rule sets. The set of classes is $\mathcal{C} = \{\text{“good”}, \text{“bad”}\}$, where the respective classes denote the acceptance or rejection of a translation rule.

From the surface form of a translation rule r, the following features can be derived:

- **$\mathbf{R}_m$, RCount:** Whether r exists in a given rule set R_m and the number of rule sets from $\mathcal{R}$ it exists in.
- **SrcSymLen, TrgSymLen, SrcTrgDiff, SrcTrgEq:** The number of source language symbols (terminal and nonterminal) and target language symbols, their absolute difference and signed equality[3].

[3] We define signed equality as $x \gtreqless y = \begin{cases} -1 & \text{if } x < y, \\ 0 & \text{if } x = y, \\ 1 & \text{if } x > y \end{cases}$.

- **NtCount, SrcTrmCount, TrgTrmCount, SrcHasTrm, TrgHasTrm:** The number of nonterminal symbols, the number of source language (target language) terminal symbols, and whether there are source (target) language terminal symbols.
- **Lhs:** The left-hand side symbol of the rule.
- **NtDist$_j$:** For the j-th nonterminal symbol, the absolute distance between the source language position and the target language position in a rule.
- **SrcPuncCount, TrgPuncCount, SrcTrgPuncEq:** The number of punctuation symbols on the source (target) language side and their signed equality.

The following features are collected during the rule extraction process of r:

- **K:** The number k of minimal graphs used for the composition of rule r.
- **SrcSpan, TrgSpan, SrcTrgSpanDiff:** The number of symbols in the source (target) language span and their absolute difference.
- **Align$_m(i,j)$:** For each rule set R_m[4], all alignment points (i,j) from $A(r)$, where $A(r)$ is the set of internal alignments of a rule r. i is the position of the source language symbol, and j the position of the target language symbol in the rule.
- **SrcAligned, SrcUnaligned, TrgAligned, TrgUnaligned:** The number of aligned and unaligned source (target) language words.

The combination of features and feature values results in a large number of feature functions. For the English-French test set there are more than 1,300 different feature functions, while the Polish-French set has over 1,100. The corresponding model parameters λ_i are learned using the *The OpenNLP Maximum Entropy Package*[5].

4 Alignment Data, Rule Sets, and Metrics

The quality of the described filter is evaluated for two language pairs, English-French and Polish-French. The English-French data was made available at the HLT-NAACL 2003 workshop on "Building and Using Parallel Texts: Data Driven Machine Translation and Beyond" [16] and comprises a subset of the Canadian Hansards [11] and a separate test set of 447 manually word-aligned sentences [12]. For the Polish-French experiments we used a subset of the *Directorate-General for Translation – Translation Memory*[6]. A small subset of 294 sentences from this corpus was set apart and manually annotated with the correct word alignments.[7]

[4] As mentioned before, the rule sets have been generated from different alignments. Rules with the same surface may have different internal alignments for different m.

[5] Available at `http://maxent.sourceforge.net`

[6] Available at `http://langtech.jrc.it/DGT-TM.html`

[7] By the moment this paper is published, manual annotation is still work in progress. The data will be made available once the task is finished. To our knowledge this will be the first word-aligned test set with Polish.

Table 1. Data used for filter training

(a) Word-aligned test data

Languages	Sentences	Source	Rules
English-French	497	HLT/NAACL 2003 and [12]	36,846
Polish-French	294	DGT Translation Memory	25,709

(b) Training data for automatic word alignments

Languages	Sentences	Source
English-French	1,130,550	Hansards [11]
Polish-French	748,734	DGT Translation Memory

The data from Tab. 1b is used to compute several automatic word alignments listed in Tab. 2. Apart from GIZA++ and the BerkeleyAligner [1], we also use a close implementation of the supervised word alignment combination method (ACME) proposed by [18], which has been trained on the human-created word alignments and three automatically created alignments (the two directed alignments and BA). In order to reduce data sparseness introduced by the rich morphology of the Polish language, word alignment computation was carried out for a lemmatized version of the Polish-French corpus. The English-French corpus was not preprocessed in this way.

A translation rule set that was created from a given word alignment is identified by the same symbol as its underlying alignment. It should follow from the context whether we refer to the underlying alignment or the generated rule set. English source language parses of all English-French data have been produced with the Stanford Parser [19]. Polish parse trees for the Polish-French data have been created with the internal parser of the Bonsai Syntactic SMT system.

The purpose of the manually word-aligned sentences from Tab. 1a is twofold. Firstly, for each language pair these sentences are used to measure the AER of the automatically created alignments. Secondly, they serve as the basis for the extraction of the reference rule set R_{H} that will be used to train the described maximum entropy model as well as for its evaluation.

Table 2. Automatically created word alignments

Symbol		Description
$\mathrm{G_{EF}}$	$\mathrm{G_{FE}}$	Directed en-fr and fr-en alignments created with GIZA++
$\mathrm{G_{PF}}$	$\mathrm{G_{FP}}$	Directed pl-fr and fr-pl alignments created with GIZA++
$\mathrm{G_I}$		Intersection of the directed word alignments
$\mathrm{G_R}$		Refined [12] combination of the directed word alignments
$\mathrm{G_G}$		Grow-Diag-Final [17] combination of the directed word alignments
$\mathrm{G_U}$		Union of the directed word alignments
BA		BerkeleyAligner [1] joint word alignment model
ACME		A supervised word alignment combination method [18]

Table 3. Comparison of AER for both language pairs

(a) en-fr

Align	Pr	Rc	AER
G_I	98.25	80.16	10.47
G_R	92.39	91.88	7.82
G_G	86.98	94.13	10.33
G_U	85.47	94.85	11.10
BA	90.74	95.99	7.24
ACME	95.47	94.72	4.84

(b) pl-fr

Align	Pr	Rc	AER
G_I	95.60	50.04	34.31
G_R	83.98	64.46	27.06
G_G	76.07	67.41	28.52
G_U	74.11	68.84	28.62
BA	82.98	63.89	27.81
ACME	86.54	75.49	19.36

4.1 Word Alignment Error Rate

The standard metric *Alignment Error Rate* (AER) proposed by [12] is used to evaluate the quality of the introduced input word alignments. AER is calculated as follows:

$$\mathrm{Pr} = \frac{|A \cap P|}{|A|} \qquad \mathrm{Rc} = \frac{|A \cap S|}{|S|} \qquad \mathrm{AER} = 1 - \frac{|A \cap S| + |A \cap P|}{|A| + |S|} \tag{3}$$

where P is the set of possible alignment points in the reference alignment, S is the set of sure alignments in the reference alignment ($S \subset P$), and A is the evaluated word alignment.

The results for all alignment methods have been compiled into Tab. 3. There are large qualitative differences concerning the alignment error rate between both language pairs, which are probably caused by the greater dissimilarity of Polish and French as well as by the characteristics of the utilised test sets. The relative number of possible alignments in the English-French test set is much greater than in its Polish-French counterpart. This makes the English-French test set more forgiving of erroneous alignments.

4.2 Consistent Translation Rule Score

So far we have not defined a formal way to measure the quality of a set of translation rules against the reference rule set R_H. For this purpose, we have adapted the *Consistent Phrase Error Rate* (CPER) from [4] to the needs of syntactic translation rules.[8] To emphasize the application of CPER to syntactic translation rules we have renamed it to *Consistent Translation Rule Score* (CTRS) and calculate it as follows:

$$\mathrm{Pr} = \frac{|R \cap R_H|}{|R|} \qquad \mathrm{Rc} = \frac{|R \cap R_H|}{|R_H|} \qquad \mathrm{CTRS} = \frac{2 \cdot \mathrm{Pr} \cdot \mathrm{Rc}}{\mathrm{Pr} + \mathrm{Rc}} \tag{4}$$

where R_H is a rule set consistent with a human-created alignment and R a rule set consistent with an automatically generated word alignment. The original

[8] The same approach has been proposed by [1] to show that the syntactic HMM word alignment models implemented in the BerkeleyAligner allow to create better and more general tree transducer rules. An evaluation of MT quality was not given.

Table 4. Comparison of CTRS for input alignments

(a) en-fr

Align	Pr	Rc	CTRS	Rules
G_I	35.49	33.94	34.70	35,115
G_R	38.32	46.89	42.17	44,977
G_G	43.99	44.20	44.09	36,972
G_U	45.39	42.43	43.86	34,403
BA	41.16	50.82	45.49	45,364
ACME	44.02	55.45	**49.08**	46,285

(b) pl-fr

Align	Pr	Rc	CTRS	Rules
G_I	31.09	27.00	28.90	22,291
G_R	34.80	32.62	33.67	24,017
G_G	39.97	28.98	33.60	18,708
G_U	41.57	27.45	33.07	17,065
BA	37.45	36.21	36.82	24,793
ACME	60.99	50.44	**55.22**	21,245

CPER is calculated as $1 - $ F-score, for CTRS we find F-score more appropriate since an increase in F-score can be directly interpreted as an increase in the quality of a rule set.

According to [4], CPER penalizes incorrect or missing alignment links more severely than AER. When AER is computed, one incorrect alignment link reduces the number of correct alignments by one, which results in slight decreases in precision and recall, while missing alignment links result in a small decrease in recall only. For CPER, incorrect or missing links may result in the elimination or addition of more than one phrase pair and thus have a stronger influence on precision and recall. This is even truer of CTRS and Syntactic SMT, where many translation rules can be created from one phrase pair.

Table 4 depicts the CTRS results of the rule sets generated from the input alignments. The absolute number of rules generated for the test set is also given. For the GIZA++ derived rule sets, a reverse trend can be seen when CTRS is compared to AER: the rule sets based on recall-oriented alignments yielded a higher precision, while the rule sets created from alignments with a higher precision had higher values for CTRS recall. The observed changes in balance between precision and recall for the majority of the data sets can be explained by the way phrases and translation rules are built from alignments. Recall-oriented alignments generally result in a smaller number of phrases since the presence of more alignment points forces the creation of longer phrase pairs. In syntax-based MT, however, these phrases must be consistent with the provided parse trees, otherwise no rules are created.

5 Experiments and Evaluation

5.1 Filter Parameters

Choice of Input Rule Sets. During training, the rule sets generated from all discussed input alignments were used and the filter achieved a CTRS of 56.16%. Removing any single rule set from the training set resulted in drops in CTRS, e.g. the filter's performance dropped to 52.54% CTRS if the ACME rule set was removed. The distance of ca. 7% to the remaining best performing single rule

Table 5. Probabilities for some example rules

Translation Rule	Probability
PP(of NP[0]) → de [0]	0.7920
PP(of the NNP[0] NNP[1]) → de la [1] [0]	0.7060
PP(IN[0] the Black Rod) → [0] la verge noire	0.7729
PP(IN[0] the Black Rod) → [0] la verge noire apporte	0.1642

set BA (45.49%) persisted. Removing only the BA rule set showed no significant impact on CTRS (56.00%). However, if both ACME and BA were removed, CTRS decreased to 49.04%. Similar results were obtained if other rule sets were removed from the training data.

A second matter of interest concerns the filter performance when only single rule sets are given as input data. Using only G_U for training yielded a CTRS of 45.75% compared to 43.86% for the unfiltered version. Repeating the experiment for ACME alone resulted in 53.23% CTRS compared to 49.08% for the input rule set. The application of the filter to a single rule set changes the balance between precision and recall. For the ACME rule set precision now amounts to 66.84% and recall to 44.22% compared to a precision of 44.02% and a recall of 55.45% for the original rule set. The number of rules was reduced by roughly 50%.

Feature Selection and Partitioning. For the English-French language pair a CTRS of 56.16% was reached when all of features described in 3.2 were used. Removing single features resulted in only small changes, while the impact of removing groups of related features was more significant. The greatest impact was observed if all alignment-related features (**Align**$_m(i, j)$**, SrcAligned, SrcUnaligned, TrgAligned, TrgUnaligned**) were discarded, CTRS dropped then to 54.34%.

For their maximum entropy based word-alignment combination method[9], [4] observed that a partition into distinct models based on the values of selected features may result in improvements. We tested this approach for different rule features and feature combinations and found that a partition based on the following features works best: **SrcSymLen** (56.60%), **K** (56.54%), and a combination of both features (56.65%).

Balancing Precision and Recall via Manually Set Thresholds. Let r be a translation rule and x_r the context or feature set representing r. Then we define $\mathrm{F}(t)$ as the rule set generated by the filter at a threshold t as

$$\mathrm{F}(t) = \{\, r : p(\text{“good”}|x_r) \geq t \,\} \tag{5}$$

where $p(c|x)$ is the probability distribution defined in (1). Table 5 contains the probabilities for the example rules from Fig. 1. The last rule, created due to an incorrect alignment link, would be discarded for an appropriate threshold.

[9] The alignment method ACME is an implementation of this method.

Table 6. Comparison of rule quality according to the test set

(a) en-fr

Filter	Pr	Rc	CTRS	Rules
F(0.2)	54.87	55.67	55.27	36,745
F(0.3)	63.48	50.94	**56.52**	29,068
F(0.4)	70.46	45.34	55.17	23,306
F(0.5)	73.92	40.28	52.14	19,735
F(0.6)	77.28	33.59	46.82	15,743
F(0.7)	82.58	24.47	37.75	10,732

(b) pl-fr

Filter	Pr	Rc	CTRS	Rules
F(0.2)	57.59	59.22	58.39	24,967
F(0.3)	66.79	54.24	**59.86**	19,716
F(0.4)	74.21	49.39	59.31	16,160
F(0.5)	78.32	45.04	57.19	13,960
F(0.6)	80.61	39.56	53.08	11,915
F(0.7)	83.50	30.50	44.69	8,869

For the machine translation task, we decided to chose six thresholds, from 0.2 to 0.7 with a step of 0.1, and present the CTRS for both test sets in Tab. 6. All results were obtained using 5-fold cross-validation for the respective test sets. As defined in (5), the symbol F(t) denotes the rule set generated by the filter at a threshold t. The extreme values for precision and recall differ between the rule sets by roughly 30%.

5.2 MT Evaluation

In this section, we will give the machine translation results for all introduced rule sets — alignment-based and filtered. Translation quality is measured with lowercased BLEU-4 and NIST. All rule sets have been generated from the first 100,000 sentences of the two previously described training corpora. This size limit is purely technical since we have to deal with 17 distinct rule sets for each language pair. Machine translation test sets for both language pairs comprise the last 1,500 sentences from the respective corpora, while the development sets (1,000 sentences each) have been taken from the middle of the same corpora. Translation model weights of the decoder for each rule set have been optimised on the development set with Z-MERT [20].

The machine translation results are described in Tab. 7 (English-French) and Tab. 8 (Polish-French). For the English-French language pair, G_U performed best among the unfiltered rule sets and G_G reached the second best scores for all metrics. Rather surprising are the weak MT results for ACME, BA and G_R since the underlying alignments of these rule sets scored best in terms of AER and the first two rule sets showed the best CTRS results. A very similar situation can be observed for the unfiltered Polish-French rule sets.

For the English-French filtered rule sets, F(0.4) showed the best BLEU score and F(0.5) the best scores for NIST among all evaluated rule sets. The rule sets F(0.2) to F(0.5) outperformed the best unfiltered rule set for all three metrics, F(0.6) had better results for NIST. The number of unique rules in each rule set is also given. F(0.4) consists of roughly 75% fewer rules than ACME and 60% fewer than G_U. Negative effects of data sparseness seem to manifest somewhere between a filter threshold of 0.6 and 0.7. Results for the Polish-French language pair are similar though less significant.

Table 7. MT scores for English-French language pair

(a) Input rule sets

Align	BLEU	NIST	Rules
G_I	0.1918	5.2983	5,023,457
G_R	0.1738	4.9788	6,148,095
G_G	0.2006	5.3016	4,576,837
G_U	**0.2049**	**5.3678**	4,182,497
BA	0.1923	5.2168	6,037,889
ACME	0.1891	5.1620	6,269,929

(b) Filtered rule sets

Filter	BLEU	NIST	Rules
F(0.2)	0.2079	5.4456	3,533,210
F(0.3)	0.2093	5.5091	2,253,812
F(0.4)	**0.2127**	5.7492	1,584,581
F(0.5)	0.2090	**5.8208**	1,243,441
F(0.6)	0.2031	5.7821	926,183
F(0.7)	0.1570	4.7648	605,946

Table 8. MT scores for Polish-French language pair

(a) Input rule sets

Align	BLEU	NIST	Rules
G_I	0.2955	6.1520	4,327,075
G_R	0.3031	6.1183	4,572,431
G_G	0.3200	6.4454	3,136,140
G_U	**0.3218**	**6.5050**	2,768,452
BA	0.3060	6.2897	4,562,305
ACME	0.2989	6.1283	3,625,969

(b) Filtered Rule Sets

Filter	BLEU	NIST	Rules
F(0.2)	0.3138	6.3625	3,855,027
F(0.3)	0.3144	6.4079	2,624,104
F(0.4)	0.3246	6.5865	1,894,711
F(0.5)	**0.3301**	**6.7269**	1,505,139
F(0.6)	0.3168	6.6654	1,161,098
F(0.7)	0.2656	5.6743	744,163

It is worth mentioning that the best MT results were reached for both language pairs at thresholds close to 0.5. In terms of the used maximum entropy model, this means that we could revert from thresholding strategies and return to hard classification as defined in (2). If this could be shown to be a generally valid result, it would confirm that the rules classified as "good" — and therefore more similar to those generated from a manually-created word alignment — are indeed well suited for Syntactic SMT. Since thresholds were chosen arbitrarily, we cannot say whether a threshold exists that would yield better MT quality. Hence, one direction for further research should include threshold optimization in terms of BLEU scores on a given development set.

6 Discussion

Previous work [4] has shown that improved results for AER and CPER (or CTRS in this work) are not good indicators for Phrase-Based SMT quality. In the context of Syntactic SMT, these findings can be confirmed for the alignments generated by the BerkeleyAligner (BA) and especially the supervised alignment method ACME. The MT results for both rule sets are significantly worse than for G_U and G_G although they show superior AER and CTRS scores. Similarly, the filtered rule sets with the highest CTRS do not reach the best MT scores, but are exceeded by filters with higher thresholds. However, the differences in CTRS between these rule sets are rather small. All rule sets that reached high

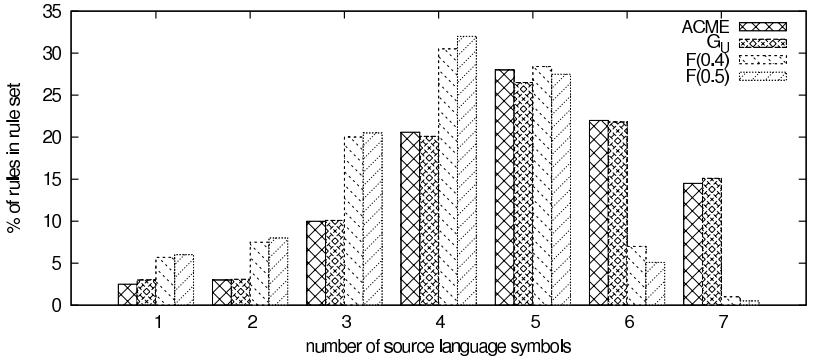


Fig. 2. Histogram of rule lengths for chosen unfiltered and filtered rule sets

MT results maintain a relatively high CTRS and prefer CTRS precision over CTRS recall. This is equally true for the unfiltered and filtered rule sets. High CTRS precision is generally connected with high AER recall.

For the alignment-based rule sets, the worst performing sets have the greatest number of rules and vice versa. The same is true for the filtered rule sets if we disregard F(0.7). The chart in Fig. 2 allows us to compare the distribution of rule lengths (the number of source language symbols) for four chosen rule sets: ACME, G_U, F(0.4), and F(0.5). There are no significant differences between the two unfiltered rule sets or between both filtered rule sets. However, when comparing the filtered rule sets to the unfiltered ones, we can see that the majority of rules longer than 5 symbols has been discarded. The decrease in number of long rules is the main factor behind the size reduction of the filtered sets. Since long rules will only be used in specific construction, it is possible that the final effect of the filtration is in some degree equivalent to the effects of significance testing described by [15] for Phrase-Based SMT, which might be an explanation for the better MT results obtained by the filtered rule sets.

We have shown that a maximum entropy model trained on a reference rule set generated from manual alignments can improve machine translation quality and reduce the number of translation rules at the same time. This simple approach could improve CTRS several percent over the best unfiltered rule set even if only one rule set is used. The findings of other researchers that AER is not necessarily related to MT quality have been confirmed; for CTRS, however, a relationship between better MT results and higher CTRS precision seems to exist. From this, it follows that alignment combination methods that aim for recall seem to be better suited for Syntactic SMT than precision-oriented methods, a result that contradicts those presented by [4] for Phrase-Based SMT.

Acknowledgements

This paper is based on research funded by the Polish Ministry of Science and Higher Education (Grant No. 003/R/T00/2008/05).

References

1. DeNero, J., Klein, D.: Tailoring word alignments to syntactic machine translation. In: Proceedings of ACL, pp. 17–24 (2007)
2. Fossum, V., Knight, K., Abney, S.: Using syntax to improve word alignment precision for syntax-based machine translation. In: Proceedings of ACL Workshop on Statistical Machine Translation, pp. 44–52 (2008)
3. Zollmann, A., Venugopal, A., Och, F., Ponte, J.: A systematic comparison of phrase-based, hierarchical and syntax-augmented statistical MT. In: Proceedings of ACL-COLING, pp. 1145–1152 (2008)
4. Ayan, N.F., Dorr, B.J.: Going beyond AER: an extensive analysis of word alignments and their impact on MT. In: Proceedings of ACL-COLING, pp. 9–16 (2006)
5. Junczys-Dowmunt, M.: It's all about the trees — towards a hybrid syntax-based MT system. In: Proceedings of IMCSIT, pp. 219–226 (2009)
6. Huang, L.: Statistical syntax-directed translation with extended domain of locality. In: Proceedings of AMTA, pp. 66–73 (2006)
7. Liu, Y., Liu, Q., Lin, S.: Tree-to-string alignment template for statistical machine translation. In: Proceedings of ACL, pp. 609–616 (2006)
8. Aho, A.V., Ullman, J.D.: Translations on a context-free grammar. Information and Control 19, 439–475 (1971)
9. Graehl, J., Knight, K.: Training tree transducers. In: Proceedings of HLT-NAACL, pp. 105–112 (2004)
10. Galley, M., Hopkins, M., Knight, K., Marcu, D.: What's in a translation rule. In: Proceedings of HLT-NAACL, pp. 273–280 (2004)
11. Germann, U.: Aligned hansards of the 36th parliament of Canada (2001)
12. Och, F.J., Ney, H.: A systematic comparison of various statistical alignment models. Computational Linguistics 29, 19–51 (2003)
13. Berger, A.L., Della Pietra, V.J., Della Pietra, S.A.: A maximum entropy approach to natural language processing. Computational Linguistics 22, 39–71 (1996)
14. Darroch, J., Ratcliff, D.: Generalized iterative scaling for log-linear models. The Annals of Mathematical Statistics 43, 1470–1480 (1972)
15. Johnson, H., Martin, J., Foster, G., Kuhn, R.: Improving translation quality by discarding most of the phrasetable. In: Proceedings of EMNLP-CoNLL, pp. 967–975 (2007)
16. Mihalcea, R., Pedersen, T.: An evaluation exercise for word alignment. In: Proceedings of HLT-NAACL, pp. 1–10 (2003)
17. Koehn, P., Och, F., Marcu, D.: Statistical phrase-based translation. In: Proceedings of HLT-NAACL, pp. 48–54 (2003)
18. Ayan, N.F., Dorr, B.J.: A maximum entropy approach to combining word alignments. In: Proceedings of HLT-NAACL, pp. 96–103 (2006)
19. Klein, D., Manning, C.D.: Accurate unlexicalized parsing. In: Proceedings of ACL, pp. 423–430 (2003)
20. Zaidan, O.F.: Z-MERT: A fully configurable open source tool for minimum error rate training of machine translation systems. The Prague Bulletin of Mathematical Linguistics 91, 79–88 (2009)

Mining Parenthetical Translations for Polish-English Lexica

Filip Graliński

Adam Mickiewicz University
Faculty of Mathematics and Computer Science
ul. Umultowska 87
61-614 Poznań, Poland
filipg@amu.edu.pl

Abstract. Documents written in languages other than English sometimes include parenthetical English translations, usually for technical and scientific terminology. Techniques had been developed for extracting such translations (as well as transliterations) from large Chinese text corpora. This paper presents methods for mining parenthetical translation in Polish texts. The main difference between translation mining in Chinese and Polish is that the latter is based on the Latin alphabet and it is more difficult to identify English translations in Polish texts. On the other hand, some parenthetically translated terms are preceded with the abbreviation "ang." (=English), a kind of an "anchor", allowing for querying a Web search engine for such translations.

1 Introduction

Bilingual lexica are of paramount importance because of their applications in such natural language processing domains as (both statistical and rule-based) machine translation, computer-assisted translation or cross-language information retrieval. With the rapid growth of the Internet, a natural question arises: how to extract bilingual lexicon entries from the huge volume of Web data, not only Web pages, but also PDF documents or files in Microsoft Word format.

One line of research is to collect bilingual sentence-level Web corpora, e.g. by exploiting pairs of Web pages that are mutual translations [1], and to automatically acquire lexical data from them [2]. Sometimes comparable rather than strictly parallel corpora (e.g. Wikipedia) are used [3].

Interestingly, some bilingual lexical data can be extracted from (purely or mostly) monolingual corpora. One method is to combine frequency information and cognate analysis [4]. Another technique exploits short bilingual snippets repeated in a similar manner in a mostly monolingual Web page [5]. Finally, some bilingual lexicon entries can be extracted from semi-structured Web data sources such as bilingual keyword listings [6].

In this paper, experiments on mining bilingual data from *parenthetical translations* put in (mostly) monolingual Polish Web texts are reported. The idea comes from the observation that Polish authors sometimes annotate words, terms, book

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 464–472, 2010.

or film titles with their translations in English. The following example illustrates this phenomenon:

> Stosować się będzie bowiem ona do działalności nie tylko operatora i dysponenta sieci telekomunikacyjnej (ang. network providers) oraz dostawcy dostępu do Internetu (ang. access providers), ale również dostawców usług w sieciach (ang. Internet Service Providers).[1]
>
> [For it will be applicable to the operations of not only an operator and owner of the telecommunications network (Eng. network providers) as well as of a provider of the access to the Internet (Eng. access providers), but also of providers of services in networks (Eng. Internet Service Providers).]

(A literal translation of the sentence is given in square brackets, the terms for which parenthetical translations were specified in the original texts are underlined and translated word-by-word here.) The three parenthetical translations were given in round brackets and were preceded by the word *ang.*, which is an abbreviated form of the adjective *angielski* (= *English*).

Even though parenthetical translations are typical for academic papers, PhD and master's theses and other types of formal texts, they can be occasionally encountered in virtually any kind of Web texts. Their frequency is rather low but the sheer size of the Web makes their number large (even for medium-sized languages such as Polish). They are valuable needles which come in thousands in the huge haystack of the Web. What makes them interesting is the very reason they are used: they are new and/or technical terms usually with no standard Polish translation, absent from conventional dictionaries.

The idea to mine parenthetical translation is not new: techniques for supervised [7], semi-supervised [8] and unsupervised [9] lexicon mining were proposed for English parenthetical translations in Chinese texts. No experiments, however, have been reported for languages other than Chinese, and in particular for languages the writing system of which is based on the Latin alphabet. It should be noted that the case of Polish is different, to some extent, from the Chinese language as far as English parenthetical expressions are concerned. First, as the same Latin alphabet is used in Polish and English,[2] it would be more difficult (for computers as well as for humans) to identify English insertions if only parentheses were to be used, therefore some additional clues (like the abbreviation *ang.*) are usually applied. Second, English parenthetical expressions rarely refer to Polish transliterations.[3] Third, the volume of English parenthetical transliterations seems to be much smaller in Polish than in Chinese, which makes some of the quantitative methods unfeasible.

[1] http://www.piit.org.pl/piit2/index.jsp?place=Lead07&news_cat_id=51&news_id=1422&layout=2&page=text

[2] Except that 9 characters with diacritics (*ą*, *ć*, *ę*, *ł*, *ń*, *ó*, *ś*, *ź*, *ż*) are used in Polish.

[3] With some minor exceptions, like Russian names, which are traditionally transliterated in Polish in a different way than in English, e.g. *Maja Plisiecka (ang. Maya Plisetskaya)*.

The paper is organised as follows: Sect. 2 is a discussion of the conventions for parenthetical transliterations used by Polish authors. Section 3 presents methods for gathering Web corpora and Sect. 4 – the methods for extracting parenthetical translations. The results come in Sect. 5 and remarks concerning future work and conclusions are provided in Sect. 6.

2 Parenthetical Translation Conventions

There are two main conventions for specifying parenthetical English translations in Polish texts:

A. $p_1\ p_2 \ldots p_m$ `(ang.` $e_1\ e_2 \ldots e_n$ `)` – the English translation is given in parentheses and is preceded by the abbreviation *ang.* (= *English*), see the example given in the Introduction;
B. $p_1\ p_2 \ldots p_m$ ($e_1\ e_2 \ldots e_n$) – the English translation is given in parentheses, in italics.

(Here, p_1, p_2,..., p_m denote Polish words, whereas e_1, e_2,..., e_n – English words.) Some variations can, however, be observed. Some of them are:

- $p_1\ p_2 \ldots p_m$ `(ang.` $e_1\ e_2 \ldots e_n$ `)` – (A) and (B) combined,
- $p_1\ p_2 \ldots p_m$ `(z ang.` $e_1\ e_2 \ldots e_n$ `)` – *z ang.* = *from English*,
- „$p_1\ p_2 \ldots p_m$” `(ang.` “$e_1\ e_2 \ldots e_n$” `)` – additional quotes are used,
- $p_1\ p_2 \ldots p_m$ `(ang.:` $e_1\ e_2 \ldots e_n$ `)` – colon used after the abbreviation *ang.*,
- $p_1\ p_2 \ldots p_m$ `[ang.` $e_1\ e_2 \ldots e_n$ `]` – non-round brackets are used.

Sometimes Polish or English synonyms or glosses are given within the parenthesis, e.g.: *Rozwój zrównoważony (ekorozwój, ang. sustainable development)*[4] or *uwalnianie leku z jego postaci farmaceutycznej*[5] *(ang. liberation, drug release)*. Acronyms are described in parenthetical expressions even more often, e.g.: *rdzeniowej atrofii mięśni (ang. spinal muscular atrophy, SMA)*. Quite frequently, instead of a Polish term, only the acronym is given and the parenthetical expressions is just the English term for which it stands, see the following example:

> W metodach wektorowych wykorzystuje się między innymi algorytmy FDTD (ang. Finite Difference Time Domain) i FMM (ang. Fourier Modal Method).[6]
>
> [In vector methods FDTD (Eng. Finite Difference Time Domain) and FMM (ang. Fourier Modal Method) algorithms are used among others.]

[4] *Rozwój zrównoważony* = lit. *stable development*, *ekorozwój* = lit. *eco-development*
[5] = lit. *release of the drug from its pharmaceutical form*
[6] `http://pl.wikipedia.org/wiki/Transformacja_genetyczna`

Such abbreviations along with their full forms might be of interest (e.g. for acronym lexicons), I decided, however, to filter them out (see Sect. 4.2) as, strictly speaking, they are not translations.

This paper focuses on convention (A) and its variations, i.e. only parenthetical expressions with the abbreviation *ang.* are considered. The reason is that visual formatting markup is usually discarded while generating text corpora from Web pages,[7] which makes recognising the convention (B) more difficult. Also, as we will see in the next section, the abbreviation *ang.* makes it possible to seek out texts with parenthetical translations on the Internet.

3 Corpora

3.1 Pre-existing Corpora

I started with the available Polish corpora (i.e. not collected with parenthetical translations in mind), namely: a general Web corpus of over 2.8 million web pages and PDF files collected from the Polish Internet, a dump of the Polish Wikipedia and a collection of Polish academic papers and abstracts (see Table 1). The frequency of the abbreviation *ang.* token turned out to be much higher in Wikipedia and scientific texts than in general Web texts. The total number of occurrences of *ang.* was 41,714. As this is the upper bound of the number of parenthetical translations with *ang.* (*ang.* can be used for other purposes than parenthetical translations, see the next subsection), the results were somewhat unsatisfactory. This is why the decision was made to actively seek parenthetical translations on the Internet.

Table 1. Corpora initially used (#*ang.* – number of *ang.* tokens in the text)

Corpus	**Bytes**	**Tokens**	**#*ang.* (/ 1M tokens)**
Web corpus	15.6GB	2.0G	24487 (12.2)
Wikipedia dump	498MB	61.5M	11300 (183.7)
Corpus of academic papers	425MB	56.2M	5927 (105.4)
Total	16.5GB	2.1G	41714 (19.7)

3.2 Dedicated Corpus

The interesting thing about the abbreviation *ang.* is that it can be used not only for the extraction of desired parenthetical expressions in a given document, but also for seeking out the document itself on the Internet, i.e. a query with *ang.* can be constructed to locate document with parenthetical translations using Web search engines.

One obstacle is that the periods (full stops) are usually discarded by search engines and *ang.* would be probably normalised to `ang`, the same goes for such

[7] The problem is even more complicated for PDF files.

words as *ang*, *Ang* and *ANG*. Fortunately, the tokens normalised to `ang` are not frequently used for purposes other than parenthetical translations, in particular *ang* is not a valid Polish word. Some of the cases which nevertheless should be taken into account are:

- *ang.* in *j. ang.* or *jęz. ang.*, a short form for *język angielski* (*the English language*),
- *ang.-pol.* (or *pol.-ang.*), a short form for *angielsko-polski* (*English-(to-)Polish*), which is usually tokenized and normalised by search engines into two strings: `ang` and `pol`,
- *Ang* as the first name of the film director Ang Lee.

In order to avoid on-line dictionaries and Web sites for Polish students of the English language (where *ang.* is often used for purposes other than parenthetical translations), three additional words were added as "negative" terms in the constructed query: *słownik* (*dictionary*), *język* (*language/tongue*), *angielski* (*English*). Hence, the final query was as follows:

```
ang -"j ang" -jęz -pol -lee -słownik -język -angielski
```

This query (and its variations) was entered into the Google and Bing search engines (with the language option set to Polish). The websites with the largest number of hits were additionally crawled by an in-house web robot. A list of 91,872 URLs was obtained in this manner. 69,493 files were successfully downloaded and converted[8] into plain text. The characteristics of the corpus are given in Table 2.

It should be noted that no dedicated corpora were gathered in the experiments concerning Chinese-English parenthetical translations ([7], [8], [9]).

Table 2. Dedicated corpus

Bytes	Tokens	#*ang.* (/ 1M tokens)
1.36GB	177M	141227 (798.9)

4 Translation Extraction

4.1 Preprocessing

Snippets containing *ang.* were first extracted using hand-crafted regular expressions. The limit for the number of tokens to the left and to the right of *ang.* was set to 7. Some words (mostly Polish conjunctions) were then "blacklisted" and discarded from the beginning of a snippet. Anomalous snippets, e.g. with no letters in pre-parenthesis nor in in-parenthesis fragments, were discarded as well.

If the fragment of a snippet suspected of being an English parenthetical translation contained characters with Polish diacritics, the snippet as a whole was discarded.

[8] Some PDFs could not be converted into text by the tool available (`pdftotext`).

4.2 Filtering Out Acronyms

As I mentioned in Sect. 2, sometimes there is no Polish equivalent to the English term in parenthesis – with only the English acronym provided. Such abbreviations were filtered out by comparing the letters of an acronym with the initial letters of the term words and taking into account some standard acronym conventions (such as using *2* instead of *to*). A random sample of ten filtered out snippets is listed in Table 3.

Table 3. A sample of snippets filtered out

Snippet
EURIBOR (ang. Euro Interbank Offered Rate)
ON (ang. over night)
odróżniającą go od większości MTA (ang. mail transport agent)
stopa depozytów jednodniowych rozpoczynających się dziś SW (ang. spot week)
jest protokołem wykorzystywanym do przeglądania WWW (ang. World Wide Web)
Systemy MES (ang. Manufacturing Execution System)
PIN (ang. Personal Identification Number)
SCORM i AICC. LMS (ang. Learning Management System)
LIBOR (ang. London Interbank Offered Rate)
Forex (ang. Foreign Exchange)

The number of candidate translation pairs after preprocessing and filtering out was 82,434 (all the corpora mentioned in Sect. 3 were used).

4.3 Word Alignment

In order to extract a parenthetical translation the *first* word of the Polish equivalent of the parenthetical translation ought to be determined. Following [9] I used a word alignment algorithm for determining the left boundary: the first pre-parenthesis word aligned with an in-parenthesis word is assumed to be the left boundary of the Polish equivalent of the English in-parenthesis translation. However, as the collection of snippets was much smaller than that obtained in [9] an external Polish-English lexicon had to be consulted. The lexicon contained 474,265 translation pairs (both single words and multi-word units), it was based on heterogeneous acquisition techniques and data sources. The translation pairs obtained from parenthetical expressions are planned to be yet another source of lexical data for this still growing lexicon.

Competitive Linking [10], a simple yet effective [11] algorithm, was used for word alignment. The algorithm is a kind of greedy, best-first search: a pair of words can be linked on condition that none of the two words were previously aligned to any other words. Potential word associations are sorted by some score.

As it was mentioned already, an external lexicon was used as the source of scores for pairs of Polish and English words. The scores had been calculated based on the number and the quality of sources confirming the given translation

pair. Contrary to [9], consecutive sequences of words are not allowed to be linked independently to one word on the other side, however, lexicon multi-word units are taken into account during linking, so many-to-many links are allowed for words being part of multi-word units.

For word pairs not listed in the external lexicon, cognate analysis was introduced as an additional source of scores:

1. The Polish and English words are normalised to abstract from most frequent differences in Polish and English spelling: *ks* → *x*, *ph* → *f*, *sz* → *sh*, *k* → *c*, *w* → *v*, *y* → *i*.
2. The longest common prefix for the Polish and English word (after normalisation) is determined. If it is longer than 4, the words can be aligned, the longer is the common prefix, the higher is the score.

5 Results

A sample of 600 snippets with the abbreviation *ang.* was randomly selected for evaluation.[9] The sample was manually inspected and 333 (55,5%) correct translation pairs were identified and marked up. The automatic translation

Table 4. Results for the sample (E – number of extracted translations, C – number of correct translations)

Method	E	C	Prec.	Recall	F-score
baseline	368	169	0.459	0.508	0.482
cognates	84	40	0.476	0.120	0.192
lexicon	204	147	0.721	0.441	0.547
lexicon + cognates	216	154	0.713	0.462	0.561
lexicon + cognates + one-word backup	318	175	0.550	0.526	0.538
“fair” lexicon + cognates	168	114	0.679	0.342	0.455
“fair” lexicon + cognates + one-word backup	315	170	0.540	0.511	0.525

Table 5. A sample of extracted translations. Extracted translations are underlined

Correct?	Snippet
yes	serwery domeny głównej (ang. root servers)
yes	będące integracją infrastruktury hurtowni danych (ang. Data Warehouse)
yes	szafa stelażowa (ang.: rack)
yes	Marynarka Wojenna Stanów Zjednoczonych (ang. United States Navy,
too long	Przerwa ta (ang. Intermission)
too long	Nicolas Dauphas z Uniwersytetu w Chicago (ang. University of Chicago)
too short	może oznaczać wystrój "bojowy" (ang. war color)
too short	posiadanie szczególnych przymiotów moralnych (ang. moral insight

[9] The sample was not used during the development.

extraction procedure described in Sect. 4 was then applied to the sample. The results are presented in Table 4. The **baseline** is simply taking the same number of Polish words as on the English side. **One-word backup** is used when no lexicon/cognate alignments were found: if the parenthetical expression is just one English word, take the last pre-parenthesis word as its Polish translation.

It should be noted that if the external lexicon is used for alignment (**lexicon** method) and if a single link for the whole parenthetical English expression (one word or a multi-word unit) can be found in the lexicon then the correct translation pair (i.e. attested in the lexicon) will be extracted. Translation extraction could be viewed more as confirmation rather than as discovery in such a case. Therefore, the **"fair" lexicon** method was introduced for comparison. "Fair lexicon" means that links for the whole parenthetical expression are not used during alignment.

Finally, the translation extraction procedure was applied to the full corpus of 82,434 snippets. 46,728 unique translation pairs were extracted using the lexicon+cognates method. A sample of extracted translations is listed in Table 5.

6 Conclusions and Future Work

The number of extracted parenthetical translations reported here is much smaller than obtained for Chinese texts ([8], [9]), even if to take into account that the Polish corpus was smaller. The main reason is that the frequency of parenthetical English translation in Polish is simply much lower than in Chinese. There is nevertheless some room for improvement: part-of-speech could be taken into account, machine learning techniques could be used for filtering out incorrect translation pairs, parenthetical translations without the abbreviation *ang.*[10] could be identified (e.g. using methods with which semantics relations are extracted [12]).

Even though the results presented in this paper are less encouraging than those reported for Chinese, the parenthetical expressions can be used as a supplementary source of Polish-English lexical data (for other examples of such sources see [6]).

The methods proposed in this paper could probably be adopted for other European languages provided that the frequency of the expression analogical to *ang.* is high enough.

Acknowledgements

The paper is based on research funded by the Polish Ministry of Science and Higher Education (Grant No 003/R/T00/2008/05).

[10] It should be noted, however, that parenthetical expressions with *ang.* constitute a substantial part (if not the majority) of all the parenthetical translation.

References

1. Resnik, P., Smith, N.A.: The web as a parallel corpus. Comput. Linguist. 29(3), 349–380 (2003)
2. Melamed, I.D.: Automatic discovery of non-compositional compounds in parallel data. In: Proceedings of the 2nd Conference on Empirical Methods in Natural Language Processing (1997)
3. Shao, L., Ng, H.T.: Mining new word translations from comparable corpora. In: COLING 2004: Proceedings of the 20th international conference on Computational Linguistics, Morristown, NJ, USA. Association for Computational Linguistics, p. 618 (2004)
4. Haghighi, A., Liang, P., Berg-Kirkpatrick, T., Klein, D.: Learning bilingual lexicons from monolingual corpora. In: Proceedings of ACL-08: HLT, Columbus, Ohio. Association for Computational Linguistics, pp. 771–779 (2008)
5. Jiang, L., Yang, S., Zhou, M., Liu, X., Zhu, Q.: Mining bilingual data from the web with adaptively learnt patterns. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, Suntec, Singapore. Association for Computational Linguistics, pp. 870–878 (2009)
6. Graliński, F., Jassem, K., Kurc, R.: Acquiring bilingual lexica from keyword listings. In: Vetulani, Z. (ed.) Proceedings of 4th Language & Technology Conference, Poznań, Wydawnictwo Poznańskie Sp. z o.o, pp. 326–330 (2009)
7. Cao, G., Gao, J., Nie, J.Y.: A system to mine large-scale bilingual dictionaries from monolingual web pages. In: MT Summit XI, pp. 57–64 (2007)
8. Wu, X., Okazaki, N., Tsujii, J.: Semi-supervised lexicon mining from parenthetical expressions in monolingual web pages. In: NAACL 2009: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Morristown, NJ, USA. Association for Computational Linguistics, pp. 424–432 (2009)
9. Lin, D., Zhao, S., Van Durme, B., Paşca, M.: Mining parenthetical translations from the web by word alignment. In: Proceedings of ACL 2008: HLT, Columbus, Ohio. Association for Computational Linguistics, pp. 994–1002 (2008)
10. Melamed, I.D.: Models of translational equivalence among words. Comput. Linguist. 26, 221–249 (2000)
11. Tiedemann, J.: Word to word alignment strategies. In: COLING 2004: Proceedings of the 20th international conference on Computational Linguistics, Morristown, NJ, USA. Association for Computational Linguistics, p. 212 (2004)
12. Pantel, P., Pennacchiotti, M.: Espresso: leveraging generic patterns for automatically harvesting semantic relations. In: ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, Morristown, NJ, USA. Association for Computational Linguistics, pp. 113–120 (2006)

Automatic Generation of Bilingual Dictionaries Using Intermediary Languages and Comparable Corpora

Pablo Gamallo Otero[1] and José Ramom Pichel Campos[2]

[1] Departamento de Língua Espanhola,
Universidade de Santiago de Compostela, Galiza, Spain
pablo.gamallo@usc.es
[2] Departamento de Tecnologia Linguística da Imaxin|Software
Santiago de Compostela, Galiza
jramompichel@imaxin.com

Abstract. This paper outlines a strategy to build new bilingual dictionaries from existing resources. The method is based on two main tasks: first, a new set of bilingual correspondences is generated from two available bilingual dictionaries. Second, the generated correspondences are validated by making use of a bilingual lexicon automatically extracted from non-parallel, and comparable corpora. The quality of the entries of the derived dictionary is very high, similar to that of hand-crafted dictionaries. We report a case study where a new, non noisy, English-Galician dictionary with about 12,000 correct bilingual correspondences was automatically generated.

1 Introduction

In this paper we describe a method to derive a new bilingual lexicon from two existing ones using comparable corpora to validate candidate correspondences. The method is entirely unsupervised and consists of two tasks. First, given two existing bilingual lexicons for two languages pairs (A, B) and (B, C), we can obtain a new pair (A, C) by simple transitivity. Second, the generated bilingual correspondences are validated using translation equivalents automatically extracted from comparable corpora. In particular, we will derive a new $(English, Galician)$ lexicon from two existing dictionaries, $(English, Spanish)$ and $(Spanish, Galician)$, by making use of English-Galician comparable corpora.

The strategy described in the paper is especially well suited to create new language resources for minority languages (e.g., Galician) from languages such as English or Spanish, which have a lot more resources. Our method does not require the minority language being provided with many and large linguistic resources: only a bilingual dictionary and some raw text is required. This is enough to automatically build a new non-noisy, bilingual lexicon.

This strategy is also useful to create new bilingual dictionaries for multilingual machine translation systems, such as Opentrad-Apertium[1]. The number of

[1] http://www.opentrad.com/

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 473–483, 2010.

bilingual dictionaries required by a multilingual translator increases as a quadratic function of the number of languages the system aims to translate [15]. So, the process of automatically deriving new bilingual resources can drastically reduce the amount of work.

The paper is organized as follows: the following section (2) introduces some related work. Then, Section 3 describes the different steps of our method. Next, in Section 4, we report a case study where a new, non-noisy, English-Galician dictionary with about 12,000 bilingual correspondences was automatically generated. Finally, some conclusions are put forward in Section 5.

2 Related Work

There exist some approaches to derive bilingual lexicons from existing ones [11,1,16,10,15]. Our work is directly inspired by [10], who sketch a very similar methodology to that proposed here. They use two bilingual lexicons sharing the same language (the pivot) and derive a new bilingual dictionary by using the pivot language as intermediate. The new lexicon is derived by transitivity. For instance, given the language pairs $(English, Spanish)$ and $(Galician, Spanish)$, as Spanish as language pivot, their method build a new bilingual pair without the pivot language: $(English, Galician)$. The crucial aspect of this strategy is the validation of correspondences. The validity of the retained correspondences was checked using a parallel corpus, i.e., only the correspondences found in the parallel corpus are kept.

The specificity of our method is the fact that we used comparable corpora, instead of parallel texts, to validate the correspondences retained by transitivity. So, our main contribution is to propose a strategy to validate new bilingual lexicons by making use of translation equivalents extracted from non-parallel, comparable corpora. This kind of corpus is easier available than parallel texts, especially for minority languages.

Unlike most approaches to extract word translations from non-parallel corpora [6,7,12,4,14,13], which are based on baseline windowing techniques, our method relies on syntactically analyzed text. In [9], it is showed that the use of syntactic dependencies instead of window-based strategies significantly improves the accuracy of the extraction.

3 The Method

Our strategy consists of two main tasks: both to generate candidate bilingual correspondences by transitivity and to validate them by using translation equivalents extracted from comparable corpora.

3.1 Generation by Transitivity

The first task is inspired by that described in [10]. Given two bilingual dictionaries represented as two relations (A, B) and (B, C), we generate a derived dictionary (A, C) as follows:

- First, we create the relation (A, C') taking two existing dictionaries (A, B) and (B, C), where B is the pivot language. For each bilingual correspondence (a_i, b_i) belonging to the relation (A, B), we create a set of new correspondences $\{(a_i, c_1), (a_i, c_2), \ldots, (a_i, c_n)\}$, where $c_1, \ldots, c_n$ are those words and terms associated with b_i within (B, C). The derived dictionary (A, C') is the set of all new bilingual correspondences.
- Then, we remove the redundant bilingual pairs from (A, C'). The result is the relation (A, C).
- Finally, we split (A, C) into two complementary subsets: $(A, C)_{amb}$, which consists of those correspondences containing at least one ambiguous word, and $(A, C)_{unamb}$, containing only unambiguous words. Note that the former is a many-to-many relationship whereas the latter is one-to-one.

As in [10], the derived dictionary with only unambigous words, $(A, C)_{unamb}$, can be considered as a non-noisy lexical resource. In Lexicography, words with only one translation equivalent behave as not ambiguous terms. Therefore, all the unique correspondences derived from unambigous words (one-to-one) are of good quality and must be validated. By contrast, $(A, C)_{amb}$ is a noisy lexicon. The translation by transitivity of ambiguous words can overgenerate odd bilingual correspondences. For instance, in one of our $(English, Spanish)$ dictionaries, the verb *subside* is translated in Spanish as *bajar*, which is translated, in turn, by the $(Spanish, Galician)$ dictionary as *baixar* and *apear*. Therefore, the derived $(English, Galician)$ dictionary must contain the correspondences $(subside, baixar)$ and $(subside, apear)$. While the former translation is correct, the latter is clearly odd. The galician verb *apear* does not mean *subside* in any context; it means *take down*, which is one of the senses of the spanish word *bajar*.

In the next task, all correspondences of $(A, C)_{amb}$ will be checked using translation equivalents between language A and C extracted from comparable corpora.

3.2 Validation with Comparable Corpora

The second process is the main contribution of our work. It consists in filtering out those ambiguous correspondences that are not in a lexicon of translation equivalents automatically generated from a non-parallel corpus syntactically annotated with dependencies. The lexicon of translation equivalents, called $(A, C)_{corpus}$ is organized as follows. Each term of language A, a_i, is assigned a ranked list of terms of language C, $c_1, c_2, \ldots, c_n$, which are the top-N best translation candidates of a_i. Conversely, each term of language C, c_i, is assigned a ranked list of terms of language A, $a_1, a_2, \ldots, a_n$, which are the top-N best translation candidates of c_i. So, the relation $(A, C)_{corpus}$ consists of correspondences between words and their candidate translations inferred from the corpus. To validate $(A, C)_{amb}$, we make the intersection between $(A, C)_{amb}$ and $(A, C)_{corpus}$. The resulting relation is a set of correct bilingual correspondences containing ambiguous words. Finally the new non-noisy, derived lexicon,$(A, C)_{not-noisy}$, is the union of this validated relation with the bilingual lexicon of unambiguous words:

$$(A, C)_{not-noisy} = (A, C)_{amb} \cap (A, C)_{corpus} \cup (A, C)_{unamb}$$

In the following subsection, it is described how $(A, C)_{corpus}$ is learned.

3.3 An Approach to Extract Translation Equivalents from Comparable Corpora

Our method to extract translation equivalents from syntactically annotated comparable corpora was described in detail in previous work [9,8]. Here, we only sketch the main properties of the approach. The starting point is the following: word w_1 is a candidate translation of w_2 if the lexical-syntactic contexts in which w_1 occurs are translations of the lexical-syntactic contexts in which w_2 occurs. Words (or multiword terms) are previously lemmatized. This strategy relies on a list of bilingual lexical-syntactic contexts (called *seed contexts*) provided by an external bilingual dictionary, (A, C), and a list of generic syntactic dependencies: subject, direct object, adjective modification, prepositional complement, etc. So, w_1 is a candidate translation of w_2 if they tend to co-occur with the same seed contexts. For instance, let's suppose that the dictionary (A, C) contains the correspondence *(subside, baixar)*. As they are two specific verbs, we can build a bilingual correspondence between two lexical-syntactic contexts introduced by their corresponding verbs:

$$(< Subject; subside, NOUN >, < Subject; baixar, NOUN >)$$

where $< Subject; subside, NOUN >$ is used to identify those English nouns appearing in the subject position of *subside*, while $< Subject; baixar, NOUN >$ allows to select those Galician nouns playing the role of subject of *baixar*. This bilingual correspondence is used as a "seed context" in the process of selecting translation equivalents. This way, if English nouns such as *fever* or *swelling* appear as subject of *subside*, the Galician nouns occurring in the subject position of *baixar* (e.g., *febre* or *inchazón*) are candidate to be their translations.

The extraction method consists of the following subtasks[2]:

Multilingual parsing. The two corpora are analyzed using a multilingual dependency based parser, DepPattern[3].

Seed contexts. A list of seed lexical-syntactic contexts is created from the noisy bilingual dictionary, (A, C), and a small set of generic syntactic rules. Note that the bilingual dictionary used as source is that derived by transitivity in the previous task. It contains both ambiguous and unambiguous correspondences, even if the former ones can contain several errors.

[2] This method was implemented in a prototype system available at `http://gramatica.usc.es/~gamallo/prototypes/BilingualExtraction.tar.gz`

[3] Available, under GPL license, at `http://gramatica.usc.es/pln/tools/deppattern.html`.

Hash table. The word dependencies identified in the corpora and the list of seed contexts are organized in a word-context matrix (stored in memory as a hash table of non-zero values). Each item of the table represents a word (or multiword term), a seed context, and the word-context frequency observed in the corpus.

Similarity. Then, we compute dice similarity [5] of each bilingual pair of words. For each word of the source language, we select its top-N ($N = 10$) most similar ones in the target language. They are their candidate translations.

At the end of the process, we obtain the relationship $(A, C)_{corpus}$, which will be used to validate $(A, C)_{amb}$ by identifying correct ambiguous correspondences. As it was stated above, the selection of correct correspondences is the result of intersecting $(A, C)_{amb}$ with $(A, C)_{corpus}$.

4 A Case Study: The Elaboration of an English-Galician Dictionary

To verify whether the method is useful, we apply it to perform a particular task, namely to derive a new English-Galician dictionary from two existing ones. This case study has two limitations: given that Galician is a language with few electronic resources, the Galician part of our comparable corpus is considerably smaller than the English one. On the other hand, since the extraction method only works at the moment on nouns, verbs, and adjectives, the dictionary elaboration is restricted to these three grammatical categories.

4.1 The Existing Dictionaries and Generation by Transitivity

The $(English, Galician)$ dictionary was derived from both $(English, Spanish)$ and $(Spanish, Galician)$ existing dictionaries, where Spanish is the pivot language. In particular, the bilingual dictionaries we used are part of the lexical resources integrated in an open source machine translation system: OpenTrad-Apertium [2]. In fact, one of the short-mid term objectives of our experiments is to update the bilingual resources of OpenTrad in order to improve the results of the machine translation system, which is used by *La Voz de Galicia*, the sixth most widely read Spanish newspaper.

The $(English, Spanish)$ dictionary contains $8,432$ bilingual correspondences, while the $(Spanish, Galician)$ reaches $27,640$. Both dictionaries are freely available[4]. Given that the former dictionary is too small, we also made use of a Collins dictionary[5], which we call $(English\_C, Spanish\_C)$, and contains $48,637$ entries. This resource is not freely available. Note that we only count bilingual correspondences between verbs, nouns, and adjectives. All of these dictionaries were manually created by lexicographers.

[4] http://sourceforge.net/projects/apertium/files/
[5] http://www.collinslanguage.com/

Table 1. Dictionaries derived by transitivity

derived dictionaries	number of entries	ambiguous entries	not ambiguous entries	source dictionaries
$(English, Galician)$	7,687	3,890	3,797	$(Galician, Spanish)$ $(Spanish, English)$
$(English\_C, Galician)$	23,094	17,601	5,494	$(Galician, Spanish)$ $(Spanish\_C, English\_C)$

Using the strategy described above in Section 3.1, we generated two new noisy bilingual dictionaries: $(English, Galician)$ and $(English\_C, Galician)$ (see Table 1). The first raw of the table shows the different elements of $(English, Galician)$, which was derived from the two OpenTrad-Apertium dictionaries (sources). It contains 7,687 correspondences that was splitted into two subsets:

- ambiguous correspondences: $(English, Galician)_{amb}$
- not ambiguous ones: $(English, Galician)_{not-amb}$

They contain 3,890 and 3,797 entries, respectively (third and fourth columns of the table). The same was made to obtain $(English\_C, Galician)$, which was derived from $(Spanish\_C, English\_C)$ (Collins) and $(Galician, Spanish)$ (OpenTrad-Apertium). Here, the size of the resulting lexicon is larger because of the higher number of entries provided by the Collins dictionary.

4.2 Comparable Corpora and Validation

To validate the English-Galician correspondences with ambiguous words, we used the strategy described in sections 3.2 and 3.3. First, we built different non-parallel, (and somehow) comparable corpora. Then, the automatic extraction of translation equivalents were performed on those corpora.

Building three comparable corpora. The Galician part was crawled from two online daily newspapers, Vieiros and Galicia-Hoxe, which are the only general purpose newspaper written in Galician language. The crawler retrieved all news published by these newspaper since they are available in the net. We built a corpus with 35 million word tokens.

The English part was divided in three different corpora:

- 35M words selected from British National Corpus (BNC)[6],
- 35M words containing breaking news from Reuters Agency[7]
- 1M words containing news crawled from New York Times (NYT)

Given that we could not find more Galician Newspapers, to obtain a corpus size comparable to that of the English part, we decided to build 3 non-parallel corpus as follows:

[6] http://www.natcorp.ox.ac.uk/

[7] http://trec.nist.gov/data/reuters/reuters.html

BNC-based. This corpus is constituted by all Galician news (35M words) and the 35M words selected from BNC.

Reuters-based. It constituted by all Galician news and the 35M words from Reuters

NYT-based. It contains 1M words selected from the Galician corpus and 1M words crawled from NYT.

So, BNC-based and Reuters-based corpora contains the same Galician corpus while NYT-based is constituted by a small partition of that corpus. We followed this strategy because of the few electronic resources in Galician language. Let's note that the BNC-based corpus is less comparable than the others since the English part does not only contain news articles. It consists of many types of documents, including oral speech.

Extraction. The extraction method was sketched in Section 3.3. First, all texts were parsed with DepPattern to extract all word dependencies (we focused on dependencies containing verbs, nouns, or adjectives). DepPattern takes as input the output of the PoS tagger Freeling[3]. Then, a list of seed lexical syntactic contexts was generated from the largest English-Galician lexicon: $(English\_C, Galician)$. Even if it is likely to contain some odd bilingual correspondences, we consider that it is sound enough to be used for stochastic-based extraction. Then, on the basis of word dependencies and a list of contexts, three context-word bilingual matrices were created (one for each corpus). Finally, word similarity was computed on each matrix. For each English word, the 10 most similar Galician words were retained to define 10 candidate bilingual correspondences. Since similarity is an asymmetric relationship, the same was done from Galician to English. At the end of the process, we built three corpus-based bilingual lexicons: $(English\_C, Galician)_{bnc}$, $(English\_C, Galician)_{reuters}$, and $(English\_C, Galician)_{nyt}$. Table depicts the number of correspondences of each dictionary.

Table 2. Corpus-based dictionaries

dictionaries	number of entries
$(English\_C, Galician)_{bnc}$	400, 440
$(English\_C, Galician)_{reuters}$	531, 710
$(Spanish\_C, English)_{nyt}$	132, 490

Table 2 shows the results obtained. Corpus-based dictionaries are much bigger than those directly derived by transitivity, and so they contain much more noisy correspondences. The goal is to generate for each word, at least, a good bilingual correspondence which will be used to validate dubious pairs derived by transitivity. Notice also than the Reuters-based dictionary is significantly larger than the BNC-based, even if the corpus size over which the extraction was performed is the same. This is probably due to the fact that the BNC-based corpus is less comparable (it is just a "non-parallel" corpus).

Validation. To check the validity of the dubious correspondences within the ambiguity-based lexicons (i.e., containing ambiguous words), we make their intersection with the corpus-based lexicons. Table 3 shows the outputs of all possible intersections between the three corpus-based dictionaries (columns) and the two lexicons with ambiguous words (rows). The third row is the union of the two ambiguity-based dictionaries, while the last column is the union of the three corpus-based lexicons. Each absolute number is assigned a percentage: the ration between the correspondences validated (i.e., resulting of the intersection) divided by the total number of correspondences found in the dictionary with ambiguous words.

Table 3. Corpus-based validation

	bnc	*reuters*	*nyt*	**Union**
$(English, Galician)_{amb}$	1,123 (29%)	1,350 (35%)	396 (10%)	1,573 (40%)
$(English\_C, Galician)_{amb}$	2,404 (14%)	2,940 (17%)	619 (4%)	3,584 (20%)
Union	2,837 (15%)	3,475 (18%)	759 (4%)	**4,248 (22%)**

For instance, The intersection of $(English, Galician)_{amb}$ with the smallest corpus-based lexicon, $(English\_C, Galician)_{reuters}$, gives rise to 1,350 correspondences, which represent 35% of $(English, Galician)_{amb}$. Notice that successive unions of dictionaries improve the results by making the output dictionary larger. The largest lexicon was obtained by intersecting the union of the corpus-based lexicons with the union of the two ambiguity-based dictionaries: 4,248 correct entries. It represents 22% of entries found in the union of the two ambiguity-based dictionaries (19,425 entries). These results are not very far from those obtained by [10] using parallel corpora. These authors reported an experiment to derive by transitivity an English-German dictionary, whose ambiguity-based correspondences were validated using parallel corpora. The result of this checking process allowed them to validate 6,282 correspondences, which represent 26% of all candidate correspondences with ambiguous words. Even if we use non-parallel corpora, our results are very close to that score, which is very promising.

The quality of the validated correspondences is very good. No error was found.

4.3 The Final Not-noisy Lexicon

At the end of the process, we made the union of the validated correspondences with the lexicons containing unambiguous words (i.e., one-to-one correspondences). Table 4 summarizes the number of entries obtained in each step of the process. The last row shows the total number of non-noisy correspondences, 12,064, our method was able to automatically generate. This represents 47% of the total correspondences, 25,790, resulting of the union of $(English, Galician)$ with $(English\_C, Galician)$.[8]

[8] The final dictionary can be downloaded at
http://gramatica.usc.es/~gamallo/dicosFromComparable.htm

Table 4. Non-noisy dictionary

	number of entries
OpenTrad + Collins	25,790
Validated correspondences	4,248
Not ambiguous correspondences	7,816
Total not-noisy dictionary	**12,064 (47%)**

To summarize, the output dictionary is the result of the following set-theoretic operations:

$$(\boldsymbol{English, Galician})_{\boldsymbol{not-noisy}} =$$

$$((English, Galician)_{amb} \cup (English\_C, Galician)_{amb})$$
$$\cap$$
$$((English_C, Galician\_C)_{bnc} \cup (English\_C, Galician\_C)_{reuters} \cup$$
$$(English\_C, Galician\_C)_{nyt})$$
$$\cup$$
$$((English, Galician)_{not-amb} \cup (English\_C, Galician)_{not-amb})$$

Let's note that the final lexicon, even if it only contains 47% of all candidate correspondences generated by transitivity, is much larger than the smallest hand-crafted dictionary, $(English, Spanish)$, which is one of the existing dictionaries used as source to derive the new one. We generated more than 12,000 correct correspondences against 7,687 entries in the smallest existing lexicon. The quality of the derived entries is similar to those found in dictionaries built by hand by lexicographers.

5 Conclusions and Future Work

The lexicographic method proposed in this paper is entirely automatic. It does not require any manual revision to generate a new bilingual dictionary since the quality of the derived correspondences is very high, similar to that achieved by a human lexicographer. The main contribution of the method is the use of lexicon extracted from syntactically annotated comparable corpora to validate correspondences derived by transitivity. Moreover, the experiments showed that the information provided by other source dictionaries and more corpus allowed us to easily make derived dictionaries much larger without losing quality.

The main drawback of the method is to be language dependent since it requires a syntactic parser to annotate the corpus. However, in order to cope with as many language as possible, we make use of a robust multilingual parser, DepPattern, designed and implemented by our research group.

In future work, we'll integrate the resulting dictionaries into a machine translation system, namely OpenTrad-Apertium, with the aim of adapting the system to new pairs of languages.

Acknowledgments

This work has been supported by the Galician Government, within the projects with reference PGIDIT07PXIB204015PR and 2008/101.

References

1. Ahn, K., Frampotn, M.: Automataic generation of translation dictionaries using intermediary languages. In: Cross-Language Knowledge Induction Workshop of EACL 2006, Trento, Italy, pp. 41–44 (2006)
2. Armentano-Oller, C., Carrasco, R.C., Corb-Bellot, A.M., Forcada, M.L., Ginest-Rosell, M., Ortiz-Rojas, S., Prez-Ortiz, J.A., Ramrez-Snchez, G., Snchez-Martnez, F., Scalco, M.A.: Open-source portuguese-spanish machine translation. In: Vieira, R., Quaresma, P., Nunes, M.d.G.V., Mamede, N.J., Oliveira, C., Dias, M.C. (eds.) PROPOR 2006. LNCS (LNAI), vol. 3960, pp. 50–59. Springer, Heidelberg (2006)
3. Carreras, X., Chao, I., Padró, L., Padró, M.: An open-source suite of language analyzers. In: 4th International Conference on Language Resources and Evaluation (LREC 2004), Lisbon, Portugal (2004)
4. Chiao, Y.-C., Zweigenbaum, P.: Looking for candidate translational equivalents in specialized, comparable corpora. In: 19th COLING 2002 (2002)
5. Curran, J.R., Moens, M.: Improvements in automatic thesaurus extraction. In: ACL Workshop on Unsupervised Lexical Acquisition, Philadelphia, pp. 59–66 (2002)
6. Fung, P., McKeown, K.: Finding terminology translation from non-parallel corpora. In: 5th Annual Workshop on Very Large Corpora, Hong Kong, pp. 192–202 (1997)
7. Fung, P., Yee, L.Y.: An IR approach for translating new words from nonparallel, comparable texts. In: Coling 1998, Montreal, Canada, pp. 414–420 (1998)
8. Gamallo, P.: Learning bilingual lexicons from comparable english and spanish corpora. In: Machine Translation SUMMIT XI, Copenhagen, Denmark (2007)
9. Gamallo, P., Pichel, J.-R.: Learning spanish-galician translation equivalents using a comparable corpus and a bilingual dictionary. In: Gelbukh, A. (ed.) CICLing 2008. LNCS, vol. 4919, pp. 413–423. Springer, Heidelberg (2008)
10. Nerima, L., Wehrli, E.: Generating bilingual dictionaries by transitivity. In: LREC 2008, pp. 2584–2587 (2008)
11. Paik, K., Shirai, S., Nakaiwa, H.: Automatic construction of a transfer dictionary considering directionality. In: COLING 2004 Multilingual Linguistic Resources Workshop, Geneva, pp. 25–32 (2004)
12. Rapp, R.: Automatic identification of word translations from unrelated english and german corpora. In: ACL 1999, pp. 519–526 (1999)
13. Saralegui, X., San Vicente, I., Gurrutxaga, A.: Automatic generation of bilingual lexicons from comparable corpora in a popular science domain. In: LREC 2008 Workshop on Building and Using Comparable Corpora (2008)
14. Shao, L., Ng, H.T.: Mining new word translations from comparable corpora. In: 20th International Conference on Computational Linguistics (COLING 2004), Geneva, Switzerland, pp. 618–624 (2004)

15. Wehrli, E., Nerma, L., Scherrer, Y.: Deep linguistic multilingual translation and bilingual dictionaries. In: Foruth Workshop on Statistical Machine Translation, Athens, Greece, pp. 90–94 (2009)
16. Zhang, Y., Ma, Q., Isahara, H.: Building japanese-chinese translation dictionary based on EDR japanese-english bilingual dictionary. In: MT Summit XI, Copenhagen, pp. 551–557 (2007)

Hierarchical Finite-State Models for Speech Translation Using Categorization of Phrases

Raquel Justo[1], Alicia Pérez[1], M. Inés Torres[1], and Francisco Casacuberta[2]

[1] Departament of Electricity of Electronics
University of the Basque Country
{raquel.justo,alicia.perez,manes.torres}@ehu.es
[2] Departament of Information Systems and Computation
Technical University of Valencia
fcn@iti.upv.es

Abstract. In this work a hierarchical translation model is formally defined and integrated in a speech translation system. As it is well known, the relations between two languages are better arranged in terms of phrases than in terms of running words. Nevertheless phrase-based models may suffer from data sparsity at training time. The aim of this work is to improve current speech translation systems by integrating categorization within the translation model. The categories are sets of phrases either linguistically or statistically motivated. Both category and translation and acoustic models are within the framework of finite-state models. In what temporal cost is concerned, finite-state models count on efficient decoding algorithms. Regarding the spatial cost, all the models where integrated on-the-fly at decoding time, allowing an efficient use of memory.

1 Introduction

The state of the art in machine translation suggests the use of phrases as translation unit instead of running words [1]. Recent approaches in this field have introduced this paradigm into the finite-state framework [2,3]. The former proposed a finite-state approach of some constituent models such as meaning-transference and reordering models (typically found in other approaches for machine translation) involving several decodings. Alternatively, we focus on the latter approach, which arose from the so called GIATI algorithm [4], and deals with stochastic finite-state transducers (SFSTs) [5]. SFSTs have shown to be versatile models that count on efficient algorithms for *inference* from training samples [6] and *decoding* [7], being the decoding time an essential issue in speech translation. One of the main drawbacks related to these example based models has to do with the sparsity of data. There are numerous applications connected, for instance, to those involving minority languages, where bilingual training material available is quite limited. Language modeling (LM) attempts at facing this problem by means of statistical smoothing techniques [8]. Nevertheless, smoothing is still an open problem for SFSTs. As an alternative, categorisation, also referred to as structural smoothing, has proved to be of help in LM [9].

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 484–493, 2010.

The aim of this work is to combine categorization with phrase-based SFSTs in order to collect more reliable statistics over the samples and allow for generalisation at the same time. The category-based model offers a hierarchical structure of several knowledge sources, ranging from general categories made up of phrases, to rather specific phrase-based SFSTs within each category, and also word-based models for each phrase-based SFST together with acoustic models. An on-the-fly integration of these models allows for an efficient use of both space and time.

In short, the proposed translation model is driven by categories of bilingual phrases, and for each category an underlying phrase-based SFST is implemented. The categories are defined over extended phrases and as a result of it, the category-based grammar, synchronises the two languages involved. The elements within each category represent aligned bilingual data, and so are used to train an SFST. As a result, this hierarchical model can be implemented by a bilingual category-based stochastic finite-state automaton (SFSA) in which each category would entail an SFST that would help to translate into target language particular source strings associated to each category.

The organization of this paper is as follows: the underlying phrase-based finite-state transducers are described in section 2. In section 3 an extension of previous translation models is proposed. In general terms, these hierarchical models consist of a category model re-covering the phrase-based transducers allowing an on-the-fly integration of both of them. The hierarchical models were experimentally assessed as shown in section 4. Finally, section 5 summarizes the conclusions of the present work as well as the proposed lines for future investigation in this field.

2 Phrase-Based Speech Translation with Finite-State Transducers

The goal of statistical speech translation is to find the most likely string in the target language ($\widehat{\mathbf{t}}$) given the the acoustic representation $\mathbf{x}$ of a speech signal in the source language:

$$\widehat{\mathbf{t}} = \arg\max_{\mathbf{t}} P(\mathbf{t}|\mathbf{x}) \tag{1}$$

The transcription of the speech into text is an unknown variable, $\mathbf{s}$, which might be introduced as a hidden variable. By doing so and applying the Bayes' decision rule we lead to the following expression:

$$\widehat{\mathbf{t}} = \arg\max_{\mathbf{t}} \sum_{\mathbf{s}} P(\mathbf{t},\mathbf{s}|\mathbf{x}) = \arg\max_{\mathbf{t}} \sum_{\mathbf{s}} P(\mathbf{t},\mathbf{s}) P(\mathbf{x}|\mathbf{t},\mathbf{s}) \tag{2}$$

Let us assume that the acoustic representation of a speech signal only depends on its transcription in the source language; that is, assuming that the pronunciation of an utterance does not depend on the translation in other language, Equation (2) can be rewritten as:

$$\widehat{\mathbf{t}} = \arg\max_{\mathbf{t}} \sum_{\mathbf{s}} P(\mathbf{t},\mathbf{s}) P(\mathbf{x}|\mathbf{s}) \tag{3}$$

Amongst the two terms involved in (3), the lexical model, $P(\mathbf{x}|\mathbf{s})$ and the translation model, $P(\mathbf{t}, \mathbf{s})$, we focus on the latter, a joint probability model that can be approached by an SFST.

2.1 Learning a Phrase-Based Transducer

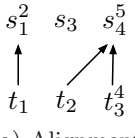
In this section is briefly summarized the training procedure of a phrase-based SFST from training samples under GIATI approach [6].

1. Phrase extraction: to start with, the translation units have to be selected. Classically, words were taken as unit, however, other unit such as word-sequences (also referred to as *phrases*) have proved to be better choice in what comes to meaning rendering from one language to the other. The phrases in each of the two languages involved can be independently extracted by means of a monolingual analysis. The segmentation process defines, thus, a finite set of atomic units for each language consisting of phrases (denoted as Σ' and Δ' for source and target languages respectively) which can be used as an alternative to the original vocabulary consisted of running words (denoted as Σ and Δ for source and target languages respectively). The phrase extraction can be either linguistic or statistically motivated.

Example: Given a pair of bilingual sentences $\mathbf{s} = s_1 s_2 s_3 s_4 s_5 = s_1^5$ and $\mathbf{t} = t_1 t_2 t_3 t_4 = t_1^4$, a segmentation procedure might give as a result the segmented sentences: $\mathbf{s} = s_1^2 s_3 s_4^5$ and $\mathbf{t} = t_1 t_2 t_3^4$. In this context, s_1 is a running word of the vocabulary Σ and s_1^2 a token in Σ'. In the same way, $s_1, s_2, s_3, s_4, s_5 \in \Sigma$, $s_1^2, s_3, s_4^5 \in \Sigma'$ and $t_1, t_2, t_3, t_4 \in \Delta$ and $t_1, t_2, t_3^4 \in \Delta'$.

2. Bilingual segmentation: Obtain the bilingual alignments taking the previously extracted phrases as atomic units. In practice, this step was carried out by means of GIZA++ free toolkit [10]. Next, from the aligned bilingual corpus, get a monotonic bilingual segmentation. As a result of this step, each training pair of sentences $((\mathbf{s}, \mathbf{t}) \in \Sigma^* \times \Delta^*)$ is converted into a single bilingual string, the so called *extended* string: $\mathbf{l} = l_1 \dots l_{|\mathbf{s}|} \in \Gamma'^* \subseteq (\Sigma' \times \Delta')^*$. Each extended token entails a single phrase in the source vocabulary, Σ', along with a sequence of zero or more phrases in the target vocabulary, Δ'.

Example: Given the segmented pair of the previous example, Fig. 1 shows the bilingual segmentation step, being λ the empty string. There, the extended token



(a) Alignment

$$\bar{l} = \underbrace{(s_1^2, t_1)}_{l_1}\underbrace{(s_3, \lambda)}_{l_2}\underbrace{(s_4^5, t_2 t_3^4)}_{l_3}$$

(b) Bilingual segmentation

Fig. 1. Monotonic bilingual segmentation

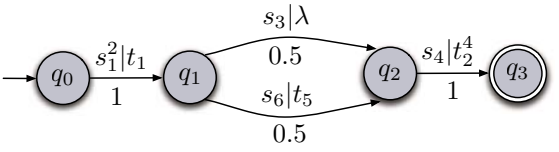
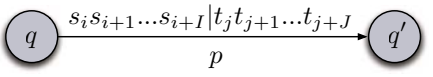
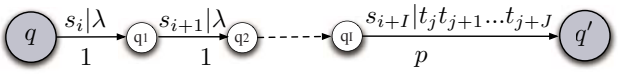


Fig. 2. Phrase-based SFST



(a) Edge of a phrase-based SFST



(b) On-the-fly integration of word-based model

Fig. 3. The integration of a left-to-right word based model in an edge of the phrase-based SFST

$l_3 = (s_4^5, t_2 t_3^4) \in \Gamma'$ consists of the source phrase ($s_4^5 \in \Sigma'$) along with a sequence of two target phrases ($t_2, t_3^4 \in \Delta'$).

3. Infer a finite-state model: Given the set of extended strings, infer a regular grammar. Note that here the vocabulary of the language model would simply be Γ'. In other words, each unit is made up of a single token from the source language and zero or more tokens from the target language (being the token a sequence of one or more running words). On account of this, the model can be used as a transducer, being the input the source language and the output its translation.

Example: The pair of sentences in previous examples could be, amongst additional data, used to train the finite-state model in Fig. 2.

4. From phrases into running words: At decoding time, the test is given in terms of running words while the edges of the SFST are labeled in terms of phrases. In order to analyze the source sentence word by word, each edge in the SFST (consisting of the source phrase s_i^{i+I}, the target phrase t_j^{j+J} and a probability p) can be expanded at decoding time on the basis of a left-to-right word-based model as shown in Fig. 3.

3 Hierarchical Translation Model $P_{M_{CP_L}}$

In this section a hierarchical translation model, $P_{M_{CP_L}}$, is defined. This model will be integrated in a speech translation system. The underlying idea is to

consider classes made up of bilingual phrases, that is, classes are made up of items inside Γ'. In this way a general class-based model is obtained and an additional phrase-based model can be obtained inside each class. That is, in an upper level a general context is considered taking into account the structure of bilingual sentences, while the lower level, where specific models inside each class are considered, carries out the translation. It has to be mentioned that a hierarchical language model under an analogous formulation was previously proposed for speech recognition [11]. The contribution of this work is indeed to extend such models for speech translation.

3.1 Definition of the Model

The probability of a pair of sentences $\mathbf{t}, \mathbf{s}$ can be written as shown in Equation (4)

$$P_{M_{CP_L}}(\mathbf{t}, \mathbf{s}) = \sum_{\mathbf{c} \in \mathcal{C}^*} \sum_{\mathbf{l} \in \Gamma'^*} P(\mathbf{t}, \mathbf{s}, \mathbf{l}, \mathbf{c}) \tag{4}$$

where $\mathcal{C}^*$ is the set of class sequences obtained from a previously defined set of classes and Γ'^* is the set of all possible sequences of bilingual phrases. That is, a sequence $\mathbf{l} = l_1 l_2 l_3$ of the example given before, where $l_1 = (s_1^2, t_1)$, $l_2 = (s_3, \lambda)$ and $l_3 = (s_4^5, t_2 t_3^4)$, belongs to this set Γ'^*.

Applying Bayes' rule:

$$P_{M_{CP_L}}(\mathbf{t}, \mathbf{s}) = \sum_{\mathbf{c} \in \mathcal{C}^*} \sum_{\mathbf{l} \in \Gamma'^*} P_{M_3}(\mathbf{t}, \mathbf{s} | \mathbf{l}, \mathbf{c}) P_{M_2}(\mathbf{l} | \mathbf{c}) P_{M_1}(\mathbf{c}) \tag{5}$$

Let us describe in more concise terms how to establish the probability distributions involved in Equation (5). To begin with, $P_{M_1}(\mathbf{c})$ was set to be an n-gram model of classes:

$$P_{M_1}(\bar{c}) \simeq \prod_{i=1}^{|\mathbf{c}|} P(c_i | c_{i-(n-1)}^{i-1}) \tag{6}$$

The second term, in Equation (5) is the probability of a sequence of bilingual phrases given a sequence of classes. Assuming zero-order models, this probability is calculated as follows:

$$P_{M_2}(\mathbf{l} | \mathbf{c}) \simeq \prod_{i=1}^{|\mathbf{c}|} P(l_i | c_i) \tag{7}$$

Finally, the third term in Equation (5) is self-evident due to the fact that each bilingual segmentation, $\mathbf{l}$, has associated a single pair of strings $(\mathbf{t}, \mathbf{s})$; note, however, that the opposite is not always satisfied, since a pair of strings may have more than one segmentation. Thus, given a bilingual segmentation, the probability of a pair is 0 unless the segmentation is congruent with the pair, being 1 in the latter case.

Summing up, the probability of a pair of sentences $\mathbf{t}$, $\mathbf{s}$ is given by Equation (8) when M_{CP_L} model is considered

$$P(\bar{t}, \bar{s}) \simeq P_{M_{CP_L}}(\mathbf{t}, \mathbf{s}) = \sum_{\mathbf{c} \in \mathcal{C}^*} \sum_{\mathbf{l} \in \Gamma'^*} \left[\prod_{i=1}^{|\mathbf{c}|} P(l_i | c_i) P(c_i | c_{i-n+1}^{i-1}) \right] \tag{8}$$

All in all, the category model, M_1, is responsible for a general context and structure selection and segmentation (bear in mind that the categories are defined over phrases), while the more specific models, M_2, defined one per category, are responsible for translation. Therefore, at decoding time, given a speech signal, all the categories and all the segmentations within the category are explored. In the end, the most successful string of categories $\mathbf{c} \in \mathcal{C}$, are obtained. Moreover, along with each category the most likely bilingual-phrase is extracted. As a result, both source and target strings, with their corresponding segmentation and categorization are obtained given a speech signal in source language.

3.2 Inference of the Model

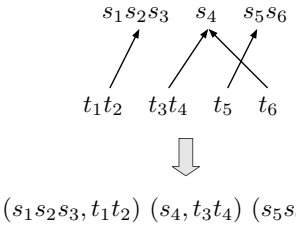
First of all a parallel training corpus in both languages is needed. The sentences of the parallel corpus were then segmented using a previously defined set of phrases in each language. Next, bilingual alignments were obtained from the segmented corpora (in source and target languages) using GIZA++. That is, the extended corpus was obtained where basic units were bilingual sequences like $l_1 = (s_1^2, t_1)$. Finally the extended corpus was classified using a previously defined set of classes and a classified corpus was obtained.

In this work a linguistic criterion was used to obtain the set of phrases. Linguistic phrases were identified by ametzagaina[1] group following the next steps described below. Furthermore, they provided us with the segmented corpus.

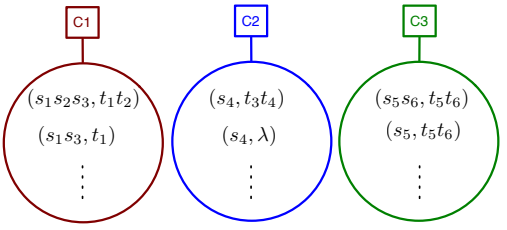
- First of all, a morpho-semantic parsing allows to assign one or more tags to each word of the corpus. These tags include information about linguistic classes such as number, declension case, verb tense and aspect, ...
- A syntactic parsing allows to remove ambiguities under the following boundary: all words within a sentence have to share compatible classes. Regular expressions and regulated exceptions are also taken into account so as to select the appropriate sets of classes.
- Once the syntactic and semantic parsing of each element is carried out unambiguously, linguistic phrases can be identified under a elementary criteria: group, all the words which share the same syntactic function whenever the frequency of that phrase in the corpus exceed a threshold. At first, just noun and verb phrases are distinguished, then, as the analysis goes ahead, more accurate groups such as composed stems, verbal periphrasis etc. are identified.

Then, a statistical criterion was employed to obtain the set of classes. The set of classes was obtained using a clustering algorithm based on a maximum likelihood approach and developed by [12]. *mkcls* is a free tool that uses this algorithm and it has demonstrate to obtain bilingual classes that outperform statistical machine translation.

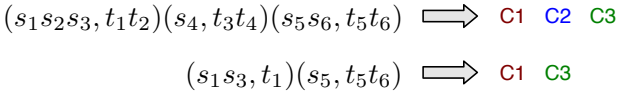
[1] Ametzagaina R&D group, member of the Basque Technologic Network, http://www.ametza.com



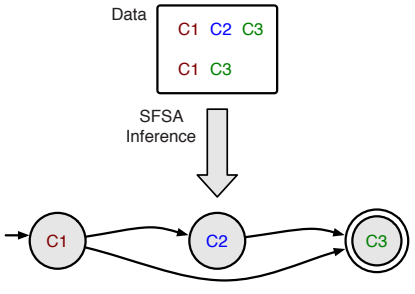
(a) Monotonic segmentation.



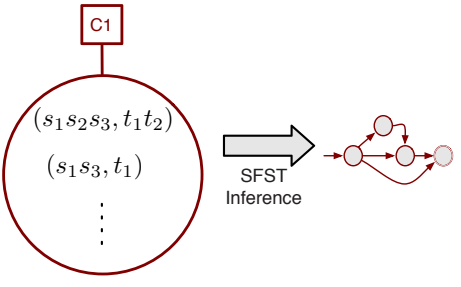
(b) Define a set of classes.



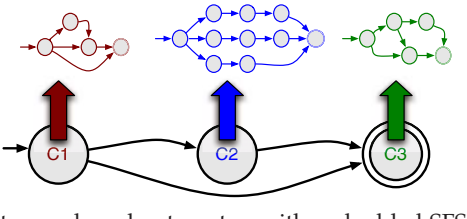
(c) Classification of data.



(d) The class-based finite-state automaton.



(e) The SFST associated to a class.



(f) Category-based automaton with embedded SFSTs.

Fig. 4. Scheme of the inference of M_{CP_L} model

From the obtained classified corpus a class-based SFSA, employed to estimate the probability of Equation (6), can be obtained. On the other hand, a phrase-based SFSA can be obtained for each class taking into account that phrases in this case are tokens of the extended vocabulary, i.e. bilingual phrases. This SFSA is employed to estimate the probability of Equation (7). Phrases made up of tokens of the extended vocabulary allows us to turn the SFSA into an SFST

considering the phrase in the source language as the input and the sequences of phrases in the target language as the output.

A general scheme of this process is shown in Figure 4.

4 Experiments

Practical issues: As it is known, the automatic speech recognition systems make use of a language model and a lexical model. For speech translation, instead, the LM is replaced by the SFST, which is, in short, a bilingual language model. The lexical model, it consists of the set of extended tokens along with their corresponding acoustic model. For a given token, the associated acoustic model is just the acoustic representation of the source phrase (the corresponding acoustics in the input language).

Task and corpus: METEUS is a text and speech corpus consisting of weather forecast reports picked up from the Internet in Basque and Spanish, the two official languages in the Basque Country. As shown in Table 1, the corpus consists of two disjoint sets for training and testing purposes respectively.

Experimental results: Experiments were carried out using a classical word-based model (M_W), a phrase-based model (M_{P_L}) and the proposed hierarchical model (M_{CP_L}). The experimental results are assessed in terms of both spatial cost and translation quality. With regard to the spatial costs, the size of the

Table 1. Main features of METEUS corpus

		Spanish	**Basque**
Training	Pair of sentences	14,615	
	Different pairs	8,445	
	Running words	191,156	187,195
	Vocabulary	702	1,135
	Singletons	162	302
	Average length	13.1	12.8
Test	Pair of sentences	1,800	
	Different pairs	500	
	Average length	17.4	16.5
	Perplexity (3-grams)	4.8	6.7

Table 2. Experimental results with word-based, phrase-based and categorized phrase-based models (M_W, M_{P_L}, M_{CP_L})

	vocabularies				**space**		**quality**	
	$\lvert\Sigma'\rvert$	$\lvert\Delta'\rvert$	$\lvert\Gamma'\rvert$	$\lvert\mathcal{C}\rvert$	states	edges	WER	BLEU
M_W	702	1,135	8,789	1	33,800	111,457	9.47	38.75
M_{P_L}	2,427	2,519	12,108	1	42,039	128,430	13.26	39.95
M_{CP_L}	2,427	2,519	12,108	4,000	25,196	4,003	10.31	39.82

models studied are given along with the size of the involved vocabularies in Table 2. Since hierarchical models consist of the on-the-fly integration of two models, a general one and a particular one, the size of the model to be located in memory has as upper threshold the size of the general model plus the size of the largest model amongst the particular ones to be integrated. This upper threshold is the one reported in Table 2 for the models using categorization.

The quality of the models was evaluated taking into account recognition in the source language and translation. That is, the models were evaluated in terms of Word Error Rate (WER) considering the recognized sentence and the reference sentence in the source language. On the other hand, the models were evaluated in terms of Biligual Evaluation Understudy (BLEU) considering the sentence given by the system and the reference sentence, both of them in the target language. The obtained results are displayed in Table 2. The experimental results show that the reduction on spatial cost of categorized model with respect to both word-based and phrase-based models is significant. Regarding the quality of the models it can be concluded that the hierarchical model provides better WER results than the M_{P_L} model but worse results than the M_W. However, in terms of BLEU the hierarchical model provides better results than the M_W and slightly worse but very similar results than the M_{P_L}. That is, the hierarchical M_{CP_L} model provides good results in terms of WER and BLEU although it does not reach the optimum result in each case. Thus, this kind of model could be very useful in applications dealing with simultaneous recognition and translation, such as transcription applications.

5 Concluding Remarks and Future Work

A hierarchical phrase-based finite-state model has been defined. In such a model the information is integrated in different depth levels, ranging from the most general model, the model based on categories, to the most specific ones, the models based on words, trough the phrase-based models. Such a model generalizes on unseen events, and as a result, they can cope with sparseness of training data, being this an essential problem to be faced in machine translation. The experimental results have shown the advantages of using the proposed models in terms of both spacial cost and system performance when simultaneous recognition and translation is required.

As it might be expected the sparsity of data increases as the amount of involved languages do. Thus, for further work it seems of special interest to apply these techniques over multi-target translation models. In addition, other kind of categorization criteria might be explored, such as taking the target phrases into account instead of the whole extended tokens. As a consequence, the category-based model would help to generalize on unseen target events instead on bilingual events. Moreover, the source language might be smoothed by means of back-off techniques.

Acknowledgments

We would like to thank Dr. Esther Alonso for her helpful discussions and contributions to this work. We would also like to like to thank Ametzagaiña group, and Josu Landa, in particular, for providing us with the linguistically motivated segmentation.

This work has been partially supported by the University of the Basque Country under grants GIU07/57, the Spanish Ministry of Science and Innovation under grants TIN2008-06856-c05-0, and by MIPRCV csd2007-00018 within the Consolider Ingenio 2010 programme.

References

1. Callison-Burch, C., Fordyce, C., Koehn, P., Monz, C., Schroeder, J.: (meta-) evaluation of machine translation. In: Proceedings of the Second Workshop on Statistical Machine Translation, Prague, Czech Republic, pp. 136–158. Association for Computational Linguistics (2007)
2. Zhou, B., Chen, S., Gao, Y.: Constrained Phrase-based Translation Using Weighted Finite State Transducer. In: Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 1, pp. 1017–1020 (2005)
3. Pérez, A., Torres, M.I., Casacuberta, F.: Speech translation with phrase based stochastic finite-state transducers. In: Proceedings of the IEEE 32nd International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2007), Honolulu, Hawaii, USA, vol. IV, pp. 113–116. IEEE, Los Alamitos (2007)
4. Casacuberta, F., Vidal, E.: Machine translation with inferred stochastic finite-state transducers. Computational Linguistics 30, 205–225 (2004)
5. Vidal, E., Thollard, F.C., de la Higuera, F.C., Carrasco, R.: Probabilistic finite-state machines - part II. IEEE Transactions on Pattern Analysis and Machine Intelligence(PAMI) 27, 1025–1039 (2005)
6. Casacuberta, F., Vidal, E.: Learning finite-state models for machine translation. Machine Learning 66, 69–91 (2007)
7. Mehryar Mohri, F.C.N.P., Riley, M.D.: AT&T FSM LibraryTM Finite-State Machine Library (2003), http://www.research.att.com/sw/tools/fsm
8. Martin, S., Ney, H., Zaplo, J.: Smoothing methods in maximum entropy language modeling. In: Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, Phoenix, AR, pp. 545–548 (1999)
9. Niesler, T.R., Woodland, P.C.: A variable-length category-based n-gram language model. In: IEEE ICASSP 1996, Atlanta, GA, vol. I, pp. 164–167. IEEE, Los Alamitos (1996)
10. Och, F.J., Ney, H.: A systematic comparison of various statistical alignment models. Computational Linguistics 29, 19–51 (2003)
11. Justo, R., Torres, M.I.: Phrases in category-based language models for Spanish and Basque ASR. In: Proceedings of the Interspeech 2007, Antwerp, Belgium, pp. 2377–2380 (2007)
12. Och, F.J.: An efficient method for determining bilingual word classes. In: Proceedings of EACL 1999, Bergen, Norway, ACL, pp. 71–76 (1999)

Drive-by Language Identification
A Byproduct of Applied Prototype Semantics

Ronald Winnemöller

Regional Computer Centre
University of Hamburg
Schlüterstr. 70
20146 Hamburg
ronald.winnemoeller@uni-hamburg.de

Abstract. While there exist many effective and efficient algorithms, most of them based on supervised n-gram or word dictionary methods, we propose a semi-supervised approach to language identification, based on prototype semantics.

Our method is primarily aimed at noise-rich environments with only very small text fragments to analyze and no training data available, even at analyzing the probable language affiliations of single words.

We have integrated our prototype system into a larger web crawling and information management architecture and evaluated the prototype against an experimental setup including datasets in 11 european languages.

1 Introduction

Crosslingual and multilingual text processing gains more and more importance, e.g. for subsequent processing steps such as word sense disambiguation but also for end user applications, such as automatic translation systems, etc. (cf. [1]).

Therefore, a common first steps in processing a text – depending on the application purpose – is language and character encoding identification. Language identification is not difficult in situations with lots of training data readily available and large, monolingual documents for analysis. Yet, there are many occasions, when this is not the case. Especially web crawlers may profit from efficient and effective language identification, because they usually operate on noisy, multilingual documents consisting of rather short text fragments and usually no training data available. Furthermore, web pages may include statements written in other languages, e.g. a non-english web page may contain english text such as "this page uses frames", "optimized for browser XYZ", etc. (cf. [2]).

In this paper, we explore the application of our Approach to Text Sense Representation (TSR) to the field of language identification in order to further demonstrate the universal nature of that methodology. The TSR approach is a philosophically and theoretically grounded methodology, that provides a suitable information representation for certain semantic and pragmatic aspects of words, phrases, sentences and other levels of text fragment granularity. In the

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 494–502, 2010.

past, it was successfully evaluated on word sense disambiguation (WSD) and other text classification problems (cf. [3,4]) but is meant to be a fundamental approach to many aspects of natural language processing in general.

2 Theoretical Considerations

The understanding of *semantics* that is applied in this paper differs substantially from conventional attribute based semantics (e.g. as employed in the Microkosmos Machine Translation System (cf. Mahesh and Nirenburg [5])) , logical form paradigms or implicit text meaning representations based on vector space models.

Instead we use the approach to prototype semantics which was described by Winnemöller (cf. [6]): according to this theory, text meaning can be based on Wittgenstein's intuitive *family resemblance* notion of the *use* of a word or text fragment in its context (cf. Wittgenstein's "Investigations" [7]). In this way, a human is - for example - able to recognize particular activities as "playing games" even though no single sharp common feature exists that is shared by every possible "game" (e.g. not every game is about winning; some, but not all games require teams, etc.). Thus our notion of text meaning is more oriented toward pragmatics and general world knowledge and covers the ability of language to assign many pragmatic aspects of meaning to particular words or text fragments, some of which might be obvious, others might be unusual. A visual interpretation of this is given in figure 1.

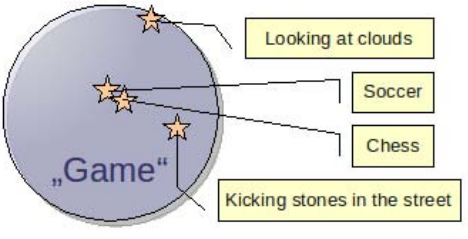


Fig. 1. Visual representation of a prototype for "game"

These "semantic spaces" can be regarded as prototypes in the sense described by Baerenfaenger [8], Meinhardt [9] and Overberg[10], with some ("typical", i.e. central) concepts located around a (ideal) nucleus and others near the periphery. These semantic spaces may also interfere with each other so that one concept might be close to the nucleus of one space and peripheral to other spaces.

In this sense, words can be seen as linguistic concepts that are part of the respective semantic spaces associated to the worlds languages, so that e.g. "*bundesweit*" is – as a word – a concept that belongs to the typical part of the german language space, whereas "*bank*" is a word that can be associated to many languages, as shown in figure 2 on the next page.

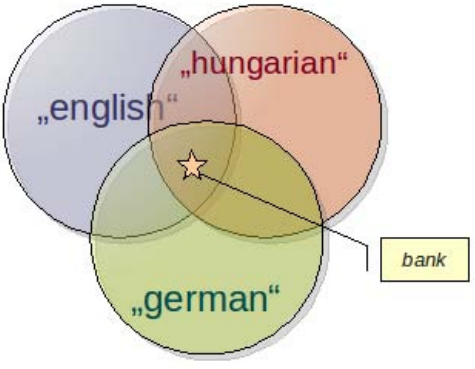


Fig. 2. Semantic spaces for “bank” as word–concept

How words are used in language is expressed by a hierarchical system of *categories* so that a unique set of categories is associated with a particular word or text fragment. It is important to note that we are not assigning predefined “sense definitions” such as WordNet synsets (cf. Miller et al [11]) to words but rather associate data structures to word instances that contain hierarchical views of many possible uses of those words. This hierarchical view is the basis of our implementation of a text meaning representation (cf. [12]).

Considering language identification purposes, in our system of categories, the possible belonging to a set of languages is just a part of the hierarchy - in the next section of this paper, we will see how this works.

3 Implementation

Any implementation of the above described theoretical ideas must be able to process content data in terms of hierarchies, in order to represent and manipulate semantic spaces as explained in the previous section.

TSRs provide a methodology to represent text meaning in such a way in a uniform and generally applicable way – while being constructed in a fully automatic fashion prior to their use.

The basic underlying data structures are tree hierarchies of labeled and weighted nodes, constructed from a web directory such as the Open Directory Project (ODP, cf. [13]).

The database generation process is divided in two distinct steps:

1. The ODP database excerpt files are downloaded, parsed and transformed into a simplified form, consisting of a sequence of records that basically contains the original path and the respective content item and is referenced by a unique primary key. What is meant by “path” and “content item” is shown in figure 3 on the facing page.

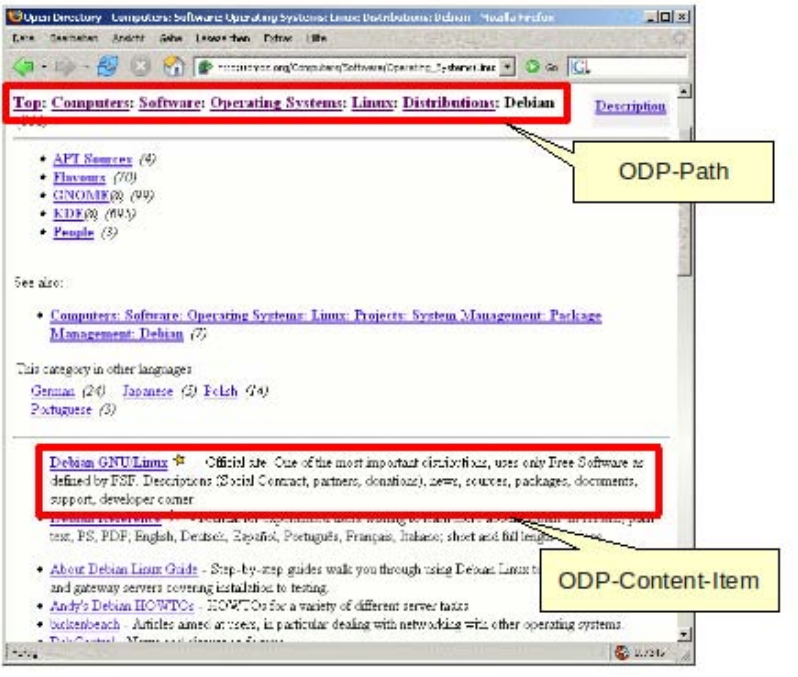


Fig. 3. ODP excerpt and structure

2. The transformed records are straightforwardly imported into an Apache Lucene[1] search index. While importing, the respective term/document frequency vectors are automatically computed for every ODP path item by the Lucene framework. Table 1 on the next page contains an example of two imported records.

Subsequently, the database can be queried through standard Lucene framework methods. TSR trees are created by issuing a content query (i.e. a query containing terms), collecting the result path entries (weighted by their respective search score) and building the tree from that path collection. In our implementation, we have also defined a number of basic TSR operations: a so-called OR operation will combine several TSRs into one by creating a TSR that consists of the *union* set of all input nodes – using the respective maximum node weight – while an AND operation will create a merged TSR that consists of the *intersection* set of all input nodes – using the respective minimum node weight (cf. [4]). These operations are similar to the equivalent operations of the fuzzy logic theory as conceived by Zadeh (cf. [14]).

Interpreting TSR trees, one can see the ODP path entries as "semantical aspects" of the query term(s). Because the ODP contains – besides domain and genre related path structures – language dependant subhierarchies, these paths

[1] Apache Lucene. http://lucene.apache.org/java/docs/index.html, 05.10.2009

Table 1. ODP data represented as Lucene documents

DocField	Content
Topic ID	`205108`
Topic Path	*Top/ Arts/ Movies/ Titles/ 1/ 10_Rillington_Place*
Content	10 Rillington Place - UK film (1971), directed by Richard Fleischer, from the book by Ludovic Kennedy. The true story of John Christie - the serial killer. Stars Richard Attenborough as Christie, with John Hurt and Judy Geeson.
Topic ID	`434222`
Topic Path	*Top/ World/ Deutsch/ Computer/ Bildung*
Content	M-Learning - Mobile Endgeräte in der Bildung - PDAs im schulischen Einsatz - Website zum M-Learning Projekt für Lehrer, Schüler und andere Personen aus dem Bildungsbereich zum Thema "m-learning mit mobilen Endgeräten (mobile Devices) in edukativen Bereichen".

can be used to determine the distribution of the query terms across all >70 languages contained within the ODP. For example, german texts are usually found within the `</top/world/german>` subtree. Some words have meaning in several languages (e.g. "bank", as in figure 2 on page 496) and will therefore produce results in these languages – weighted by the term frequencies of their respective subtree occurrences. These results are the basis of our technique of language identification: retrieving TSR trees for short text fragments and computing their language subtree occurrence frequencies (The same method can be used to compute "belongingness" to a particular topic, genre or domain – but we will not elaborate on this here).

For computational reasons, we remove the structural aspects of the TSRs so that only a vector representation remains, the TSR vectors.

4 Related Work

Most approaches on language identification rely on supervised n-gram or dictionary based methods:

Martins, Bruno and Silva implemented a character n-gram frequency based algorithm, originally invented by Cavnar and Trenkle (cf. [15]) but enhanced by a more efficient similarity measure (cf. [2]). The authors evaluation is based on documents from 12 languages with 500 test documents per language (fragment size was "whole document"). Their results are quite comparable to the results presented in this paper, yet we only need about 10 words to achieve these results and do not need to employ specific model training.

Zavarsky, Mikami and Wada based their approach on quad-gram vector distance and character encoding identification, coming from a project that is aimed at discovering whole-web language and encoding distributions (cf. [16]). Unfortunately they did not provide any results but their notion of vector distances as

classification measure seems related to the use of vector operations in the TSR methodology.

Singh used pruned character n-grams for language identification – either alone or augmented by word n-grams (cf. [17]). "Pruning" means that only the n-grams with the highest frequencies remain, while all others are dismissed. Training was based on documents of about 2.500 – 10.000 words. The author evaluates and compares several distance measures. In his evaluation he uses 39 languages and 19 character encodings: on average, there were about 22.600 words per language/encoding used for training. Because of the authors goal of comparing distance measures, his results are not really comparable to the ones presented in this paper (In our paper, we employed only a very simple distance measure but it appears probable that our results might gain from using a more elaborate technique here).

Rehurek and Kolkus proposed two techniques, one based on using n-grams in combination with an expectation maximization algorithm, the other being a dictionary based method (cf. [18]). The algorithms are trained on wikipedia dumps and are tuned specifically for web data. 9 languages. able to identify unknown languages through threshold settings. While the algorithm is completely different from ours, their experimental setup and results are comparable. In this respect we conclude that our results can be regarded state of the art.

Apart from these "supervised" examples, there are also unsupervised methods:

Biemann and Teresniak criticise supervised methods in that they can only choose from a predefined set of languages, which means that they rely on the quality (and quantity) of appropriate training data and that there is usually no "unknown language" option, so that these algorithms, being faced with unlearned language material, will necessarily fail (cf. [19]). Their proposed method is a sentence-based co-occurrence graph clustering algorithm that is able to identify clusters of languages and also creates frequencies of different language belongings for each word. The algorithm apparently excels when analyzing monolingual documents of at least 100 sentences where it scores over 98% (F-measure, i.e. combination of precision and recall) for every language tested. This related work is particularly interesting, since it shows several advantages over supervised methods which it shares with our approach, namely the ability to work without training data and to identify "unknown" languages. Yet our method can only be regarded as "semi-supervised" since it uses a pre-built database and does not purely rely on clustering techniques. On the other hand, it is able to name the identified languages, which a clustering algorithm cannot.

5 Evaluation

For evaluation, we constructed a multilingual corpus through fetching 1000 randomly selected (monolingual) documents per language from Wikipedia, using the respective "random article"-link of each target language[2]. From these

[2] English language "random article".
`http://en.wikipedia.org/wiki/Special:Random` 02.10.2009

Table 2. ODP language distribution

Language	Code	References
english	eng	roughly 600.000
german	deu	511.642
french	fra	234.306
italian	ita	200.394
spanish	spa	161.307
dutch	nld	97.513
polish	pol	77.990
danish	dan	50.883
turkish	tur	42.718
swedish	swe	38.509
finnish	fin	10.670
unknown	??	n.a.

documents we extracted the first paragraph for use in our test corpus. For this reason, all individual experiments are based on a corpus of about 11.000 documents except the one based on paragraph-size analysis (40 terms) – here, only 5.585 test instances could be used (all others were too short).

From the ~70 languages contained in our index database, we tested 11 languages plus one "unknown" language (this really is czech but due to an error during data import, czech is not recognizable as such by our present system/database). For the languages of our testset, the distribution within ODP is (in terms of absolute ODP reference counts) summarized in table 2.

The experimental setup was very straightforward – we ran our algorithm over all test instances and gathered accuracy data (i.e. the percentage of correct identifications). As for the identification of an "unknown" language, there was no need for changing that setup because the respective testing data records were also tagged "unknown". We conducted 6 experiments, 5 of them identified exactly one candidate as text fragment language and one experiment (the "Term occurrence" experiment) output a set of languages for every term, computing whether the correct language was at all present in this set. The results of our experiments are shown in table 3 on the facing page.

Looking at the specific data, there are some noteworthy observations:

- It is interesting to see that certain languages show very good results, namely english and german, while other languages perform relatively poor – especially french and finnish. We explain the finnish results by the amount of words not covered by the ODP but for french we assume a relatively large overlap of the french and english languages, especially in a web based environment.
- Even though the other ~59 languages are not part of the test environment, they show up occasionally. For example, danish sometimes is mistaken for norwegian, even though norwegian is not part of the setup.

Table 3. Accuracy (%) for different number of terms

Setup	deu	nld	ita	dan	spa	fra	tur	swe	pol	eng	fin	??
Term occ.	83	77	78	61	85	75	67	63	69	98	49	---
1 Term	80	69	75	52	80	66	66	57	67	98	49	64
2 Terms	76	60	78	63	68	52	68	64	83	99	63	79
5 Terms	94	89	89	77	91	76	89	80	97	100	84	---
10 Terms	98	94	96	88	97	84	94	89	98	100	93	---
40 Terms	99,8	98,8	99,8	98,4	99,8	95,6	98,8	99,2	100	100	99,6	99,8

– When examining the 5-terms and 10-terms german subcorpus results, all misclassifications can be explained by poor input data, consisting only of proper names or numeric terms – these usually default to "english" language because of the larger amount of english text within the ODP and therefore the higher score for that language (poor input data quality was also identified as major factor by Rehurek and Kolkus, cf. [18]).

  In future applications, these misclassifications can probably be avoided by using a named entity tagger as preprocessing step.
– Examination of the 2-terms dutch subcorpus reveals that very short fragments of this language are often dominated by english or german words — in this case, as well as for the french language, we suspect a high language overlap.

6 Conclusions

Concluding from our experimental results, the TSR approach provides an effective methodology not only for text classification but also for language identification purposes at the benefit of analyzing text fragments in a uniform fashion and in only one run.

The basis of prototype semantics has been further proven a good and fruitful theory for augmenting conventional semantics on the field of automated text analysis. As the TSR approach and it's understanding of prototype semantics aim to provide a methodology for specifying the use of language, the task of language identification is re-formulated into analyzing the language-identifying context of the use of particular linguistic fragments, i.e. terms, words, phrases, and sentences. This method, of course can be extended onto virtually any similar context identification goal, e.g. for syntactic structure and possibly even for ontological aspects. It is this unifying ability, that sets the TSR approach apart as meta-theoretic yet pragmatic methodology.

In the future, we plan to extend our language identification method to other areas of TSR application, specifically onto domain and genre identification.

References

1. Pedersen, T., Mihalcea, R.: Advances in word sense disambiguation. In: 43rd Annual Meeting of the Association for Computational Linguistics, University of Michigan, Ann Arbor, USA (2005)

2. Martins, B., Silva, M.J.: Language identification in web pages. In: SAC 2005: Proceedings of the 2005 ACM symposium on Applied computing, pp. 764–768. ACM, New York (2005)
3. Winnemöller, R.: Knowledge based feature engineering using text sense representation trees. In: International Conference RANLP - 2005, Borovets, Bulgaria (2005)
4. Winnemöller, R.: Using meaning aspects for word sense disambiguation. In: 9th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing), Haifa, Israel (2008)
5. Mahesh, K., Nirenburg, S.: A situated ontology for practical nlp. In: Workshop on Basic Ontological Issues in Knowledge Sharing, International Joint Conference on Artificial Intelligence (IJCAI 1995), Montreal, Canada (1995)
6. Winnemöller, R.: Zur bedeutungsorientierten Auflösung von Wortmehrdeutigkeiten - Vorschlag einer Methodik. PhD thesis, University of Hamburg, Hamburg, Germany (2009)
7. Wittgenstein, L.: Philosophische Untersuchungen. In: Werkausgabe, B.I. (ed.) Frankfurt am Main. Suhrkamp Verlag (1984)
8. Bärenfänger, O.: Merkmals- und prototypensemantik: Einige grundsätzliche überlegungen. Linguistik online 12 (2002)
9. Meinhardt, H.J.: Invariante, variante und prototypische merkmale der wortbedeutung. Zeitschrift für Germanistik 5, 60–69 (1984)
10. Overberg, P.: Merkmalssemantik vs. prototypensemantik - anspruch und leistung zweier grundkonzepte der lexikalischen semantik. Master's thesis, Universität Münster (1999)
11. Miller, G.A., Fellbaum, C., Tengi, R., Wolff, S., Wakefield, P., Langone, H., Haskell, B.: Wordnet - a lexical database for the english language (2005), http://www.cogsci.princeton.edu/~wn/index.shtml
12. Winnemöller, R.: Constructing text sense representations. In: Hirst, G., Nirenburg, S. (eds.) ACL 2004: Second Workshop on Text Meaning and Interpretation, Barcelona, Spain, pp. 17–24. Association for Computational Linguistics (2004)
13. Netscape Communications Corporation: Open directory project (2004), http://dmoz.org
14. Zadeh, L.: Fuzzy sets. Information Control 8, 338–353 (1965)
15. Cavnar, W.B., Trenkle, J.M.: N-gram-based text categorization. In: Proceedings of SDAIR 1994, 3rd Annual Symposium on Document Analysis and Information Retrieval, pp. 161–175 (1994)
16. Zavarsky, P., Mikami, Y., Wada, S.: Language and encoding scheme identification of extremely large sets of multilingual text. In: Conference Proceedings: the tenth Machine Translation Summit, Phuket, Thailand, pp. 354–355 (2005)
17. Singh, A.K., Surana, H.: Can corpus based measures be used for comparative study of languages? In: Proceedings of Ninth Meeting of the ACL Special Interest Group in Computational Morphology and Phonology, Prague, Czech, pp. 40–47 (2007)
18. Rehurek, R., Kolkus, M.: Language identification on the web: Extending the dictionary method. In: Gelbukh, A. (ed.) CICLing 2009. LNCS, vol. 5449, pp. 357–368. Springer, Heidelberg (2009)
19. Biemann, C., Teresniak, S.: Disentangling from babylonian confusion - unsupervized language identification. In: Gelbukh, A. (ed.) CICLing 2005. LNCS, vol. 3406, pp. 762–773. Springer, Heidelberg (2005)

Identification of Translationese: A Machine Learning Approach

Iustina Ilisei[1], Diana Inkpen[2], Gloria Corpas Pastor[3], and Ruslan Mitkov[1]

[1] Research Institute in Information and Language Processing,
University of Wolverhampton, Wolverhampton, United Kingdom
iustina.ilisei@wlv.ac.uk, r.mitkov@wlv.ac.uk
[2] School of Information Technology and Engineering,
University of Ottawa, Ottawa, Canada
diana@site.uottawa.ca
[3] Department of Translation and Interpreting,
University of Málaga, Málaga, Spain
gcorpas@uma.es

Abstract. This paper presents a machine learning approach to the study of translationese. The goal is to train a computer system to distinguish between translated and non-translated text, in order to determine the characteristic features that influence the classifiers. Several algorithms reach up to 97.62% success rate on a technical dataset. Moreover, the SVM classifier consistently reports a statistically significant improved accuracy when the learning system benefits from the addition of simplification features to the basic translational classifier system. Therefore, these findings may be considered an argument for the existence of the Simplification Universal.

1 Introduction

The characteristics exhibited by translated texts compared to non-translated texts have always been of great interest in Translation Studies. Translated language is believed to manifest certain universal features, as a consequence of the translation process. Translations exhibit their own specific lexico-grammatical and syntactic characteristics [1–3]. These"fingerprints"left by the translation process were first described by Gellerstam and named translationese [4]. Fairly recently, it has been stated that there are common characteristics which all translations share, regardless of the source and the target languages [5]. Toury proposed two laws of translation: the law of standardisation and the law of interference [6], and it was Baker who defined four possible translation universals [5, 7]. However, the notion of these universals is based on intuition and introspection. Laviosa continued this line of research by proposing features for simplification in a corpus-based study [8]. Despite some evidence of the existence of such a phenomenon, there is still a remarkable challenge in defining the features which characterise the simplification universal.

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 503–511, 2010.

The aim of this study is twofold: first, to model a language-independent learning system able to distinguish between translated and non-translated texts. The main advantages of such a data representation are obvious: the system has a wide applicability for other languages, and thus, the "universal" label of this hypothesis is easier to investigate. Second, the goal is to investigate the validation of the simplification hypothesis and to explore the characteristic features which most influence the translated language.

2 Related Work

The simplification universal is described as the tendency of translators to produce simpler and easier-to-follow texts [5]. The follow-up research methodology in the investigation of translation universals is based on comparable corpora, and some empirical results sustaining the universal were provided [8]. Laviosa investigates lexical patterns for English and the obtained results show a relatively low proportion of lexical words over function words in translated texts, and a high proportion of high-frequency words compared to the low-frequency words. Moreover, great repetition of the most frequent words and less variety in the most frequently used words has been emphasised [9].

Recently, a corpus-based approach which tests the statistical significance of features proposed to investigate the simplification universal has been exploited for Spanish [10, 11]. The experiments were on both the medical and technical domains, and the translated texts were produced by both professional and semi-professional translators. In [10] the simplification universal is confirmed only for lexical richness. The results for the following parameters appear to be against this universal: complex sentences, sentence length, depth of syntactical trees, information load, senses per word. The experiments in [11] revealed that translated texts exhibit lower lexical density and richness, seem to be more readable, have a smaller proportion of simple sentences and appear to be significantly shorter, and discourse markers were used significantly less often. Simplification fingerprints were found on the technical translation and seemed to show that texts written by non-professional translators do not have such simplification traits.

A different perspective over the same line of research is employed by Baroni and Bernardini [12], who exploit machine learning techniques for the task of classifying Italian texts as translated or non-translated texts. The results obtained show that the SVM classifier depends heavily on lexical cues, the distribution of n-grams of function words and morpho-syntactic categories in general, and on personal pronouns and adverbs in particular. Therefore, it is proved that shallow data representations can be sufficient to automatically distinguish professional translations from non-translated texts with an accuracy above the chance level, and hypothesise that this representation captures the distinguishing features of translationese. Moreover, human accuracy on the same task seems to be much lower compared to the success rate of the learning system. In this study, the exploitation of n-grams indicators is avoided because of their language dependence.

3 Methodology

The approach in this paper is based on supervised machine learning techniques which aim to distinguish between translated and non-translated, spontaneous texts. Therefore, a training dataset and a test dataset were constructed comprising random instances from both classes. By using Weka [1] [13, 14], the classifiers are trained including and excluding the features proposed for the simplification universal within the data representation, and afterwards the T-test evaluates the statistical significance between the accuracies obtained in both cases. Therefore, if the success rate of the learning system including the simplification indicators in the feature vector is high, then it may be stated that this is an argument for the existence of the simplification universal.

As is proposed, these universals can be studied by comparing translations with non-translations in the same language [15], thus strictly avoiding any foreign interference [16]. The resource exploited is the monolingual comparable corpora for Spanish language extensively described in [10], which comprise three pairs of translated and non-translated texts, as follows:

- Corpus of Medical Translations by Professionals (MTP), which is comparable to the Corpus of Original Medical texts by Professionals (MTPC);
- Corpus of Medical Translations by Students (MTS), which is comparable to the Corpus of Original Medical texts by Students (MTSC);
- Corpus of Technical Translations by Professionals (TT), which is comparable to the Corpus of Original Technical texts by Professionals (TTC).

The training set comprises 450 randomly selected instances and the overall test set has 150 randomly selected instances from all three pairs of comparable texts. The same proportion of texts is kept for both selected training and test datasets. In order to extract the feature vector for the learning process, all the texts of the corpora were parsed with the Connexor Machinese [17], which provides the dependency parser for the Spanish language model.

The learning system exploits twenty-one language-independent features. Some of these parameters are designed to capture the simplicity characteristic of texts, which is expected to improve the performance of the classifiers, on the assumption that the simplification universal is valid. Additionally, in order to prevent learning to classify according to the topic of a text, the current approach avoids the bag-of-words model.

The first set of features which grasp general characteristics of texts, considered to stand for the translationese effect, are the following:

- the proportion in texts of grammatical words, nouns, finite verbs, auxiliary verbs, adjectives, adverbs, numerals, pronouns, prepositions, determiners, conjunctions, and the proportion of grammatical words to lexical words.

For the last parameter above, the following parts of speech are considered to belong to the class of grammatical words: determiners, prepositions, auxiliary

[1] http://www.cs.waikato.ac.nz/ml/weka/

verbs, pronouns, and interjections. Lexical words, also known as content words, are represented by nouns, verbs, adjectives, adverbs, and numerals.

The data representation for the learning system comprises all the above parameters and includes the proposed simplification features described below:

1. average sentence length,
2. sentence depth as the parse tree depth,
3. proportion of simple sentences, complex sentences and sentences without any finite verb,
4. ambiguity as the average of senses per word[2],
5. word length as the proportion of syllables per word,
6. lexical richness,
7. information load as the proportion of lexical words to tokens.

Most of the features employed (1-4, 6-7) in the data representation were originally proposed in [10] for the investigation of the simplification universal. The experiments in [11] deal with the universal in a slightly different manner (e.g. using readability measures), hence the results previously mentioned are slightly different from the ones reported in [10] but by and large compatible.

The next stage of the study consists of evaluation on separate datasets corresponding to each corpus domain, in order to determine the performance of the text classification for each type and genre. Therefore, the system is trained on the entire training dataset and it is tested on the following datasets: the technical domain written by professional translators dataset, and on the medical domain written by students dataset. As the medical domain written by professionals dataset has insufficient class instances, no separate dataset was considered.

The machine learning classifiers applied on the categorisation task are the following: Jrip, Decision Tree, Naïve Bayes, BayesNet, SVM, Simple Logistic and one meta-classification algorithm: the Vote meta-classifier with the Majority Voting combination rule, which considers the Decision Tree, Jrip and Simple Logistic classifiers output. To assess the statistical significance of the improvement of the machine learning system when including simplification features compared to the learning system without these features, the paired two-tailed t-test has been applied with 0.5 significance level.

4 Evaluation

The accuracy obtained with the data representation including the simplification features is compared to the accuracy obtained by the system without the simplification features. The assumption is the following: if the lack of simplification features causes a statistically significant difference, this can be considered as an argument for the existence of the simplification universal.

[2] Note that the ambiguity parameter is obtained exploiting the Spanish Wordnet synsets [18].

Table 1. Classification Results: Accuracies for several classifiers

	Including Simplification Features		Excluding Simplification Features	
Classifier	*10-fold cross-validation*	*Test set*	*10-fold cross-validation*	*Test set*
Baseline	65.33%	64.86%	65.33%	64.86%
Naive Bayes	*76.67%	79.05%	69.33%	75.00%
BayesNet	78.67%	79.73%	75.11%	77.03%
Jrip	79.56%	83.11%	73.33%	77.03%
Decision Tree	78.22%	81.76%	78.22%	81.76%
Simple Logistic	*77.33%	83.11%	71.11%	80.41%
SVM	*79.11%	*81.76%	69.33%	73.65%
Meta-classifier	*80.00%	87.16%	73.33%	85.81%

4.1 Classification Results

The accuracies for the 10-fold cross-validation evaluation on the training data and the accuracy for the test dataset evaluation are reported in Table 1. The training dataset comprises 450 instances, with 156 for the translation class and 294 for non-translation class instances, and the test dataset comprises 148 instances, with 52 for the translation class and 96 for non-translation class.

An asterisk by the accuracy value indicates that a statistically significant improvement is registered when including the simplification features compared to the same classifier without the simplification features. There are no worse cases, therefore only improvement is marked.

The baseline classifier, ZeroR, considers the majority class from the dataset. As the majority class is the non-translated class, the baseline is 64.5%. The meta-classifier, which takes the majority vote between Decision Tree, Jrip and Simple Logistic classifiers, reaches 87.16% for the randomly selected test set and 80% for 10 fold cross-validation.

4.2 Experiments on Separate Test Datasets

The experiments continue with the evaluation of the system on three subsets of the test set according to the three types of corpora: the test set pair 1 for MTP-MTPC, test set pair 2 for MTS-MTSC, and test set pair 3 for TT-TTC. The same proportion of class instances is kept as in the previous stage: test set pair 2 has 66 and 36 instances for non-translated and translated class, respectively; test set pair 3 has 28 non-translated class instances and 14 translated class instances. As pair 1 has only 5 instances for both classes, it is not relevant to test the classifiers on such a small dataset.

In Table 2, the accuracies for the classifiers tested on these three datasets are reported. As expected from the previous experiment, none of them report worse results when adding the simplification features. Moreover, the SVM classifier shows a statistically significant improvement for the technical domain written by professionals, reaching the highest performance of 97.62% accuracy. Nevertheless, BayesNet, Simple Logistic, and the meta-classifier register similar values

Table 2. Classification accuracy results on the medical and technical test datasets

	Including Simplification Features		Excluding Simplification Features	
Classifier	MTS-MTSC	TT-TTC	MTS-MTSC	TT-TTC
Baseline	64.71%	66.67%	64.71%	66.67%
Naive Bayes	71.57%	95.24%	71.57%	80.95%
BayesNet	73.53%	97.62%	71.57%	92.86%
Jrip	79.42%	95.24%	72.55%	92.86%
Decision Tree	77.45%	92.86%	75.49%	95.24%
Simple Logistic	77.45%	97.62%	79.41%	83.33%
SVM	75.49%	*97.62%	74.51%	69.05%
Meta-classifier	82.35%	97.62%	78.43%	92.86%

for the same pair (technical domain), not statistically significant according to the t-test.

The learning system retrieves outstanding results for the technical domain, with all the classifiers having above 95% success rates.

Aiming to determine the most salient features which led to these results, the following subsection provides the feature analysis output from the learning system and the attribute evaluators selection.

4.3 Results Analysis

A deeper result analysis is undertaken and the rules considered by the classifiers are described in figures 1 and 2. The Jrip and the Decision Tree classifiers are two algorithms which provide an intuitive output for analysis [19].

As can be noticed from the pruned tree output in Figure 1, the most informative feature is undoubtedly lexical richness, followed by sentence length and proportion of grammatical words by lexical words. Both lexical richness and sentence length are features considered to be indicative of the simplification hypothesis, widely discussed and studied in the past decade. Sentence length is a characteristic which posed a certain difficulty in its interpretation in the study undertaken by [10, 11]. Additionally, the proportion of grammatical words and lexical words makes a valuable contribution in the classification. This is a feature first proposed for this task, and considered to stand for the translationese phenomenon in general, rather than for any particular universal. On the third level is the proportion of pronouns and conjunctions in texts.

The rules observed by the JRip classifier, according to which the classifier takes its decisions, is presented in Figure 2.

The first rule considers lexical richness and proportion of finite verbs, whilst sentence length, word length, proportion of nouns and prepositions appear in the second and third rule output from this classifier.

Furthermore, the feature selection evaluators output is exploited in order to see the ranking of the attributes, regardless of any classifier. The Information Gain and Chi-square algorithms provide the information from Figure 3. The

```
lexicalRichness <= 0.16
|   sentenceLength <= 16.81: non-translation (30.0)
|   sentenceLength > 16.81
|   |   ratioProns <= 0.05
|   |   |   lexicalRichness <= 0.11: translation (46.0/1.0)
|   |   |   lexicalRichness > 0.11
|   |   |   |  ..................................
|   |   |   ratioNumerals <= 0.03: non-translation (15.0/1.0)
|   |   |   ratioNumerals > 0.03
|   |   |   |  ..................................
lexicalRichness > 0.16
|   grammsPerLexics <= 0
|   |   ratioConjs <= 0.03
|   |   |   ratioAdjectives <= 0.09: translation (9.0/1.0)
|   |   |   ratioAdjectives > 0.09: non-translation (2.0)
|   |   ratioConjs > 0.03
|   |   |   ratioNumerals <= 0.06
|   |   |   |  ..................................
|   |   |   ratioNumerals > 0.06
|   |   |   |  ..................................
|   grammsPerLexics > 0
|   |   lexicalRichness <= 0.31: non-translation (88.0/2.0)
|   |   lexicalRichness > 0.31
|   |   |   ratioConjs <= 0.04: non-translation (11.0)
|   |   |   ratioConjs > 0.04
|   |   |   |   ratioNumerals <= 0.04
|   |   |   |   |  ..............................
|   |   |   |   ratioNumerals > 0.04: translation (4.0)
```

Fig. 1. Pruned tree output from the Decision Tree classifier

```
Rule 1: (lexicalRichness <= 0.16) and (ratioFiniteVerbs <= 0.08)
             => class=translation (86.0/15.0)
Rule 2: (simpleSentences >= 0.3) and (wordLength <= 2.46) and
             (sentenceLength >= 20.7) and (ratioNouns >= 0.33)
             => class=translation (24.0/3.0)
Rule 3: (ratioFiniteVerbs <= 0.09) and (ratioPreps <= 0.13)
             => class=translation (17.0/6.0)
Rule 4: => class=non-translation (323.0/53.0)
```

Fig. 2. JRip classifier rules output

notation of the sentences without any finite verb is marked in the program as the zeroSentences attribute.

As can be observed, the two feature selection algorithms acquire approximately the same knowledge, particularly for the top seven attributes. The slight variation in the ranking is minimal, and the most intriguing part is that ambiguity is listed as one of the less informative features in the classification system. Moreover, this attribute is consistently disregarded by both decision tree and Jrip classifiers. Therefore, the assumption that the more ambiguous a text is, the more probable that it is a non-translation is rejected by the employed learning system, in line with one of the findings by [10].

In addition, analysing the confusion matrix, a high proportion of misclassified instances are due to the labelling of the translated text as non-translations. Thus, it can be asserted that this behaviour is expected, as the main purpose of a translation is to be easily confused with a spontaneous, non-translated text.

Table 3. Attributes Ranking Filters

Information Gain	**Chi squared**
0.1 lexicalRichness	61.79 lexicalRichness
0.08 grammsPerLexics	43.55 grammsPerLexics
0.07 ratioFiniteVerbs	39.28 ratioFiniteVerbs
0.05 ratioNumerals	33.12 ratioNumerals
0.05 ratioAdjectives	23.89 ratioAdjectives
0.04 sentenceLength	23.55 sentenceLength
0.04 ratioProns	22.64 ratioProns
0.03 simpleSentences	21.07 wordLength
0.03 wordLength	19.74 simpleSentences
0.03 grammaticalWords	15.37 zeroSentences
0.03 zeroSentences	13.79 ratioNouns
0.02 ratioNouns	11.46 lexicalWords
.....	.....

5 Conclusions and Further Research

This paper presents a new study on the investigation of universals of translations in Spanish. A supervised learning approach is employed to identify the most informative features that characterise translations compared to non-translated texts. The learning system is trained on two domains, medical and technical, and the novelty consists of its language-independent data representation. The outstanding accuracy provided by several classifiers is evidence that translations can indeed be identified.

On the categorisation task, the algorithms achieve an accuracy of 87.16% on a test set, and reach up to 97.62% for separate test datasets from the technical domain. However, the removal of the features related to simplification from the machine learning process leads to decreased accuracy of the classifiers. Therefore, the retrieved results may be considered as an argument for the existence of the simplification universal. A performance analysis of our classifiers' output reveals that the learning system relies heavily on the following features: lexical richness, proportion of grammatical words to lexical words, sentence length, word length and some morphological attributes like nouns, pronouns, finite verbs, conjunctions and prepositions.

The main research direction to be tackled in the future is the investigation of the other translation universals. An additional subject of investigation will be a deeper analysis of the indicative features which influence translated language.

References

1. Borin, L., Prütz, K.: Thorough a dark glass: part of speech distribution in original and translated text. In: Computational Linguistics in the Netherlands, pp. 30–44. Rodopi, Amsterdam (2001)
2. Hansen, S.: The Nature of Translated Text. Saarland University, Saarbrücken (2003)

3. Teich, E.: Cross-linguistic Variation in System and Text. Mouton de Gruyter, Berlin (2003)
4. Gellerstam, M.: Translationese in Swedish novels translated from English. Translation Studies in Scandinavia. CWK Gleerup, Lund (1986)
5. Baker, M.: Corpus Linguistics and Translation Studies – Implications and Applications. In: Text and Technology: In Honour of John Sinclair, pp. 233–250. John Benjamins, Amsterdam (1993)
6. Toury, G.: Descriptive Translation Studies and Beyond. John Benjamins, Amsterdam (1995)
7. Baker, M.: Corpus-based Translation Studies: The Challenges that Lie Ahead. In: Terminology, LSP and Translation: Studies in Language Engineering in Honour of Juan C. Sager, pp. 175–186. John Benjamins, Amsterdam (1996)
8. Laviosa, S.: Corpus-based Translation Studies. Theory, Findings, Applications. Rodopi, Amsterdam (2002)
9. Laviosa, S.: Core patterns of lexical use in a comparable corpus of English narrative prose. In: The Corpus-Based Approach, pp. 557–570. Les Presses de L'Université de Montréal, Montréal (1998)
10. Corpas, G.: Investigar con corpus en traducción: los retos de un nuevo paradigma. In: Frankfurt am Main. Peter Lang, Berlin (2008)
11. Corpas, G., Mitkov, R., Afzal, N., Pekar, V.: Translation universals: Do they exist? a corpus-based nlp study of convergence and simplification. In: Proceedings of the AMTA, Waikiki, Hawaii (2008)
12. Baroni, M., Bernardini, S.: A new approach to the study of translationese: Machine-learning the difference between original and translated text. Literary and Linguistic Computing 21(3), 259–274 (2006)
13. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.: The weka data mining software: An update. SIGKDD Explorations 11(1), 10–18 (2009)
14. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)
15. Olohan, M.: Introducing Corpora in Translation Studies. Routledge (2004)
16. Pym, A.: On Toury's laws of how translators translate. In: Beyond Descriptive Translation Studies, Benjamins, pp. 311–328 (2008)
17. Tapanainen, P., Jarvinen, T.: A non-projective dependency parser. In: Proceedings of the 5th Conference of Applied Natural Language Processing, Washington D.C, USA, pp. 64–71 (1997)
18. Verdejo, F.M.: The spanish wordnet. Technical report, Universitat Politenica de Catalunya, Madrid, Spain (1999)
19. Quinlan, J.R.: Induction of decision trees. Machine Learning 1, 81–106 (1986)

Acquiring IE Patterns through Distributional Lexical Semantic Models

Roberto Basili, Danilo Croce, Cristina Giannone, and Diego De Cao

Dept. of Computer Science,
University of Roma Tor Vergata, Roma, Italy
{basili,croce,giannone,decao}@info.uniroma2.it

Abstract. Techniques for the automatic acquisition of Information Extraction Pattern are still a crucial issue in knowledge engineering. A semi supervised learning method, based on large scale linguistic resources, such as FrameNet and WordNet, is discussed. In particular, a robust method for assigning conceptual relations (i.e. roles) to relevant grammatical structures is defined according to distributional models of lexical semantics over a large scale corpus. Experimental results show that the use of the resulting knowledge base provide significant results, i.e. correct interpretations for about 90% of the covered sentences. This confirms the impact of the proposed approach on the quality and development time of large scale IE systems.

1 Introduction

Contemporary Web-based Information Extraction (IE) systems are usually integrated with large scale knowledge bases. The latter determine the semantic constraints needed for a correct interpretation of usually domain specific texts. Unfortunately, the manual construction of these resources is a time-consuming task that is often highly error-prone due to the subjectivity and intrinsic vagueness that affects the semantic modeling process.

One approach to the knowledge acquisition task is to use machine learning algorithms to automatically learn the domain-specific information from annotated data. One of the hard problems in this task is the involved complexity required to induce general patterns and rules from the individual sentences in domain texts. These come in fact in large volumes, but are characterized by high levels of ambiguity due to the involved qualitative and often vague nature of natural language. For example in the following sentences a seemingly unique event is described expressed by a verb like *condemn*:

He wrote a statement condemning the Committee 's behavior (1)

Mayor Jacques Chirac condemned,
without reservation what he called the absurd attack (2)

The message must condemn the lack of security of our systems (3)

Moreover, the syntactic structure of the sentences is similar as we can recognize a `Subj:X-`*condemn*`-Obj:Y` grammatical pattern. However, the semantic structure of the sentences is different: in sentence (1) and (3) above, `X` refers to a *content* (i.e. words

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 512–524, 2010.

sentence and *message*), while in (2) `X` makes reference to a *person* (i.e. *Chirac*). The main consequence, in linguistic terms, is that the semantic role of the same grammatical relation (i.e. `Subj`) changes as at least two ways exist to assign a role to the `Subj` relation.

In this work we adopt the *frame semantics* theory [1], it provides a general and well-founded linguistic model for which linguistic predicates and roles can be defined in terms of *frames*, that can be applied to different domains. A frame is a conceptual structure modeling a prototypical situation and evoked in a sentence through the occurrence of a set of lexical units. A *lexical unit* (LU) is a predicate (e.g. a noun or a verb) that linguistically expresses the situation of the frame. The above (1)-(3) sentences are occurrences of the JUDGMENT COMMUNICATION frame (Figure 1) introduced by the lexical unit *condemn*. Other lexical units, as verbs, nouns or adjectives are *acclaim.v, accuse.v* and *censure.n*. A noticeable contribution of a frame is the prediction of a set of prototypical semantic roles, i.e. semantic arguments, called *Frame Elements* (FE), that characterize the participants to the underlying event, irrespectively from individual lexical units. The JUDGMENT COMMUNICATION frame has 19 FEs as for example the COMMUNICATOR, the EVALUEE notion, or the MEDIUM of the judgment. Frames can be thus thought as patterns that describe conceptual primitives, highly general and domain independent through linguistic constraints given by the lexical units.

Table 1. Frame *Judgment Communication* from the FrameNet Database

Frame: JUDGMENT COMMUNICATION	
A COMMUNICATOR communicates a judgment of an EVALUEE to an ADDRESSEE.	
Frame Elements	
COMMUNICATOR	**Jon** belittled Madie to her colleagues.
EVALUEE	Jon belittled **Madie** to her colleagues.
EXPRESSOR	She viewed him with a critical **gaze**.
MEDIUM	Jon belittled Madie **over the telephone**.
REASON	Jon extolled Madie **for her efforts**.
Predicates	
acclaim.v, accuse.v, belittle.v, censure.n, censure.v, cite.v, condemn.v, critical.a, criticize.v, critique.n, damn.v, denigration.n, deride.v, extol.v, excoriate.v, ...	

In [2] Frame Semantics was used for a IE task by finding predicate argument relations in texts and map resulting structures into templates via hand-written simple rules. This approach can noticeably reduce the develop time of a IE system. The FrameNet project [3], launched during late 90's, aims at developing an extensive description of frame semantics for a large portion of modern English, and gather a large corpus of examples fully annotated through frames and related frame elements. The annotated version of sentence (1) for the lexical unit *condemn* would be: *He wrote a [statement]*$_{Medium}$ *[condemning]*$_{lu}$ *[the Committee 's behavior]*$_{Evaluee}$. The resulting semantic pattern is

`Subj:X:Medium-`*condemn*`-Obj:Y:Evaluee`

that is also valid for sentence (3). On the contrary, the annotation of sentence (2) requires the pattern:

Subj:X:Communicator-*condemn*-Obj:Y:Evaluee.

The switch of meaning is due to the semantics of the involved words: *Mayor* here refers to a person while *statement*, similarly to the semantics of *message* in sentence (3), refers to a content, i.e. an abstraction.

The major problem here is that the amount of ambiguity is not negligible: even when a specific predicate (e.g. a single verb) is targeted, a given grammatical structure still gives rise to a huge number of possible interpretations. The interpretation of a grammatical pattern (such as Subj:X-*condemn*-Obj:Y) is not straightforward as we have 19 possible roles for JUDGMENT COMMUNICATION and ambiguous meanings of words appearing in the grammatical relations with the verb (e.g. *behavior* as the Obj of *condemn*). In WordNet ([4]) *behavior* and *statement* have 4 and 7 different senses, respectively. An interpretation is obtained by selecting the proper senses and then a correct semantic role for each grammatical relation. The number of such different choices is obtained by multiplying the 171 permutations (given the 19 different roles for two grammatical relations), multiplied by the number of potential senses of both words (i.e. 7×4): this amounts to $4,788$ different interpretations, among which only one is correct. During the knowledge acquisition process, the induction of semantic and syntactic constraints from sentences like (1)-(3) should pick the valid interpretations by governing this combinatorial explosion. Although manual validation is always applied, the above proliferation decreases the productivity of the knowledge engineer that needs to analyze too many candidates.

In recent work (e.g. [5], [6]) models of corpus-driven induction of patterns have been focusing on FrameNet and WordNet as semantic systems of predicate, role and sense information. In these frameworks, the above critical problems suggest that the selection of proper semantic roles for the observed grammatical patterns requires the combination of the following evidence:

- the individual grammatical relations (such as Subj or Obj)
- the semantic type of the grammatical heads, i.e. the intended word sense *content, message* for words found in such syntactic positions, like *statement* or *message* as Subj

In [5] an approach for the induction of semantic patterns from a large corpus, such as the Web, is presented. Although it discuss a method for the acquisition of lexical patterns, it does not provide a robust solution for the assignment of semantic roles to each pattern.

Here, we will present a *distributional model* aimed at: (1) *inducing semantic role preference for individual grammatical relations involving frames*, (2) *producing complete role assignments for lexical patterns* and (3) *compile them into a knowledge base*. The method relies on measures of semantic similarity between the grammatical heads (such as *statement* and *message*) based on a vector space model ([7]): syntactic heads are modeled as vectors of their co-occurrences in the corpus, used to define semantic similarity as distances. Examples of correct role assignments are derived from the FrameNet annotated corpus. Role preferences for novel instances (in the domain corpus) are obtained as distances from the positive (i.e. FrameNet) examples. Results allow to define distributional preferences for individual roles in given grammatical relations. Moreover the application of joint preferences to full patterns, as derived from

the corpus, results in a set of constraints predicting roles and senses for all the individual relations in a pattern: these rule sets constitute the output *Pattern Knowledge Base* (PKB).

While the main learning process, previously presented in [5], is summarized in Section 2, the algorithmic details of the overall pattern acquisition are described in Section 3. In Section 4 the empirical evaluation specifically designed for measuring the semantic accuracy of the resulting knowledge base is discussed.

2 Acquiring Patterns from Corpora

The previous section showed that the acquisition of patterns is critically tied to the resolution of the semantics ambiguities characterizing occurrences of typical grammatical heads (such as *statement* or *message*) as observable in the corpus examples. The learning process discussed in [5] exploits the lexical KB provided by WordNet to limit the sense ambiguity of source examples: large sets of generalized lexical patterns such as

$$condemn\text{: (Subj : } \{message, content, ...\} \text{ : Medium) (Obj : } \{state\} \text{ : Evaluee)}$$

are produced where *message, content, ...* and *state* correspond to synsets, i.e. semantic generalizations for individual words such as *statement* found in `Subj` position or *lack* as an `Obj`.

The induction process will be here summarized. First grammatical patterns such as $lu : r_1 : x_1$, ..., $r_n : x_n$ are gathered from a corpus, where lu are frame's lexical units. Notice that a different number of grammatical structures can be found for a lu. WordNet then allows to generalize the grammatical heads x_i of each syntactic relation r_i. This results in lexical patterns such as

$$lu\text{:}(r_1 : \alpha_1), ..., (r_n : \alpha_n)$$

where α_i are generalizations (i.e. synsets). For example, for sentence (1) and (3) the following pattern can be derived

condemn: (Subj : *"message, content, ..."*) (Obj : *"state"*)

where *"message,content"* generalizes the heads *message* and *statement*, while *behavior* and *lack* are generalized in *state*.

Finally, the following pattern is derived:

condemn: (Subj:*"message,content,..."*:Medium) (Obj:*"state"*:Evaluee)

through the proper role assignment for the relation-sysnsets pairs. The implied meaning of each pattern is that sentences where the subjects activate a sense that is an hyponym of the *"message, content, ..."* synset and the grammatical objects trigger hyponyms of the *"state"* synset, then

- the involved frame is JUDGMENT COMMUNICATION
- the subject expresses the MEDIUM FE and
- the grammatical object expresses the EVALUEE FE

In [5] the above patterns are encoded according to the underlying OWL FrameNet model. Semantic constraints can be applied during other knowledge acquisition steps as well as in the Information Extraction processes. The core steps in the above sketched process are: (1) the generalization α_i of words as found in some grammatical relations r_i, and (2) the selection of the proper frame elements the (r_i, α_i) pairs. In [5], a methodology based on WordNet for solving the problem (1) is discussed. After observing a *grammatical pattern* (gp) such as $lu : r_1, \ ..., r_n$, a large number of heads, i.e. the word fillers of a relation r_i in at least one sentence, is made available from the corpus sentences. These words form a set called hereafter H_i. First, all the common hypernyms α_i of at least two words in H_i are collected from WordNet. Each α_i corresponds to a subset of H_i and it can be scored according to a WordNet based measure of the semantic similarity among its members. The measure adopted here is the conceptual density, *cd* ([8]) that is a n-ary similarity measure sensitive to the WordNet hierarchy structure as well as to the number of different words generalized by an α_i. Given the entire set H_i the subset of most useful (i.e. conceptually dense) synsets α_i able to generalize all (or most) words in H_i is thus defined. In the following Section 3 problem (2) will be discussed, i.e. the selection of the proper frame elements given a sequence of (r_i, α_i) pairs. Input to this step are the lexical patterns:

$$lp = \text{lu:}(r_1 : \alpha_1), ..., (r_n : \alpha_n)$$

Producing a compact number of correct interpretations for individual lps is challenging as the proliferation of possible role assignments increases the size of the search space of a factor between 10 to 100.

3 Learning Information Extraction Patterns

As a lexical pattern lp expresses a sequence of grammatical relation r_i and lexical sense α_i pairs, the induction of information extraction patterns is carried out as a sequence labeling task. First, a distributional approach is applied to label each sense-relation (r_i, α_i) pair with the most suitable frame element FE^{k_i}: k_i is a choice function within the dictionary of valid frame elements FE_j of the correspondent frame. As possibly multiple choices are suggested for each relation r_i, the above step is followed by a sequence labeling process, where the best role sequence is correspondingly assigned to the entire lexical pattern. The overall sequence of $FE^{k_1,...,k_n}$ gives rise to an interpretation as a single *information extraction pattern* (iep). Finally, the subset of all plausible iep is processed to select a compact KB that will be finally compiled in OWL.

In section 3.1 the geometrical model of frame semantics defined to compute preferences for individual pairs (r_i, α_i) is described. In Section 3.2 the sequence labeling technique is presented, while Section 3.3 is devoted to the patterns selection methods based on different strategies, from strictly conservative to more greedy ones.

3.1 Modeling Individual Frame Elements

Semantic spaces have been widely applied in several NLP tasks, ranging from information retrieval to paraphrase rules extraction [9]. In these models word meanings are

described by the set of textual contexts in which they occurs (*Distributional Hypothesis* [10]). Words with similar vectors are semantically related. Our goal is to leverage semantic spaces to capture similarity among words that exhibit the same role (i.e. appearing in the same frame elements). The annotated corpus provides examples of semantic heads h for the individual frame elements FE_i: they form the set hereafter referred as H^{FE_i}. Heads $h \in H^{FE_i}$ can be mapped into a space where every word is represented as a vector $\overrightarrow{h}$. The distance in the space between $\overrightarrow{h}$ and $\overrightarrow{w}$ can be thus used as a measure for the role preference of future words w with respect to FE_i.

The Semantic Word Space. The word space is built over contexts of individual words w. A context is a sentence of the corpus: each vector $\overrightarrow{s}$ expresses a sentence whose weights are $tf \cdot idf$ factors of words within sentences. The overall space is thus represented by a matrix M, whose rows describe sentences and columns correspond to terms. A dimensionality reduction technique called *Latent Semantic Analysis* is applied to the matrix M through the Singular Value Decomposition (SVD) transformation [11]. SVD produces an approximation of the original matrix M, aiming to capture semantic dependencies between source vectors, i.e. contexts. The original space is replaced by a lower k-dimensional space whose dimensions are pseudo concepts, i.e. linear combinations of the original words. These newly derived features may be thought of as artificial concepts, each one representing an emerging meaning component as a linear combination of many different words (or, dually, contexts). The SVD reduction has two main advantages. First, the overall computational cost of the model is reduced, as similarities are computed on a space with much fewer dimensions. Second, it allows to capture second-order relations among words, thus implicitly defining a meaningful similarity measure between word pairs.

We aim at modeling a specific role FE_i related to a syntactic relation r_i, through the examples h_i observable in the annotated corpus.

Examples of role fillers $h \in H^{FE}_{\texttt{Subj}}$ observed in the annotated corpus for the JUDGMENT COMMUNICATION frame are: *people, he, mayor, opponent* and *general* for the frame element COMMUNICATOR, *newspaper, term, judgment* and *page* for MEDIUM and *act, idea, her* and *government* for EVALUEE.

Notice that every $h \in H^{FE}_r$ is mapped into a vector $\overrightarrow{h}$. A vector representation for each $\overrightarrow{FE}_r$ can be thus computed as the geometric centroid of the vectors $\overrightarrow{h}$, with $h \in FE_r$. Unfortunately, such a simple approach is prone to errors due to the heterogeneous semantics of frames. Role fillers for FE_i can typically describe different situations in different contexts. For example, although the nouns *general* and *people* are potential COMMUNICATORS in a JUDGMENT COMMUNICATION events, they are likely to appear in very different linguistic contexts. In this case, vectors are likely to be very distant in the semantic space. We thus assume that different and separated regions of the semantic space are better representations for a given frame.

The vectors $\overrightarrow{h}$ for the heads $h \in FE_r$, i.e. corresponding to the same syntactic relation r for a lu, constitute the source set $H^{FE_i}_r$. Then a clustering approach is applied to $H^{FE_i}_r$ for deriving the subsets with high internal similarity. These are computed by an adaptive algorithm, based on k-means [12,13], applied to $H^{FE_i}_r$. The resulting clusters form the set $C^{FE_i}_r$. Each cluster $c \in C^{FE_i}_r$ is represented in the space by a vector $\overrightarrow{c}$,

i.e. the geometric centroid of all its members. Given a frame element FE_i for a frame F, every relation r gives rise to a set $C_r^{FE_i}$ that allows to define a geometric preference model.

A preference model for individual role assignments. Individual role assignment are needed to assign a specific frame element (e.g. MEDIUM) to a (r_i, α_i) pair of a lexical pattern lu, such as (`Sbj`,"*message, content, ...*"). As every synset α_i is a generalization within the set $H_{r,\alpha}$ (i.e. the set of grammatical heads in a syntactic relation r with a lu), it expresses a set of words. We can thus exploit again the geometrical space to evaluate the semantic tendency of the members of $H_{r,\alpha}$. For every head $h \in H_{r,\alpha}$ the vector representation $\overrightarrow{h}$ is obtained in the semantic space, and a geometric centroid of all words h is computed: this $\overrightarrow{t}_{r,\alpha}$ vector represents a pseudo word $t_{r,\alpha}$ as the linear combination of words $h \in H_{r,\alpha}$. It establishes the region of the word space where the semantics of the synset α is realized.

A model of the similarity between WordNet synsets α and frame elements FE_i is the closeness between their geometrical representations. Given an underlying syntactic relation r, the $t_{r,\alpha}$ vectors depending on the domain corpus, are compared with the sets $C_r^{FE_i}$ whose word clusters have been derived from the FrameNet examples. The similarity between the j-th word cluster c_{ij} of $C_r^{FE_i}$ and a pseudo word $t_{r,\alpha}$ is the *cosine similarity*, i.e.:

$$\forall c_{ij} \in C_r^{FE_i},\ sim_{cos}(t_{r,\alpha}, c_{ij}) = \frac{\overrightarrow{t}_{r,\alpha} \cdot \overrightarrow{c}_{ij}}{\|\overrightarrow{t}_{r,\alpha}\| \, \|\overrightarrow{c}_{ij}\|}$$

Irrelevant clusters c_{ij} are removed imposing a threshold τ, so a cluster c_{ij} is *plausible* **iff** $sim(t_{r,\alpha}, c_{ij}) \geq \tau$. The resulting set of all plausible clusters c_{ij}, i.e. representations for the semantics of the $t_{r,\alpha}$, is then ranked according to decreasing values of $sim(t_{r,\alpha}, c_{ij})$, and the k most similar clusters are selected. The resulting set will be denoted by $D_{r,\alpha}$.

The likelihood of a specific FE_i as a valid frame element for a pair $(r, \alpha) \in lp$ is obtained through majority voting within the k clusters in $D_{r,\alpha}$, i.e.:

$$pr(FE_i|\alpha, r) = \frac{|C_r^{FE_i} \cap D_{r,\alpha}|}{k} \tag{4}$$

Figure 1 shows an example of the projection (in a planar bidimensional word space) of several heads for three different frame elements of JUDGMENT COMMUNICATION: COMMUNICATOR, EVALUEE, MEDIUM. Clusters c_{ij} are shown by circles.

The lexical sense $t_{r,\alpha}$ = "*message, content, ...*", representing heads such as $message$ and $statement$, is projected in the space. When the $k = 3$ most similar clusters are selected, two of them *vote* for the FE MEDIUM and one for COMMUNICATOR.

Resulting probabilities are estimated as $pr(\text{MEDIUM}|t_{\texttt{Subj},\alpha}, \texttt{Subj}) = 2/3$, $pr(\text{COMMUNICATOR}|t_{\texttt{Subj},\alpha}, \texttt{Subj}) = 1/3$: equation 4 thus properly rejects the EVALUEE role as $pr(\text{EVALUEE}|t_{\texttt{Subj},\alpha}, \texttt{Subj}) = 0$.

3.2 Induction of Semantic Patterns

The equation 4 provides a preference model for the individual roles (r_i, α_i) in a lexical pattern $lp = lu : (r_1 : \alpha_1), \ldots, (r_n : \alpha_n)$. In the general case, more than one frame

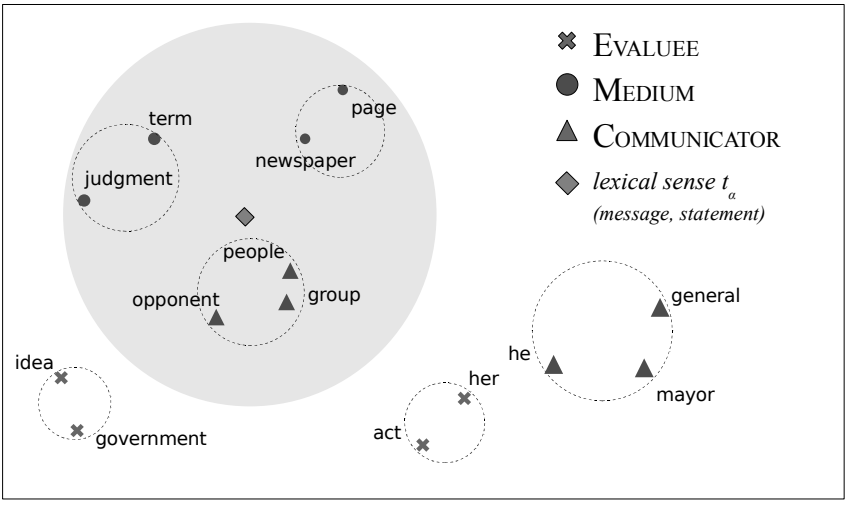


Fig. 1. Example of induction of semantic role

element FE_j gets a probability $pr(FE_j|r_i, \alpha_i) > 0$. The entire sequence $FE^{k_1, \ldots, k_n}$ is an interpretation for a lexical pattern lp if frame elements are correspondingly assigned to all the (r_i, α_i) pairs in lp through a choice function k_i within the set of valid frame elements for the targeted lu.

Notice that not all sequences are acceptable. A sequence $FE^{k_1, \ldots, k_n}$ is *valid* for the lp of a frame F *iff* $\forall(r_i, \alpha_i) \in p, FE^{k_i}$ is a frame element of F and all frame elements FE^{k_i} appear only once in the sequence.

As independence between the occurrences of frame elements can be assumed with a good approximation, the likelihood of any valid sequence of frame elements $FE^{k_1, \ldots, k_n}$ found for a corpus pattern lp is given by:

$$pr(FE^{k_1, \ldots, k_n} \mid p) = \prod_{i=1}^{n} pr(FE^{k_i} | \alpha_i, r_i) \tag{5}$$

Equation 5 makes available a ranked list of $FE^{k_1, \ldots, k_n}$ sequences for a lexical pattern lp. Every sequence $FE^{k_1, \ldots, k_n}$ defines an information extraction pattern iep induced from lp where specific role assignments and semantic type constraints are made available for every relation r_i:

$$iep = \text{lu:}\{(r_1, \alpha_1, F^{k_1}),\ \ldots, (r_n, \alpha_n, F^{k_n})\}$$

A sentence s including a lexical unit lu *is covered by* a pattern iep, when for all i:

- a syntactic relation r_i in s connects the head h_{r_i} to lu
- h_{r_i} has a WordNet sense as an hyponim of α_i

A sentence covered by iep inherits all frame element FE^{k_i} assignments, as a *fully interpreted sentence*: $s = lu : \{(r_1, h_1, F^{k_1}),\ \ldots, (r_n, h_n, F^{k_n})\}$

3.3 Compiling Patterns in the Knowledge Base

Eq. (4) and (5) allow to rank all potential information extraction patterns iep as they are induced from a lexical pattern lp. A *Pattern Knowledge Base* (*PKB*) made of all FE sequences evoked by a lp determined by a lexical unit of a frame F is the result.

Notice how a potential over-generation can arise due to the combinatory explosion of possible interpretations of lp. In order to control this undesirable effect a strategy to cut out some of the derived ieps can be defined according to two criteria:

$C1$: Eq. 5 provides probabilities over the ieps derived from one lp: distribution thresholds on the corresponding ranked iep list can be defined to prune the unlikely iep patterns of a given lp

$C2$: Some patterns can be subsumed by more general ones, when they express the same FE sequences and the involved synsets in the latter are correspondingly hyperonims of the former ones

$C1$ leads to three selection strategies:

- PKB_{best}: for every lp only the most likely $FE^{k_1}, \ldots, FE^{k_n}$ sequence is added to the PKB: $\arg\max_{FE^{k_1}, \ldots, FE^{k_n}} pr(FE^{k_1}, \ldots, FE^{k_n} | \, lp)$
- PKB_{better}: each distribution induced by Eq. 5 over a lp is characterized by a mean value μ_{lp} and a standard deviation σ_{lp}. The selected sequences are only those that exhibit a reliable probability given the lp, i.e. the sequence for which: $pr(FE^{k_1}, \ldots, FE^{k_n} | \, lp) > \mu_{lp} + \sigma_{lp}$
- PKB_{all}: no cut is imposed and all $FE^{k_1}, \ldots, FE^{k_n}$ are included in the PKB

After the generation of the resulting PKB, $C2$ suggests that subsumption can still hold between some pattern pairs. A pattern iep_1 is subsumed by a pattern iep_2 if (a) they correspond to the same frame element sequence and (b) each synset of the former is the same or in an hyponymy relation with the corresponding synset in the latter. The subsumed pattern is lexically redundant, as it is implied by some other patterns in the PKB. It will be then *pruned out* from the PKB.

4 Empirical Evaluation

In this section the quality of the acquired knowledge base will be evaluated in terms of the accuracy reachable in an IE task. As the FrameNet project makes available 135.000 annotated examples for about 800 frames, we will use these sentences as a gold-standard to evaluate the coverage and accuracy of the induced KB. Notice that we focus here on the pattern KB and not on a specific underlying IE system. Our aim is to evaluate how the pattern knowledge bases (PKB) support the interpretation of the corpus sentences, although we do not know *how to select among possibly competing interpretations*[1]. The *quality* of a PKB will be thus evaluated in terms of the amount and accuracy of the (possibly ambiguous but) correct interpretations derivable over the FrameNet sentences.

[1] This would require to train an IE system over the PKB: this is certainly possible but out of the scope of this paper.

The *correctness* of the labeling is the percentage of sentences for which the FrameNet annotation are the same as the annotation suggested by the patterns in PKB. Other aspects, such as the evaluation of other steps of the FOLIE system [14] (such as the *lu* discovery step) has been discussed elsewhere ([5]).

Experimental Setup. The corpus adopted in these experiments for acquiring grammatical patterns (as already discussed in [5]), is TREC-2002 Vol.2, made of about 110 million tokens in 230,401 documents. The semantic word space is derived from the British National Corpus [2] corpus, consisting of about 100 million words. Contexts for the description of a word are sentences from the corpus for a resulting set of about 362,000. The vector representing an individual word is derived by computing $tf \cdot idf$ scores across the corpus. The SVD reduction is run over the resulting 4,530,000 $\times$ 362,000 matrix, with a dimension cut of $k = 250$. The training data used to build the role preference of our approach (Eq. 4 in Section 3.1) are derived from the 135.000 sentences tagged FrameNet corpus [3], where lexical unit and frame elements are explicitly annotated. The TREC and the FrameNet corpus were previously processed with LTH parser [15] that provides the individual syntactic relations between lexical units and FE fillers (as grammatical heads). Syntactic relations of a lexical unit (i.e. a verb in these experiments) provide a set of words that are the individual semantic heads. Data from FrameNet are used to derive the clusters $C_r^{FE_i}$ defined in Section 3.2. Subsets of the heads observed in the TREC corpus are obtained through their generalization in WordNet 2.0. Clusters centroids are considered plausible according to $\tau = 0.1$ in all experiments. Moreover, the probability distribution across frame elements (Eq. 5) is estimated among the k most similar clusters to $\overrightarrow{t}_{r,\alpha}$, with $k = 12$.

The test sentences are a subset of the FrameNet corpus not used to gather the examples of annotated frame elements as discussed in section 3.2. The comparative evaluation affects sentences of three frames COMMERCE_BUY, KILLING and JUDGMENT COMMUNICATION, yet previously used for performance evaluation in [6]. Unfortunately we are not able to compare our evaluation since [6] provides a manual evaluation of the acquired knowledge-base. The details of the test set are reported in Table 2 : it shows the number of the test sentences (and their different FEs), as well as the different FEs acquired from the corpus and used as hypothesis in annotating test sentences. Notice that the input of the induction process is given by large sets of lexical patterns generated by FOLIE, as reported, for each frame, in row 5. All the test sentences constitute the oracle O, where individual roles and their FE tags are the gold-standard of all the measures discussed hereafter. Finally, the last row presents the amount of corpus sentences observed by FOLIE, i.e. the source evidence used to induce the lexicalized (i.e. WordNet disambiguated) patterns.

Evaluation measures. For each information extraction pattern $iep \in PKB$, the set of test sentences ($s \in O$) covered by iep inherits one corresponding role interpretation. The *coverage* of a PKB is the percentage of test sentences which receives at least one interpretation (even when this is possibly wrong).

Any $s \in O$ is *correctly interpreted* iff, for every syntactic argument, the frame element implied by iep is the same as the one defined in the oracle. The *sentence level*

[2] Notice that FrameNet annotated sentences come from this corpus.

Table 2. Experimental set-up for three frames

	COMMERCE BUY	KILLING	JUDGMENT COMMUNICAT.
# sentences	129	142	900
# of different FEs	183	162	1205
# lexical patterns	82,800	9,001	1,203,880
frame elements	BUYER - GOODS MONEY - RECIPIENT SELLER	CAUSE - INSTRUMENT KILLER - MANNER MEANS - VICTIM	ADDRESSEE COMMUNICATOR EVALUEE - GROUNDS MANNER - MEDIUM REASON TIME TOPIC
# TREC sentences	44,307	5,450	62,043

Table 3. Experimental results

(a) Evaluation of PKBs over frames

	sent_acc	*role_acc*	*overg*	PKB size	*prune*
	COMMERCE_BUY				
best	0.836	0.884	4.992	4800	96.1%
better	0.844	0.890	5.109	5508	95.9%
all	0.922	0.951	12.297	37234	95.3%
	JUDGMENT COMMUNICATION				
best	0.929	0.949	3.671	12916	90.2%
better	0.932	0.952	3.685	15380	90.2%
all	0.974	0.982	12.436	141354	92.1%
	KILLING				
best	0.777	0.833	2.050	1900	80.1%
better	0.793	0.846	2.289	2614	74.2%
all	0.967	0.983	8.231	24266	88.0%

(b) Micro-average results over the three frames

	sent_acc	*role_acc*	*overg*
best	0.847	0.889	3.571
better	0.856	0.896	3.694
all	0.954	0.972	10.988

accuracy ($sent\_acc_{PKB}$) is thus defined as the percentage of sentences in O for which at least one correct interpretation through some $iep \in PKB$ can be found. The *role level accuracy* ($role\_acc_{PKB}$) of a PKB is the percentage of frame elements of sentences in O that receives the correct FE from some patterns $iep \in PKB$.

The over-generation factor $overg_{PKB}$ of PKB is the ratio between the useful patterns, i.e. patterns iep that give rise to at least one interpretation for some $s \in O$, and the size $|O|$ of the oracle O. Smaller over-generation scores characterize more compact PKBs. The pruning factor ($prune_{PKB}$) amounts to the percentage of patterns $iep \in PKB$ that are removed because subsumed by other patterns in PKB (Section 3.1).

Results. In Table 3(a) the evaluation of the three different strategies for deploying the final pattern knowledge bases PKB are shown. They are individually reported for the three test frames: COMMERCE_BUY, JUDGMENT COMMUNICATION and KILLING. As discussed in section 3.2 three different selection strategies are used. The two models, PKB_{best} and PKB_{better} represent a conservative and more greedy automatic strategy,

respectively. PKB_{all} refers to a strategy closer to the manual validation phase of the obtained patterns: in this case, any interpretation with not null probability is preserved and presented to the knowledge engineer for validation. This latter represents obviously an upper bound with respect to the coverage and accuracy reachable, with a larger over-generation effect.

As it is shown in Table 3(a) at looser cut strategy implies a greater over-generation and accuracy as well, because many more interpretations better capture the sentence semantics: PKB_{all} represents the upper bound. Anyway, by using the PKB_{best}, that is made by the most likely interpretations for a pattern, we obtain remarkable results, in term of accuracy. It must noticed that the coverage cov_{PKB} of the three methods is the same and it is 85.6% on the test set. This seems to suggest that a not negligible number of sentences in O did not receive any interpretation.

The above measures act on individual frames, and a synthetic result can be obtained through a micro average over the original $sent\_acc_{PKB}, role\_acc_{PKB}$ and $overg_{PKB}$ scores: these are thus summarized as in Table 3(b). These results show that the most important issue for a knowledge engineer using our system is the trade-off between the expected quality and the amount of the proposed interpretations for lexical patterns lp.

5 Discussion

The presented corpus-driven methodology for an information extraction pattern acquisition is based on the integration of semantic paradigms (such as FrameNet and WordNet) and distributional models applied in an unsupervised fashion to unlabeled texts.

The presented model combines distributional information, gathered from a large scale corpus (above 250,000 documents) with a semantic paradigm (i.e. FrameNet) in a novel way. The quality of the resulting information extraction knowledge base was evaluated over a gold standard (i.e. FrameNet). The suitable interpretations of grammatical structures in the gold standard are obtained for around 85% of the sentences, with a role level accuracy of about 90%. It must noticed that about 13% of the uncovered sentences were characterized by parsing errors, that made impossible to determine the correct grammatical heads for the roles. A remaining 2% were indeed complex semantic cases characterized by metaphorical uses of the verb (such as "*to kill a song by singing it badly*") that have been correspondingly refused by our models as correct examples of the underlying frame (e.g. KILLING). We think that the estimated *semantic* coverage of our method is under estimated, and it is higher in a realistic IE scenario. The experiments confirm that our weakly supervised technique for pattern acquisition is very effective for a concrete IE task over raw text. We carried out a experimental evaluation over a semantic role labeling task, achieving an surprising accuracy if compared with state-of-art supervised systems, as those discussed [16]. The huge impact of this desirable result on the productivity of the knowledge engineering phase, implies that a rapid deployment of large scale resources is made possible. Future work will be carried out to measure the speed-up achievable through comparative studies across different systems and domains.

References

1. Fillmore, C.J.: Frames and the semantics of understanding. Quaderni di Semantica 4(2), 222–254 (1985)
2. Surdeanu, M., Surdeanu, M., Harabagiu, A., Williams, J., Aarseth, P.: Using predicate-argument structures for information extraction. In: Proceedings of ACL 2003 (2003)
3. Baker, C.F., Fillmore, C.J., Lowe, J.B.: The Berkeley FrameNet project. In: Proc. of COLING-ACL, Montreal, Canada (1998)
4. Miller, G., Beckwith, R., Fellbaum, C., Gross, D., Miller, K.: Introduction to wordnet: An on-line lexical database. International Journal of Lexicography 13(4), 235–312 (1990)
5. Basili, R., Giannone, C., De Cao, D.: Learning domain-specific framenets from texts. In: Proc. of 3rd Workshop on Ontology Learning and Population (OLP3), Greece (2008)
6. Coppola, B., Gangemi, A., Gliozzo, A., Picca, D., Presutti, V.: Frame detection over the semantic web. In: Aroyo, L., Traverso, P., Ciravegna, F., Cimiano, P., Heath, T., Hyvönen, E., Mizoguchi, R., Oren, E., Sabou, M., Simperl, E. (eds.) ESWC 2009. LNCS, vol. 5554, pp. 126–142. Springer, Heidelberg (2009)
7. Sahlgren, M.: The Word-Space Model. PhD thesis, Stockholm University (2006)
8. Agirre, E., Rigau, G.: Word sense disambiguation using conceptual density. In: Proc. of COLING 1996, Copenhagen, Denmark (1996)
9. Lin, D., Pantel, P.: DIRT-discovery of inference rules from text. In: Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD 2001), CA (2001)
10. Harris, Z.: Distributional structure. In: Katz, J.J., Fodor, J.A. (eds.) The Philosophy of Linguistics. Oxford University Press, New York (1964)
11. Landauer, T., Dumais, S.: A solution to plato's problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge. Psychological Review 104 (1997)
12. Heyer, L., Kruglyak, S., Yooseph, S.: Exploring expression data: Identification and analysis of coexpressed genes. Genome Research (9), 1106–1115 (1999)
13. Basili, R., De Cao, D., Marocco, P., Pennacchiotti, M.: Learning selectional preferences for entailment or paraphrasing rules. In: Proc. of RANLP 2007 (2007)
14. De Cao, D., Giannone, C., Basili, R.: Frame-based ontology learning for information extraction (demo paper). In: Proc. of The 16th International Conference on Knowledge Engineering and Knowledge Management Knowledge Patterns (EKAW 2008), Italy (2008)
15. Johansson, R., Nugues, P.: Semantic structure extraction using nonprojective dependency trees. In: Proceedings of SemEval 2007, Prague, Czech Republic, June 23-24 (2007)
16. Johansson, R., Nugues, P.: The effect of syntactic representation on semantic role labeling. In: Proceedings of COLING, Manchester, UK, August 18-22 (2008)

Multi-view Bootstrapping for Relation Extraction by Exploring Web Features and Linguistic Features

Yulan Yan, Haibo Li, Yutaka Matsuo, and Mitsuru Ishizuka

The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
yulan@mi.ci.i.u-tokyo.ac.jp
lihaibo@mi.ci.i.u-tokyo.ac.jp
matsuo@biz-model.t.utokyo.ac.jp
ishizuka@i.u-tokyo.ac.jp

Abstract. Binary semantic relation extraction from Wikipedia is particularly useful for various NLP and Web applications. Currently frequent pattern mining-based methods and syntactic analysis-based methods are two types of leading methods for semantic relation extraction task. With a novel view on integrating syntactic analysis on Wikipedia text with redundancy information from the Web, we propose a multi-view learning approach for bootstrapping relationships between entities with the complementary between the Web view and linguistic view. On the one hand, from the linguistic view, linguistic features are generated from linguistic parsing on Wikipedia texts by abstracting away from different surface realizations of semantic relations. On the other hand, Web features are extracted from the Web corpus to provide frequency information for relation extraction. Experimental evaluation on a relational dataset demonstrates that linguistic analysis on Wikipedia texts and Web collective information reveal different aspects of the nature of entity-related semantic relationships. It also shows that our multi-view learning method considerably boosts the performance comparing to learning with only one view of features, with the weaknesses of one view complement the strengths of the other.

1 Introduction

Recent attention to automatically harvesting semantic resources from Wikipedia has encouraged Data Mining researchers to develop algorithms for it. Many efforts have been focused on extracting semantic relations between entities, such as *birth_date* relation, *CEO* relation, and other relations.

Currently one type of the leading methods in semantic relation extraction are based on collecting relational frequency patterns or terms from a local corpus or use the Web as corpus [17]; [15]; [2]; [11]; [4]. Let us call them frequent pattern mining-based methods. The standard process is to scan or search the corpus to collect co-occurrences of word pairs with strings between them, then from collective strings calculate term co-occurrence or generate textual patterns. In order to clearly distinguish from linguistic features below, let us call them Web features. For example, given an entity pair $< x, y >$ with $Spouse$ relation, string "*x is married to y*" is a textual pattern example. The method is used widely, however, frequent pattern mining is non-trivial since

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 525–536, 2010.

the number of unique patterns is loose but many are non-discriminative and correlated. One of the main challenges and research interest for frequent pattern mining is how to abstract away from different surface realizations of semantic relations to discover discriminative patterns efficiently.

Another type of leading methods are using linguistic analysis for semantic relation extraction from well-written texts(see e.g., [14]; [5]; [22]). Let us call them syntactic analysis-based methods. Currently, syntactic analysis-based methods for semantic relation extraction are almost all supervised, relying on pre-specification of the desired relationship or hand-coding initial training data. The main process is to generate linguistic features based on the analysis of the syntactic, dependency or shallow semantic structure of text, then through training to identify entity pairs which assume a relationship and classify them into pre-defined relationships. For example, given an entity pair $< x, y >$ and the sentence "*x is the wife of y*", syntactic, dependency features will be generated by analysis of the sentence. One of the main disadvantages is that semantic relations maybe expressed in different dependency/syntactic structures. Moreover, for the heterogeneous text found on the Web, it often runs into problems to apply "deep" linguistic technology.

Syntactic analysis-based methods extract relation instances with similar linguistic features to abstract away from different surface realizations of semantic relations, while frequent pattern mining-based methods group different surface patterns for one relation instance from redundancy Web information are expected to address the data sparseness problem. Wikipedia, unlike the whole Web corpus, as an earlier report [12] explained, Wikipedia articles are much cleaner than typical Web pages, we can use "deep" linguistic technologies, such as syntactic or dependency parsing. Considering the complementary of the strengths and the weaknesses of both two views, we propose a multi-view learning approach for relation extraction from Wikipedia with view disagreement detection which can be advantageous when compared to learning with only a single view. To decide whether two relation instances share the same relationship, a common assumption in multi-view learning is that the samples from each view always belong to the same class. In realistic settings, linguistic-view and Web-view are often corrupted by noise. For example, it happens that dependency parsing for some long sentences will be erroneous. Thus we also consider filtering view corruption which is a source of view disagreement.

In this paper we present a method for performing multi-view learning by filtering view disagreement between linguistic features and Web features. We learn a classifier in a bootstrapping way for each relation type from confident trained instances with view disagreement detected by exploiting the joint view statistics.

The main contributions of this paper are as follows:

– With a novel view on integrating linguistic analysis on Wikipedia text with redundancy information from the Web, we propose a multi-view learning approach for bootstrapping relationships between entities with the complementary between the Web view and linguistic view. From the Web view, related information between entity pairs are collected from the whole Web. From linguistic view, syntactic and dependency information are generated from appropriate Wikipedia sentences.

- Different from traditional multi-view learning approaches for relation extraction task, we filter view disagreement to deal with view corruption between linguistic features and Web features, only confident instances without view disagreement are used to bootstrap learning relations.
- Our study suggests an example to bridge the gap between Web mining technology and "deep" linguistic technology for information extraction tasks. It shows how "deep" linguistic features can be combined with features from the whole Web corpus to improve the performance of information extraction tasks. And we conclude that learning with linguistic features and Web features is advantageous comparing to only one view of features.

The remainder of the paper is organized as follows. In section 2 we will consider related work of this work. In section 3 we present out our approach. In section 4 we will report on our experimental results. Finally, in section 5 we will conclude the paper.

2 Related Work

In this section, we review several past research works that are related to our work, including, frequency pattern mining-based relation extraction, syntactic analysis-based relation extraction and multi-view bootstrapping methods.

The World Wide Web is a vast resource for information. Snowball[1] introduced strategies for generating patterns and extracting tuples from plain-text documents that required only a handful of training examples from users. At each iteration of the extraction process, Snowball evaluated the quality of these patterns and tuples without human intervention, and kept only the most reliable ones for the next iteration. [15] extracted underlying relations among entities from social networks (e.g., person-person, person-location net- work). They obtained a local context in which two entities co-occur on the Web, and accumulated the context of the entity pair in different Web pages. They defined the context model as a vector of terms surrounding the entity pair. [4] proposed a relational similarity measure, using a Web search engine, to compute the similarity between semantic relations implied by two pairs of words. They represented various semantic relations that exist between a pair of words using automatically extracted lexical patterns. The extracted lexical patterns were then clustered to identify the different patterns that expressed a particular semantic relation. In this paper, motivated by the work of [15] and [4], we extract relational terms and textual pattern from Web contexts as Web view.

Currently syntactic analysis-based relation extraction approaches for semantic relation extraction are almost supervised. Many methods, such as feature-based [14]; [23], tree kernel- based ([20]; [9]) and composite kernel-based ([21]; [22], have been proposed in literature. Zhang et al. (2006)[22] presented a composition kernel to extract relations between entities with both entity kernel and a convolution parse tree kernel. As show in their paper, composition of entity features and structured features outperforms using only one kinds of features. Their work also suggests that structured syntactic information has good predication power for relation extraction and the structured syntactic information can be well captured by the tree kernel. This indicates that the flat and the structured features are complementary and the composite of features is effective

for relation extraction. Motivated by the work of (Zhang et al., 2006), we here generate entity features and tree sub-structure features as linguistic view.

Multi-view learning approaches form a class of semi-supervised learning techniques that use multiple views to effectively learn from partially labeled data. [3] introduced co-training which bootstraps a set of classifiers from high confidence labels. [8] proposed a co-boost approach that optimizes an objective that explicitly maximizes the agreement between each classifier, while [18] defined a co-regularization method that learns a multi-view classifier from partially labeled data using a view consensus-based regularization term. [7] have reported a filtering approach to handle view disagreement, and developed a model suitable for the case where the view corruption is due to a background class.

In this study, we propose a multi-view bootstrapping approach for relation extraction from linguistic and Web views. On the one hand, from the Web view, Web features are generated from the Web redundancy information to provide frequency information. On the other hand, from the linguistic view, syntactic features are generated from Wikipedia sentences by linguistic analysis to abstract information away from surface realizations of texts. Our approach bootstrap learns a classifier for each relation type from confident trained instances by applying Christoudias et al. [7]'s view disagreement detection strategy.

3 Multi-view Bootstrapping

We propose a multi-view bootstrapping approach for relation extraction from Wikipedia based on two views of features - Web features and linguistic features - with view agreement detection strategy.

3.1 Outline of the Proposed Method

The proposed method consists of three steps. In this section, we give a brief overview of each of those steps. The subsequent sections will explain the steps in detail.

Let us assume that we are given a set of entity pairs (X, Y), the task is to classify all entity pairs into several groups, each of which represent a pre-specified semantic relationship. We first query a Web search engine to find the contexts in which the two entity words co-occur, and extract Web features that express semantic relations between the entity pair. Then we select sentences containing both entity words from Wikipedia articles, generate linguistic features such as dependency sub-structures by parsing the selected sentences using a linguistic parser. Next, since there can be more than one features that express the same semantic relation, we cluster the features to identify the ones that express a particular semantic relation. Finally, we present a multi-view bootstrapping method that learns from confident instances with view disagreement detection.

The approach consists of three steps:

– **Step 1:** Feature Acquisition. For each entity pair, generates linguistic features from corresponding Wikipedia texts using linguistic analysis and extracts Web features from context information by searching the Web.

- **Step 2:** Feature Clustering. Clusters Web feature and linguistic features respectively to identify the ones that express a particular semantic relation. We cluster features to avoid computing the similarities of features during the bootstrapping.
- **Step 3:** Multi-View Bootstrapping. For each relation type, learns a classifier which initially trained from a seed set. During bootstrapping, confidently classified samples in each view are used to label instances in the other views.

3.2 Feature Acquisition

For each entity pair, we generate two kinds of features: linguistic features from Wikipedia texts through linguistic analysis and Web features by searching context information from the Web.

Web Feature Generation. Querying an entity pair using a search engine (e.g. Yahoo!), we characterize the semantic relation between the pair by leveraging the vast size of the Web. Our hypothesis is that there exist some key terms and patterns that provide clues to the relations between entity pairs. From the snippets retrieved by the search engine, we extract relational information of two kinds: ranked relational terms as keywords and surface patterns.

- Relational Terms Collection

To collect relational terms as indicators for each entity pair, we look for verbs and nouns from qualified sentences in the snippets instead of simply finding verbs. Using only verbs as relational terms might engender the loss of various important relations, e.g. noun relations "CEO", "founder" between a person and a company. Therefore, for each concept pair, a list of relational terms is collected. Then all the collected terms of all concept pairs are combined and ranked using an entropy-based algorithm which is described in [6]. With their algorithm, the importance of terms can be assessed using the entropy criterion, which is based on the assumption that a term is irrelevant if its presence obscures the separability of the dataset. After the ranking, we obtain a global ranked list of relational terms T_{all} for the whole dataset (all the entity pairs). For each entity pair, a local list of relational terms T_{ep} is sorted according to the terms' order in T_{all}. Then from the relational term list T_{ep}, a keyword t_{ep} is selected for each entity pair *ep* as the first term appearing in the term list T_{ep}. t_{ep} will be used to generate surface patterns below.

- Surface Pattern Generation

Because simply taking the entire string between two entity words captures an excess of extraneous and incoherent information, we use T_{ep} of each entity pair as a key for surface pattern generation. We classified words into Content Words (CWs) and Functional Words (FWs). From each snippet sentence, two entity words and the keyword t_{ep} is considered to be a Content Word (CW). Our idea of obtaining FWs is to look for verbs, nouns, prepositions, and coordinating conjunctions that can help make explicit the hidden relations between the target nouns.

Table 1. Surface pattern samples for an entity pair

Pattern	Pattern
ep ceo *es*	*es* found *ep*
ceo *es* found *ep*	*es* succeed as ceo of *ep*
es be ceo of *ep*	*ep* ceo of *es*
ep assign *es* as ceo	*ep* found by ceo *es*
ceo of *ep es*	*ep* found in by *es*

Surface patterns have the following general form.

$$[\textbf{CW1}]\ Infix_1\ [\textbf{CW2}]\ Infix_2\ [\textbf{CW3}] \tag{1}$$

Therein, $Infix_1$ and $Infix_2$ respectively contain only and any number of FWs. A pattern example is "*ep* assign *ep* as *ceo (keyword)*". All generated patterns are sorted by their frequency, and all occurrences of the principle entity and the second entity are replaced with "*ep*" and "*es*", respectively for pattern matching of different entity pairs.

Table 1 presents examples of surface patterns for a sample entity pair. Pattern windows are bounded by CWs to obtain patterns more precisely because 1) if we use only the string between two entity words, it may not contain some important relational information, such as "ceo *ep* resign *es*" in Table 1; 2) if we generate patterns by setting a windows surrounding two entity words, the number of unique patterns is often exponential.

Linguistic Feature Extraction. We select sentences from Wikipedia articles containing both entities. We define the composite feature vector with flat and the structured features generated from these sentences by using a syntactic parser.

– Flat Features

Using a syntactic parser (Connexor[1]), rich linguistic tags can be extracted as features for each entity in an entity pair. For each pair of entities, we extract the following syntactic features as flat features:

– Morphology Features: tells the details of word forms used in text. Connexor Parser defines 70 morphology tags such as *N*(noun), *NUM* (numeral) .
– Syntax Features: describes both surface syntactic and syntactic function information of words. For example, *%NH* (nominal head) and *%>N* (determiner or premodifier of a nominal) are surface syntactic tags, *@SUB* (Subject) and *@F-SUBJ* (Formal subject) are syntactic function tags.

– Structure Features

To obtain structured features for an entity pair, we generate dependency patterns. After preprocessing, selected sentences that contain at least one mention of both entity words are parsed into dependency structures. We define dependency patterns as subpaths of the shortest dependency path between an entity pair for two reasons. One is

[1] www.connexor.com

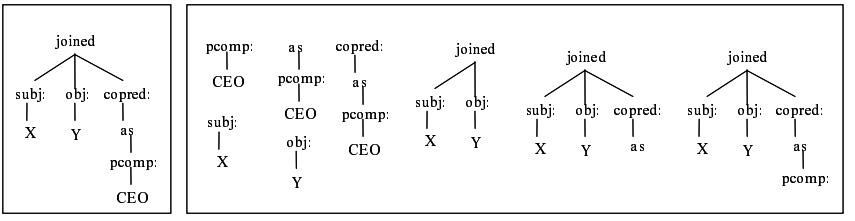


Fig. 1. Example showing how to generate dependency patterns for an entity pair

that the shortest path dependency kernels outperform dependency tree kernels by offering a highly condensed representation of the information needed to assess their relation [5]. The other reason is that embedded structures of the linguistic representation are important for obtaining good coverage of the pattern acquisition, as explained in [9]; [22]. The process of inducing dependency patterns has two steps, as shown in Fig. 1.

1. Shortest dependency path inducement. From the original dependency tree structure by parsing the selected sentence for each entity pair, we first induce the shortest dependency path from the Wikipedia sentence with the pair of entity words, as shown in the left side of Fig. 1.

2. Dependency pattern generation. We use a frequent tree-mining algorithm [19] to generate sub-paths as dependency patterns from the shortest dependency path, as shown in the right side of Fig. 1.

3.3 Feature Clustering

A semantic relation can be expressed using more than one pattern. When we compute the relational similarity between two entity pairs, it is important to know whether there is any correspondence between the sets of patterns extracted for each entity pair. If there are many related patterns between two entity pairs, we can expect a high relational similarity. To find semantically related lexical patterns for each view, we apply Sequential pattern clustering algorithm in [4] by using distributional hypothesis [13]. Distributional hypothesis claims that words that occur in the same context have similar meanings.

Given a set P of patterns and a clustering similarity threshold , their algorithm returns clusters (of patterns) that express similar semantic relations. First, their algorithm sorts the patterns into descending order of their total occurrences in all word pairs. Next, it repeatedly takes a pattern p_i from the ordered set P, finds the cluster that is most similar to p_i. To compute the similarity between a pattern and a cluster, first they represent a cluster by the vector sum of all entity pair frequency vectors corresponding to the patterns that belong to that cluster. Next, they compute the cosine of the angle between the vector that represents the cluster (c_j), and the word-pair frequency vector of the pattern (p_i). The sequential nature of their algorithm avoids pairwise comparisons among all patterns. Moreover, sorting the patterns by their total word-pair frequency prior to clustering ensures that the final set of clusters contains the most common relations in the data-set.

3.4 Multi-view Bootstrapping with View Disagreement Detection

In this section we present a multi-view bootstrapping algorithm that uses the idea of view disagreement detection. We apply (Christoudias, et al., 2008)[7]s conditional view entropy measure to detect and filter entity pairs with view disagreement in a pre-filtering step.

Multi-view learning can be advantageous when compared to learning with only a single view especially when the weaknesses of one view complement the strengths of the other. A common assumption in multi-view learning is that the samples from each view always belong to the same class. In realistic settings, datasets are often corrupted by noise. Thus we need to consider view disagreement caused by view corruption. We apply the method in (Christoudias, et al., 2008)[7] for Multi-view Bootstrapping by learning a classifier in one view from the labels provided by a classifier from another view with a view disagreement strategy. Their Method consists of two steps:

- **Step 1:** View disagreement detection. Detect and filter entity pairs with view disagreement using an information theoretic measure based on conditional view entropy.
- **Step 2:** Multi-view Bootstrapping. Mutually train a set of classifiers, on an unlabeled dataset by iteratively evaluating each classifier and re-training from confidently classified entity pairs.

Firstly, to detect view disagreement, they use conditional entropy $H(x|y)$ which is a measure of the uncertainty in x given that we have observed y. In the multi-view setting, the conditional entropy between views, $H(x_i|x_j)$, can be used as a measure of agreement that indicates whether the views of a sample belong to the same class or event. Under the assumptions the conditional view entropy is expected to be larger when conditioning on entity pairs with disagreement compared to those without disagreement. When computing the conditional entropy between views, we use the pattern clusters to replace features when measuring the conditional entropy between views so we can avoid computing the distance between two similar patterns.

Secondly, with the conditional entropy measure, we mutually train a set of classifiers for each relation type, on an unlabeled dataset iteratively evaluating each classifier and re-training from confidently classified samples. In the presence of view disagreement, we detect classified samples which are not in view disagreement. Only those detected classified samples are used to train classifiers iteratively. During bootstrapping, confidently classified samples in each view are used to label corresponding samples in the other views.

4 Experiments

In this section, we evaluate our multi-view bootstrapping approach on the relation extraction from Wikipedia, and show the effectiveness of the proposed approach.

4.1 Experimental Setup

We conduct our experiments on relation extraction task using the dataset that was created for evaluating relation extraction from Wikipedia in [10]. This data contains

Table 2. Overview of the dataset

relation	#of Instances	Instance samples for each relation type
job_title	216	(Charles Darwin, naturalist), (Jack Kerouac, novelist)
birth_year	157	(Hillary Clinton, 1947), (George H. W. Bush, 1924)
education	106	(James Bowdoin, Harvard), (Franklin Schaffner, Columbia University)
death_year	104	(Abraham Lincoln,1865), (James Bowdoin, 1790)

Wikipedia pages for which links between pages have been annotated with a relation type, e.g. birth_year, education, nationality, etc. We evaluate on a subset which consists of four relation types $job\_title, birth\_year, education, death\_year$. For each relation type, in Table 2, we show some of the instances and the total number of entity pairs. Each entity pair in the dataset has one accompanying sentence from a Wikipedia article.

We build three baseline systems on the dataset. One baseline system is built by semi-supervised learning from only the linguistic view which shows the performance of learning with only linguistic features. Another system is built by learning from only the Web view which shows the performance of learning with Web features. We also evaluate on bootstrap learning from the linguistic view and Web view without view disagreement detection in a traditional way.

To evaluate the performance of our approach, we run the feature generation algorithm described in section 3.2 for each entity pair in our dataset to extract Web features and linguistic features. We collect Web features through querying with each pair of entity words by a search engine (We use Yahoo, the top 1000 snippets are downloaded as collective context). We collect relational terms and textual patterns as Web features by look for verbs, nouns, prepositions, and coordinating conjunctions that can help make explicit the hidden relations between the target nouns. To collect linguistic features, for each entity pair, the accompanying sentence is parsed by a linguistic parser. We collect entity features for each entity word and generate dependency patterns as sub-paths from the shortest dependency path containing two entities by making use of a frequent tree-mining algorithm [19].

In these experiments, we use precision, recall, and F-value to measure the performance of different methods. The following quantities are considered to compute precision, recall, and F-value:

- p = the number of detected entity pairs.
- p' = the number of detected entity pairs which are actual relation instances.
- n = the number of actual relation instances.

$$\text{Precision } (P) = \text{p'/p} \qquad \text{Recall } (R) = \text{p'/n}$$
$$F\text{-value } (F) = 2 * P * R/(P + R)$$

4.2 Feature Clusters

We use the clustering algorithm described in Section 3.3 to cluster the extracted Web features and linguistic features respectively.

For each relation cluster in Table 3, we show top four Web features that occur with the largest frequency. From Table 3, it is clear that each cluster contains different Web

Table 3. Examples of frequent Web features from Web feature clustering

ep was a es	ep was elected es	ep was the es	ep was the leading es
ep was born in es	ep born in es	ep born D es	ep was born on es
es graduate ep	ep graduated from es	ep is a graduate of es	ep graduated from the es
ep died es	ep died in D es	ep who died in es	ep who died in D es

Table 4. Examples of frequent features from linguistic feature clustering

(be(ep))	(be(es))	(mainroot(be(es)))	(be(ep)(es))
(bear(die))	(bear(be)(die))	(mainroot(bear(die)))	(bear(be(ep))(die))
(graduate(ep))	(mainroot(graduate(ep)))	(mainroot(graduate))	(graduate(ep)(from))
(attend(ep))	(attend(ep)(es))	(mainroot(attend(ep)(es)))	(mainroot(attend(ep)))
(bear(es))	(bear(be)(in))	(bear(be(ep)))	(bear(in))

features that express a specific semantic relation. ep and es in feature expressions are used to label the first entity and second entity of a relation instance respectively. Similarly, in Table 4, for each relation cluster, we show the top four linguistic features that occur with the largest frequency. We see that linguistic features in different surface expressions are clustered to represent the same semantic relation. Moreover, each cluster contains different linguistic features that express a specific semantic relation. Each linguistic feature denotes one tree transaction represented in strict S-expression. Strict means that all nodes, even leaf nodes, must be bracketed.

4.3 Empirical Analysis

Table 5 presents the overall evaluation of the comparison of our approach and three baseline systems. The first two columns of results show learning with only one view of features respectively: linguistic view, Web view. It shows that the performance of using Web features is better than using linguistic features. Moreover, by applying traditional bootstrapping method with Web features and linguistic features without view disagreement detection, the performance is even better. It means Web features and linguistic features provide different information for the relation extraction task. The final column shows using multi-view bootstrapping approach with view disagreement detection, the performance is improved over traditional bootstrapping approach. It means that by dealing with view corruption, relations can be learned with better reliability from confident samples.

We also compared the above three baseline systems with our proposed method for the four relation types, shown in Table 6. Using only linguistic features, the performance is much worse than Web views for some relationships, such as "birth_year". A closer look into the features extracted for some entity pairs reveals that some instances which belong to different relation types are often described in the same Wikipedia sentence. This kind of sentences are often hard to be parsed in an appropriate way to generate the correct linguistic features. For Example, "*Aldous Leonard Huxley (July 26, [[1894]] C November 22, [[1963]]) was a British [[writer]]*" is the Wikipedia sentence containing instances of relations $job\_title, birth\_year, death\_year$.

Table 5. Overall evaluation over different methods

	Single-View Learning		Multi-View Bootstrapping	
	Linguistic Feature	Web Feature	Traditional	Proposed
Pre	46.30	51.80	54.14	68.19
Rec	40.82	47.00	51.03	63.95
F1	43.39	49.28	52.54	66.00

Table 6. Evaluation on each relation type over different methods

Relation	Single-View Learning						Multi-view Bootstrapping					
	Linguistic-View			Web-View			Traditional			Proposed		
	Pre.	Rec.	F-v.	Pre.	Rec.	F-v.	Pre.	Rec.	F-v.	Pre.	Rec.	F-v.
job_title	69.82	54.63	61.30	66.20	21.76	32.75	69.75	52.31	59.79	91.18	57.41	70.45
birth_year	21.43	15.29	17.84	40.00	53.50	45.78	43.38	37.58	40.27	57.71	64.33	60.84
education	56.52	12.26	20.16	52.63	47.17	49.75	42.39	36.79	39.40	69.57	60.38	64.65
death_year	39.52	79.81	52.87	60.78	89.42	72.37	48.19	89.42	62.63	53.33	92.31	67.61
overall	46.30	40.82	43.39	51.80	47.00	49.28	54.14	51.03	52.54	68.19	63.95	66.00

All the experimental results support our idea mainly in three main ways: 1) the combination of Web features and linguistic features is effective in relation extraction task; 2) It has been shown that multi-view bootstrapping is advantageous to learning with only a single view when the weaknesses of one view complement the strengths of the other. 3) the detection and filtering of view disagreement considerably increases the performance of traditional multi-view learning approaches.

5 Conclusions

We propose a multi-view learning approach for bootstrapping relationships between entities from Wikipedia with the complementary between the Web view and linguistic view. From Web view, related information for entity pairs are collected from the whole Web. From linguistic Web, analysis information from sentences are generated from Wikipedia sentences. We filter view disagreement to deal with view corruption between linguistic features and Web features, with only confident trained instances used for classifiers. Experimental evaluation on a relational dataset demonstrates that linguistic analysis and Web collective information reveal different aspects of the nature of entity-related semantic relationships. Our multi-view learning method considerably boosts the performance comparing to learning with only one view, with the weaknesses of one view complement the strengths of the other. This study suggests an example to bridge the gap between Web mining technology and "deep" linguistic technology for information extraction tasks.

References

1. Agichtein, E., Gravano, L.: Snowball: Extracting relations from large plain-text collections. In: Proceedings of the Fifth ACM International Conference on Digital Libraries (2000)
2. Banko, M., Cafarella, M.J., Soderland, S., Broadhead, M., Etzioni, O.: Open information extraction from the Web. In: Proceedings of IJCAI 2007 (2007)

3. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proceedings of COLT 1998 (1998)
4. Bollegala, D., Matsuo, Y., Ishizuka, M.: An Integrated Approach to Measuring the Similarity between Implicit Semantic Relations from the Web. In: Proceedings of WWW 2009 (2009)
5. Bunescu, R., Mooney, R.: A shortest path dependency kernel for relation extraction. In: Proceedings of HLT/EMLNP 2005 (2005)
6. Chen, J., Ji, D., Tan, C.L., Niu, Z.: Unsupervised Feature Selection for Relation Extraction. In: Proceedings of IJCNLP 2005 (2005)
7. Christoudias, C., Urtasun, R., Darrell, T.: Multi-view learning in the presence of view disagreement. In: Proceedings of UAI 2008 (2008)
8. Collins, M., Singer, Y.: Unsupervised models for named entity classification. In: Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (1999)
9. Culotta, A., Sorensen, J.: Dependency tree kernels for relation extraction. In: Proceedings of the ACL 2004 (2004)
10. Culotta, A., McCallum, A., Betz, J.: Integrating probabilistic extraction models and data mining to discover relations and patterns in tex. In: Proceedings of the HLT-NAACL 2006 (2006)
11. Davidov, D., Rappoport, A.: Classification of Semantic Relationships between Nominals Using Pattern Clusters. In: Proceedings of ACL 2008 (2008)
12. Giles, J.: Internet encyclopaedias go head to head. Nature 438, 900–901 (2005)
13. Harris, Z.: Distributional structure. Word 10, 146–162 (1954)
14. Kambhatla, N.: Combining lexical, syntactic and semantic features with maximum entropy models. In: Proceedings of ACL 2004 (2004)
15. Mori, J., Tsujishita, T., Matsuo, Y., Ishizuka, M.: Extracting Relations in Social Networks from the Web using Similarity between Collective Contexts. In: Proceedings of ISWC 2006 (2006)
16. Nigam, K., Ghani, R.: Analyzing the effectiveness and applicability of cotraining. In: Workshop on Information and Knowledge Management (2000)
17. Pantel, P., Pennacchiotti, M.: Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In: Proceedings of ACL 2006 (2006)
18. Sindhwani, V., Niyogi, P., Belkin, M.: A coregularization approach to semi-supervised learning with multiple views. In: Proceedings of ICML 2005 (2006)
19. Zaki, M.: Efficiently mining frequent trees in a forest. In: Proceedings of KDD 2002 (2002)
20. Zelenko, D., Aone, C., Richardella, A.: Kernel Methods for Relation Extraction. Journal of Machine Learning Research 3, 1083–1106 (2003)
21. Zhao, S., Grishman, R.: Extracting relations with integrated information using kernel methods. In: Proceedings of ACL 2005 (2005)
22. Zhang, M., Zhang, J., Su, J., Zhou, G.: A Composite Kernel to Extract Relations between Entities with both Flat and Structured Features. In: Proceedings of ACL 2006 (2006)
23. Zhou, G., Su, J., Zhang, J., Zhang, M.: Exploring various knowledge in relation extraction. In: Proceedings of ACL 2005 (2005)

Sequential Patterns to Discover and Characterise Biological Relations

Peggy Cellier[1], Thierry Charnois[1], and Marc Plantevit[2]

[1] Université de Caen, CNRS
Université de Caen, GREYC, UMR6072, F-14032, France
firstname.lastname@info.unicaen.fr
[2] Université de Lyon, CNRS
Université de Lyon 1, LIRIS, UMR5205, F-69622, France
marc.plantevit@liris.cnrs.fr

Abstract. In this paper, we present a method to automatically detect and characterise interactions between genes in biomedical literature. Our approach is based on a combination of data mining techniques: frequent sequential patterns filtered by linguistic constraints and recursive mining. Unlike most Natural Language Processing (NLP) approaches, our approach does not use syntactic parsing to learn and apply linguistic rules. It does not require any resource except the training corpus to learn patterns.

The process is in two steps. First, frequent sequential patterns are extracted from the training corpus. Second, after validation of those patterns, they are applied on the application corpus to detect and characterise new interactions. An advantage of our method is that interactions can be enhanced with modalities and biological information.

We use two corpora containing only sentences with gene interactions as training corpus. Another corpus from PubMed abstracts is used as application corpus. We conduct an evaluation that shows that the precision of our approach is good and the recall correct for both targets: interaction detection and interaction characterisation.

1 Introduction

Literature on biology and medicine represents a huge amount of knowledge: more than 19 million publications are currently listed in PubMed repository[1]. A critical challenge is then to extract relevant and useful knowledge dispersed in such collections. Natural Language Processing (NLP), in particular Information Extraction (IE), and Machine Learning (ML) approaches have been widely applied to extract specific knowledge, for example biological relations. The need for linguistic resources (grammars or linguistic rules) is a common feature of the IE methods. That kind of approach applies rules such as regular expressions for surface searching [5] or syntactic patterns [14, 4]. However rules are handcrafted, those methods are thus time consuming and very often devoted to a

[1] http://www.ncbi.nlm.nih.gov/pubmed/

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 537–548, 2010.

specific corpus. In contrast, machine learning based methods, for example support vector machines or conditional random fields [8], are less time consuming than NLP methods. They give good results, but they need many features and their outcomes are not really understandable by a user and not usable in NLP systems as linguistic patterns. A good trade-off is the cross-fertilization of IE and ML techniques which aims at automatically learning the linguistic rules [10, 17]. However in most cases the learning process is done from the syntactic parsing of the text. Therefore, the quality of the learned rules relies on syntactic process results. Some works such as [6] do not use syntactic parsing and learn surface patterns using sequence alignment of sentences to derive "motifs". That method allows only interaction patterns to be learned and no new terms to be discovered. Indeed, it is based on a list of terms that represent interactions. In contrast, our proposed approach automatically discovers patterns of interactions and their characterisations (e.g., kind of interaction, modality). In particular, terms representing interactions (and characterisations) are automatically extracted from texts without other knowledge. From our best knowledge, there is no method that in the same time extract interactions and their characterisations.

In this paper, we aim at showing the benefit of using data mining methods [1] for Biological Natural Language Processing (BioNLP). Data mining allows implicit, previously unknown, and potentially useful information to be extracted from data [3]. We present an approach based on frequent sequential patterns [1], a well-known data mining technique, to automatically discover linguistic rules. The sequential pattern is a paradigm more powerful than n-grams. Indeed, n-gram can be seen as a specific instance of sequential pattern. A drawback of n-grams is that the size of the extracted patterns is set for all patterns to n whereas in sequential pattern mining, discovered patterns can have different sizes. In addition, items (i.e. words of texts) within sequential patterns are not necessarily contiguous. Unlike most NLP approaches, the proposed method does not require syntactic parsing to learn linguistic rules and to apply them. In addition, no resources are needed except the training corpus. Moreover, rules coming from sequential patterns are understandable and manageable by an expert.

The process proposed in this paper is in two steps. First, frequent sequential patterns are automatically extracted from the training corpus. In addition, constraints and recursive mining [2] are used to give prominence to the most significant patterns and to filter the specific ones. The goal is to retain frequent sequential patterns which convey linguistic regularities (e.g., entity named relations). Second, after a selection and categorisation of those patterns by an expert, they are applied on the application corpus. The approach is used for the detection of new gene interactions in biomedical literature. An advantage of our method is that interactions can be enhanced with modalities and biological information. Note that no knowledge except the training corpus is used. In addition, in the training corpus the interactions are not annotated.

The paper is organized as follow. Section 2 presents the approach to compute the linguistic rules that allow gene interactions to be extracted and characterised. Section 3 gives an evaluation of our method on biomedical papers from PubMed.

2 Sequential Patterns for Information Extraction

In this section, we introduce sequential patterns defined by Agrawal *et al.* [1]. We explain how we use sequential patterns to extract potential linguistic extraction rules to discover interactions and to identify modalities and biological situations. Linguistic constraints and recursive mining are then presented to reduce the number of extracted patterns. Finally, we show the selection and categorisation of extracted sequential patterns.

2.1 Sequential Patterns

Sequential pattern mining is a well-known technique introduced by Agrawal *et al.* [1] that finds regularities in sequence data. There exist a lot of algorithms that efficiently compute frequent sequential patterns [18, 13, 20].

A *sequence* is an ordered list of literals called *items*, denoted by $\langle i_1 \dots i_m \rangle$ where $i_1 \dots i_m$ are items. A sequence $S_1 = \langle i_1 \dots i_n \rangle$ is *included* in a sequence $S_2 = \langle i'_1 \dots i'_m \rangle$ if there exist integers $1 \leq j_1 < ... < j_n \leq m$ such that $i_1 = i'_{j_1}$, ..., $i_n = i'_{j_n}$. S_1 is called a *subsequence* of S_2. S_2 is called a *supersequence* of S_1. It is denoted by $S_1 \preceq S_2$. For example the sequence $\langle a\ b\ c\ d \rangle$ is a supersequence of $\langle b\ d \rangle$: $\langle b\ d \rangle \preceq \langle a\ b\ c\ d \rangle$.

Table 1. SDB_1, a sequence database

Sequence ID	Sequence
1	$\langle a\ b\ c\ d \rangle$
2	$\langle b\ d\ e \rangle$
3	$\langle a\ c\ d\ e \rangle$
4	$\langle a\ d\ c\ b \rangle$

A sequence database SDB is a set of tuples (sid, S), where sid is a sequence identifier and S a sequence. Table 1 gives an example of database, SDB_1, that contains four sequences. A tuple (sid, S) *contains* a sequence S_α, if S_α is a subsequence of S. The *support*[2] of a sequence S_α in a sequence database SDB is the number of tuples in the database containing S_α: $sup(S_\alpha) = |\{(sid, S) \in SDB \mid (S_\alpha \preceq S)\}|$ where $|A|$ represents the cardinality of A. For example, in SDB_1 $sup(\langle b\ d \rangle) = 2$. Indeed, sequences 1 and 2 contain $\langle b\ d \rangle$. A frequent *sequential pattern* is a sequence such that its support is greater or equal to the support threshold: *minsup*. The sequential pattern mining extracts all those regularities which appear in the sequence database.

2.2 Extraction of Sequential Patterns in Texts

For the extraction of sequential patterns from biological texts, we use a training corpus which is a set of sentences that contain interactions and where the

[2] Sometimes the relative support is used:
$sup(S_\alpha) = \frac{|\{(sid, S) \mid (sid, S) \in SDB \wedge (S_\alpha \preceq S)\}|}{|SDB|}$.

Table 2. Excerpt of the sequence database

Sequence ID	Sequence
...	...
S1	⟨ *here@rb we@pp show@vvp that@in/that AGENE@np ,@, in@in synergy@nn with@in AGENE@np ,@, strongly@rb activate@vvz AGENE@np expression@nn in@in transfection@nn assay@nns .@sent* ⟩
S2	⟨ *the@dt AGENE@np -@: AGENE@np interaction@nn be@vbd confirm@vvn in@in vitro@nn and@cc in@in vivo@rb .@sent* ⟩
...	...

genes are identified. In this paper we consider sentences containing interactions and at least two gene names to avoid problems introduced by the anaphoric structures [21].

From those sentences, sequential patterns representing gene interactions are extracted. The items are the combination of the lemma and their POS tag. The sequences of the database are the interaction sentences where each word is replaced by the corresponding item. The order relation between items in a sequence is the order of words within the sentence. For example, let us consider two sentences that contain gene interactions:

– " *Here we show that <Gene SOX10>, in synergy with <Gene PAX3>, strongly activates <Gene MITF > expression in transfection assays.*"
– " *The <Gene Menin>-<Gene JunD> interaction was confirmed in vitro and in vivo.*"

The gene names are replaced by a specific item, *AGENE@np*, and the other words are replaced by the combinations of the lemmas and their POS tag. An excerpt of the database that contains the sequences associated to those two sentences is given Table 2.

The choice of the support threshold *minsup* is a well-known problem in data mining. With a high *minsup*, only few very general patterns can be extracted. With a low *minsup*, a lot of patterns can be found. In our application, some interesting words, for example "interaction", are not very frequent so that we set a low value of *minsup*. As a consequence, a huge set of patterns is discovered and it needs to be filtered in order to return only interesting and relevant patterns.

2.3 Constraints and Recursive Mining

We use a combination of data mining methods which are well-known to select the most interesting and promising patterns [12, 2]. The constraint-based pattern paradigm enables one to discover patterns under user-defined constraints in order to drive the mining process towards the user objectives. Recursive mining gives prominence to the most significant patterns and filters the specific ones.

Linguistic Constraints. In data mining, the constraints allow the user to define more precisely what should be considered as interesting. Thus, the most

commonly used constraint is the constraint of frequency (*minsup*). However, it is possible to use different constraints instead of the frequency [11]. We use three constraints on sequential patterns to mine gene interactions.

The first constraint is that the pattern must contain two named entities (C_{2ne}). $SAT(C_{2ne})$ represents the set of patterns that satisfy C_{2ne}:
$SAT(C_{2ne}) = \{S = \langle i_1 i_2 \dots i_m \rangle \mid |\{j \ s.t. \ i_j = AGENE@np\}| \geq 2\}$.

The second constraint is that the pattern must contain a verb or a noun (C_{vn}). $SAT(C_{vn})$ represents the set of patterns that satisfy C_{vn}:
$SAT(C_{vn}) = \{S = \langle i_1 i_2 \dots i_m \rangle \mid \exists \ i_j, \ verb(i_j) \ or \ noun(i_j)\}$ where $verb(i)$ (resp. $noun(i)$) is a predicate that returns true if i is a verb (resp. noun).

The last constraint is that the pattern must be *maximal* ($\mathcal{C}_{max}$). A frequent sequential pattern, S_1, is maximal if there is no other frequent sequential pattern, S_2, such that $S_1 \preceq S_2$. $SAT(\mathcal{C}_{max})$ represents the set of patterns that satisfy C_{max}:
$SAT(\mathcal{C}_{max}) = \{s \mid support(s) \geq minsup \wedge \nexists s' \ s.t. \ support(s') \geq minsup, \ s \preceq s'\}$. That last constraint allows the redundancy between patterns to be reduced.

All contrainsts can be grouped in only one constraint $\mathcal{C}_G$ which is a conjunction of previously presented constraints. $SAT(C_G)$ is the set of patterns satisfying C_G.

Recursive Mining. Even if the new set of sequential patterns, $SAT(\mathcal{C}_G)$, is significantly smaller, it can still be too large to be analysed and validated by a human user. Therefore we use *recursive mining* [2] to give prominence to the most significant patterns and to filter the specific ones.

The key idea of recursive pattern mining [2] is to reduce the output by successively repeating the mining process in order to preserve the most significant patterns. More precisely, for each step, the previous result is considered as the new dataset. That recursive process is ended when the result becomes stable.

We divide $SAT(\mathcal{C}_G)$ into several subsets E_{X_i} where the subset E_{X_i} is the set of all sequential patterns of $SAT(\mathcal{C}_G)$ containing the item X_i. More formally, $E_{X_i} = \{s \in SAT(\mathcal{C}_G) \mid \langle X_i \rangle \preceq s\}$. Note that X_i are elements labeled as a verb or a noun. Indeed, we want to identify at least one pattern by verb or noun that appears in the sequential patterns. All verbs and nouns are thus used.

The most k ($k > 1$) representative elements for each E_{X_i} are then computed. Each subset E_{X_i} is recursively mined with *minsup* equals to $\frac{1}{k}$ in order to extract frequent sequential patterns satisfying $\mathcal{C}_G$ previously introduced. The recursion stops[3] when the number of extracted sequential patterns satisfying $\mathcal{C}_G$ is less than or equal to k. It means that the extracted sequential patterns become the sequences of the new database to mine. This process ends when the number of extracted patterns is less than or equal to k. For each subset E_{X_i}, the k extracted sequential patterns are frequent sequential patterns in the first database with respect to *minsup*.

[3] The constraint $\mathcal{C}_{max}$ ensures ending of recursion.

At the end of that step, the number of sequential patterns is controlled. It is less than or equal to $n \times k$ where n is the number of subsets E_{X_i} in $SAT(\mathcal{C}_G)$. Note that k is set *a priori* by the user. Thus, the number of sequential patterns allows them to be analysed by a human user. The sequential patterns are then validated by the user and considered as linguistic information extraction rules for the detection of interactions between genes and their modalities or biological situation. Moreover, it is interesting to note that the subcategorisation of the verb given by the POS tagging indicates the passive or active verb and identifies the direction of the interaction. Prepositions can also allow that kind of information to be found when the pattern does not contain a verb.

2.4 Selection and Categorisation of Patterns

After the extraction of sequential patterns, a human user analyses them as information extraction rules. Some extracted patterns, which are not relevant for interaction detection or characterisation, are removed. The other patterns are selected as information extraction rules. A selected pattern is classified with respect to the kind on information that can be extracted with that pattern. Figure 1 shows the taxonomy that we define and use in our experiments with biological texts. That taxonomy is defined by observation of the extracted patterns. It can be completed with other classes if other kinds of information extraction rules are found. There are three main classes of patterns.

The first class is *interaction patterns* that allows interactions between genes to be found.

The second class is *modality patterns* that allows modalities of interactions to be found. Modalities induce the confidence in the detected interactions. For example, the sentence "It suggests that <gene_name=MYC> interacts with <gene_name=STAT3>." has a lower confidence than "It was demonstrated that <gene_name=MYC> interacts with <gene_name=STAT3>.". We define four levels of confidence: *Assumption*, *Observation*, *Demonstration* and *Related work*, and another subclass representing the *Negation*. A negation modality pattern is for example "AGENE@np absence AGENE@nn".

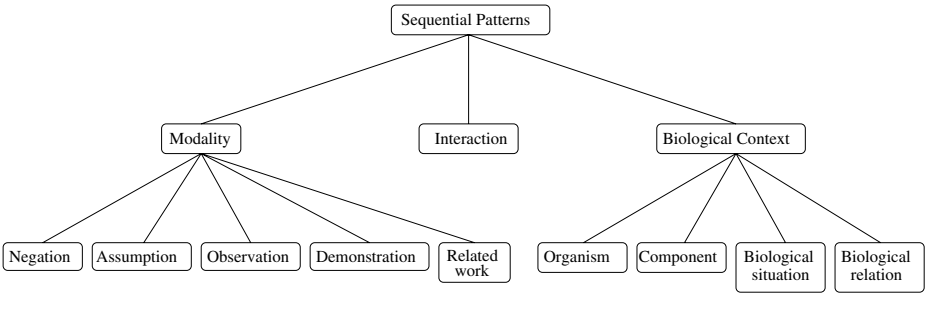


Fig. 1. Taxonomy for pattern selection

The last class is *biological context patterns* that allow information about the biological context of interactions, for example the disease or the organism involved in the interaction, to be found. That class has four subclasses: *organism*, *component*, *biological situation* and *biological relation*. The subclass *organism* enables the organisms involved in the interaction to be found, for example "mouse" or "human". The subclass *component* enables the biological components (e.g. "breast" or "fibroblast") to be detected. The subclass *biological situation* enables to give the framework of interactions, for example, "cancer", "tumor" or "in vitro". The last subclass enables to give the type of biological relation when it is possible, for example "homology".

When the human user has selected and classified all patterns in the different categories, they are applied as extraction rules on the application corpus to discover and characterise new interactions. Note that detection with sequential patterns representing interaction, modalities or biological context is much more elaborated than just a cooccurence detection. Indeed, the order of the words and the context are important, for example $\langle$*these@dt suggest@vvp AGENE@np AGENE@np*$\rangle$ or $\langle$*AGENE@np with@in AGENE@np in@in Vitro@np .@sent*$\rangle$.

3 Experiments

We conducted experiments with our method in order to discover interactions between genes in biological and medical papers. In this section, we present the extraction and validation of linguistic patterns for gene interaction detection and characterisation, and then the application of the selected patterns on a real dataset.

3.1 Extraction Rules

Training Data. Genes can interact with each other through the proteins they synthesize. Moreover, although there are conventions, biologists generally do not distinguish in their papers between the gene name and the protein name synthesized by the gene. Biologists know in context if the sentence is about the protein or gene. Thus, to discover the linguistic patterns of interactions between genes, we merge two different corpus containing genes and proteins.

The first corpus contains sentences from PubMed abstracts, selected by Christine Brun[4] as sentences that contain gene interactions. It contains 1806 sentences. That corpus is available as a secondary source of learning tasks "Protein-Protein Interaction Task (Interaction Award Sub-task, ISS)" from BioCreAtIvE Challenge II [8].

The second corpus contains sentences of interactions between proteins selected by an expert. That dataset, containing 2995 sentences with gene interactions, is described in [15].

[4] Institut de Biologie du Développement de Marseille-Luminy.

Sequential Pattern Extraction. We merged the two datasets previsouly presented and assigned a unique tag for the named entities: *AGENE@np*. A POS tagging is then performed using the *treetagger* tool [16]. The sentences are then ready to extract all the frequent sequential patterns. We set a support threshold, *minsup* equals to 10. It means that a sequential pattern is frequent if it appears in at least 10 sentences (i.e. 0.2% of sentences). Indeed, with that threshold some irrelevant patterns are not taken into account while many patterns of true gene interactions are discovered. Note that other experiments have been conducted with greater *minsup* values (15 and 20). With those greater *minsup* relevant patterns for interaction detection are lost. The number of frequent sequential patterns that are extracted is high. More than 32 million sequences are discovered. Although the number of extracted patterns is high the extraction of all frequent patterns takes only 15 minutes. The extraction tool is *dmt*4 [9].

The application of constraints significantly reduces the number of sequential patterns. Indeed, the number of sequential patterns satisfying the constraints is about $65,000$. However, this number is still prohibitive for analysis and validation by a human expert. Note that, the application of constraints is not time consuming. It takes a couple of minutes.

The recursive mining reduces significantly the number of sequential patterns. The sequential patterns obtained in the previous step are divided into several subsets. The recursive mining of each subset exhibits at most k sequential patterns to represent that subset. In this experiment, we set the parameter k to 4. It allows several patterns to be kept for each noun or verb in order to cover sufficient different cases (for example 4 patterns corresponding to 4 syntactic constructions with the verb *inhibit@vvn* are computed). In the same time it allows the patterns to be analysed by a user. The number of subsets, which are built, is 515 (365 for nouns, 150 for verbs). At the end of the recursive mining, there remains 667 sequential patterns that can represent interactions or their categorisations. That number, which is significantly smaller than previous one, guarantees the feasibility of an analysis of those patterns as information extraction rules by an expert. The recursive mining of those subsets is not time consuming. It takes about 2 minutes.

The 667 remaining sequential patterns were analyzed by two users. They validated 232 sequential patterns for interaction detection and 231 patterns for categorisation of interactions in 90 minutes. It means that 232 sequential patterns represent several forms of interactions between genes. Among those patterns, some explicitly represent interactions. For example, $\langle$*AGENE@np interact@vvz with@in AGENE@np .@sent*$\rangle$, $\langle$*AGENE@np bind@vvz to@to AGENE@np .@sent*$\rangle$, $\langle$*AGENE@np deplete@vvn AGENE@np .@sent*$\rangle$ and $\langle$*activation@nn of@in AGENE@np by@in AGENE@np .@sent*$\rangle$ describe well-known interactions (binding, inhibition, activation). Note that when the patterns are applied, 0 or several words may appear between two consecutive items of the pattern. For example, the pattern $\langle$*AGENE@np interact@vvz with@in AGENE@np .@sent*$\rangle$ matches the sentence "<gene_name=MYC> interacts with <gene_name=STAT3>." and also the sentence "<gene_name=MYC>

interacts with genes in particular <gene_name=STAT3>." Other patterns represent more general interactions between genes, meaning that a gene plays a role in the activity of another gene like ⟨*AGENE@np involve@vvn in@in AGENE@np .@sent*⟩, ⟨*AGENE@np play@vvz role@nn in@in the@dt AGENE@np .@sent*⟩ and ⟨*AGENE@np play@vvz role@nn in@in of@in AGENE@np .@sent*⟩. Note that the *"involve"* verb and the *"play role in"* phrase do not belong to the word lists of [19] and [7], also used by Hakenberg *et al.* [6] as terms representing interactions.

The remaining patterns represent modalities or biological context as described in Section 2.4.

The sequential patterns obtained are linguistic rules that can be used on biomedical texts to detect and characterise interactions between genes. Note that to be applied, those patterns do not need a syntactic analysis of the sentence. The process just tries to instantiate each element of the pattern in the sentence.

3.2 Application: Detection and Characterisation of Gene Interactions

In order to test the quality of the sequential patterns found in the previous section, we consider 442,040 biomedical papers from PubMed. In that dataset, the names of genes or proteins are labeled thanks to[5]. We randomly took 200 sentences and tested whether the linguistic patterns can be applied. For each sentence, we manually measure the performance of linguistic sequential patterns to detect those interactions and their characteristics. Note that we also carried out a POS tagging of those sentences in order to correctly apply the pattern language, most of applications of the linguistic sequential patterns is almost instantaneous.

Table 3. Detection and characterisation of interactions

	Precision	Recall	*f-score*
Interaction detection	0.83	0.75	0.79
Interaction categorisation	0.88	0.69	0.77

Table 3 presents the scores of the application of the patterns as extraction rules: Precision, Recall and *f-score*[6]. For the gene interaction detection, the precision is good and the recall is correct. Those results are comparable to the results of other methods in literature, however, we can note that the tasks are not the same [8]. For the interaction characterisation, the precision is good and the recall is about 69%. There are several reasons that explain why the recall is not greater. They are discussed in the next section.

3.3 Discussion

About Interaction Detection. Although the results of the POS tagger tool are mainly correct, there still be some labeling errors on lemmatization or

[5] http://bingotexte.greyc.fr/

[6] The used *f-score* function is : $f\text{-}score = \frac{2 \times Precision \times Recall}{Precision + Recall}$.

assignment of a grammatical category. Our method is robust with respect to that phenomenon, indeed those errors are also present in the extracted patterns. Thus, if an error is frequent, it appears in a pattern. For example, *treetagger* does not lemmatize the word *cotransfected* but some extracted patterns contain *cotransfected@vvn*.

Note that for the experiments the scope of extracted linguistic patterns is the whole sentence. That scope may introduce ambiguities in the detection of interactions when more than two genes appear in sentences. Several cases are possible: when several binary interactions are present in the sentence, when the interaction is n-ary ($n \geq 3$) or when an interaction is found with a list of genes. The case of n-ary interactions can be solved with a training dataset containing n-ary interactions. The other two cases can be treated by introducing limitations of pattern scope, for example cue-phrases (e.g. but, however).

False negatives depend on the absence of some nouns or verbs of interaction in the patterns. For example, the noun "modulation" is not learned in a pattern whereas the verb "modulate" appears in patterns. This suggests that the use of linguistic resources (e.g. lexicon or dictionary), manually or semi-automatically, can improve patterns and thus interaction detection.

About Interaction Characterisation. The false negatives, which are dependent on the absence of some patterns, are also an important problem for interaction characterisation. For example, in our experiments in the sentence *"<gene_name=SP1> binding is enhanced by association with <gene_name=CDK2> and <gene_name=CDK2>, both in vivo and in vitro ."* the biological situation "in vitro" is detected whereas "in vivo" is not detected. Indeed, there is no sequential pattern extracted from the training corpus that contains "in vivo". That case is considered as a false negative. The recall (69%) is strongly dependent on the number of false negatives. Note that the false negatives mainly come from missing biological context (about 92%). It is explained by the difficulty to have a training corpus that contains all biological context (e.g. body parts ("liver", "pituitary gland", ...), disaeses). The false negatives due to missing modalities are seldom (about 8%). Those false negatives are explained by the fact that patterns containing "perform" have not been validated by the human users as IE rules whereas those patterns may find some modalities.

4 Conclusion

The proposed approach aims at automatically discovering linguistic IE rules using sequential patterns filtered by linguistic constraints and recursive mining. Unlike existing methods, our approach is independent of syntactic parsing and does not require any resource except the training corpus to learn patterns. Note that in this training corpus interactions are not annotated. In addition, the implementation is simple. The sequential patterns, which are automatically generated, are used as linguistic rules. An advantage of the use of sequential patterns is that they are understandable and manageable IE rules. The expert can easily

modify the proposed rules or add other ones. We illustrated the method on the problem of the detection and characterisation, with some modalities and biological information, of gene interactions. However, the proposed approach can be straightforwardly applied to other domains without additional effort to develop custom features or handcrafted rules.

The experiments related on PubMed annotated corpus show that results are close to other approaches in literature. We are convinced that those results can be easily improved. Indeed, we used directly the discovered patterns as IE rules, without modifying them. Adding or enhancing patterns with expert knowledge, or using a specialized dictionary to enhance manually or semi-automatically the discovered patterns should reduce false negatives (and false positives also). Using heuristics to limit the scope of applied patterns (e.g. cue-phrases) should also improve the precision.

Acknowledgments. The authors would like to thank Christophe Rigotti (Université de Lyon - LIRIS, Fr) for invaluable discussions and for *dmt4*. This work is partly supported by the ANR (French National Research Agency) funded project Bingo2 ANR-07-MDCO-014.

References

[1] Agrawal, R., Srikant, R.: Mining sequential patterns. In: International Conference on Data Engineering (1995)
[2] Crémilleux, B., Soulet, A., Kléma, J., Hébert, C., Gandrillon, O.: Discovering Knowledge from Local Patterns in SAGE data. IGI Publishing (2008)
[3] Frawley, W.J., Piatetsky-Shapiro, G., Matheus, C.J.: Knowledge discovery in databases: An overview. In: Knowledge discovery in databases, pp. 1–30. AAAI/MIT Press (1991)
[4] Fundel, K., Küffner, R., Zimmer, R.: RelEx - relation extraction using dependency parse trees. Bioinformatics 23(3), 365–371 (2007)
[5] Giuliano, C., Lavelli, A., Romano, L.: Exploiting shallow linguistic information for relation extraction from biomedical literature. In: Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference (EACL). The Association for Computer Linguistics (2006)
[6] Hakenberg, J., Plake, C., Royer, L., Strobelt, H., Leser, U., Schroeder, M.: Gene mention normalization and interaction extraction with context models and sentence motifs. Genome biology 9(Suppl. 2) (2008)
[7] Hao, Y., Zhu, X., Huang, M., Li, M.: Discovering patterns to extract protein-protein interactions from the literature: Part ii. Bioinformatics (2005)
[8] Krallinger, M., Leitner, F., Rodriguez-Penagos, C., Valencia, A.: Overview of the protein-protein interaction annotation extraction task of BioCreative II. Genome Biology (2008)
[9] Nanni, M., Rigotti, C.: Extracting trees of quantitative serial episodes. In: Džeroski, S., Struyf, J. (eds.) KDID 2006. LNCS, vol. 4747, pp. 170–188. Springer, Heidelberg (2007)

[10] Nédellec, C.: Machine learning for information extraction in genomics - state of the art and perspectives. In: Text Mining and its Applications: Results of the NEMIS Launch Conf. Series: Studies in Fuzziness and Soft Comp. Sirmakessis, Spiros (2004)
[11] Ng, R.T., Lakshmanan, L.V.S., Han, J., Pang, A.: Exploratory mining and pruning optimizations of constrained association rules. In: SIGMOD Conference (1998)
[12] Pei, J., Han, B., Lakshmanan, L.V.S.: Mining frequent itemsets with convertible constraints. In: Proc. of the 17th Int. Conf. on Data Engineering, ICDE 2001 (2001)
[13] Pei, J., Han, B., Mortazavi-Asl, B., Pinto, H.: Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. In: Proc. of the 17th Int. Conf. on Data Engineering, ICDE 2001 (2001)
[14] Rinaldi, F., Schneider, G., Kaljurand, K., Hess, M., Romacker, M.: An environment for relation mining over richly annotated corpora: the case of genia. BMC Bioinformatics 7(S-3) (2006)
[15] Rosario, B., Hearst, M.A.: Multi-way relation classification: application to protein-protein interactions. In: Proc. of the conf. on Human Language Technology and Empirical Methods in Natural Language Processing. Association for Computational Linguistics (2005)
[16] Schmid, H.: Probabilistic part-of-speech tagging using decision trees. In: Proceedings of International Conference on New Methods in Language Processing (September 1994)
[17] Schneider, G., Kaljurand, K., Rinaldi, F.: Detecting protein-protein interactions in biomedical texts using a parser and linguistic resources. In: Gelbukh, A. (ed.) CICLing 2009. LNCS, vol. 5449, pp. 406–417. Springer, Heidelberg (2009)
[18] Srikant, R., Agrawal, R.: Mining sequential patterns: Generalizations and performance improvements. In: Apers, P.M.G., Bouzeghoub, M., Gardarin, G. (eds.) EDBT 1996. LNCS, vol. 1057. Springer, Heidelberg (1996)
[19] Temkin, J.M., Gilder, M.R.: Extraction of protein interaction information from unstructured text using a context-free grammar. Bioinformatics (2003)
[20] Zaki, M.: Spade: An efficient algorithm for mining frequent sequences. Machine Learning 42(1/2) (2001)
[21] Zweigenbaum, P., Demner-Fushman, D., Yu, H., Cohen, K.B.: Frontiers of biomedical text mining: current progress. Brief Bioinform. (2007)

Extraction of Genic Interactions with the Recursive Logical Theory of an Ontology

Alain-Pierre Manine, Erick Alphonse, and Philippe Bessières

[1] Université Paris 13
LIPN-CNRS UMR7030
F93430 Villetaneuse
{manine,alphonse}@lipn.univ-paris13.fr
[2] Institut National de la Recherche Agronomique
MIG-INRA UR1077
F78352 Jouy-en-Josas
philippe.bessieres@jouy.inra.fr

Abstract. We introduce an Information Extraction (IE) system which uses the logical theory of an ontology as a generalisation of the typical information extraction patterns to extract biological interactions from text. This provides inferences capabilities beyond current approaches: first, our system is able to handle multiple relations; second, it allows to handle dependencies between relations, by deriving new relations from the previously extracted ones, and using inference at a semantic level; third, it addresses recursive or mutually recursive rules. In this context, automatically acquiring the resources of an IE system becomes an ontology learning task: terms, synonyms, conceptual hierarchy, relational hierarchy, and the logical theory of the ontology have to be acquired. We focus on the last point, as learning the logical theory of an ontology, and *a fortiori* of a recursive one, remains a seldom studied problem. We validate our approach by using a relational learning algorithm, which handles recursion, to learn a recursive logical theory from a text corpus on the bacterium *Bacillus subtilis*. This theory achieves a good recall and precision for the ten defined semantic relations, reaching a global recall of 67.7% and a precision of 75.5%, but more importantly, it captures complex mutually recursive interactions which were implicitly encoded in the ontology.

1 Introduction

The elucidation of molecular regulations between genes and proteins, as well as the associated physical interactions, is essential in the understanding of living organisms, as they underlie the control of biological functions. However, their knowledge is usually not available in formatted information from widely accessed international databanks, but scattered in the unstructured texts of scientific publications.

For this reason, numerous works in recent years have been carried out to design Information Extraction (IE) systems, which aim at automatically extracting

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 549–563, 2010.

genic interaction networks from bibliography (see e.g. [1] for a review). To perform extraction, a possible method is to start with a model of the domain, i.e. an ontology, which defines concepts (e.g. gene, protein) and an interaction relation [2]. Then, an *ontology population* procedure is achieved [3]: concepts and relations mentioned in the text are recognized and instantiated. To do so, after a preliminary *terms* and *named entities recognition* step, which leads to the instantiation of main concepts, semantic relations are usually extracted by applying so-called *extraction patterns*, or *rules*. For instance, in the following sentence:

> Production of sigmaK about 1 h earlier than normal does affect Spo0A [...]

the protein concepts are first instantiated (sigmaK, Spo0A); subsequently, an interaction relation is instantiated between the sigmaK and Spo0A proteins. Rules applied to identify the former relation exhibit syntactico-semantic features (e.g., syntactic relations between sigmaK and Spo0A words) originated from NLP (Natural Language Processing) modules.

Designing rules in order to capture the relevant knowledge underlying the concept of *genic interaction* is a very difficult challenge, as this concept covers a wide variety of interdependent phenomenons (protein and gene regulations, DNA binding, phosphorylation, etc.). For instance, the previous example implies an unspecified regulation (sigmaK is only stated to "affect" Spo0A), whereas in the following sentence:

> Here, we show that GerE binds near the sigK transcriptional start site [...]

something very specific, a physical binding between the GerE protein and a DNA site, is described; furthermore, a more generic relation, an interaction between GerE and sigK, can be deduced from this binding, on which it depends. Despite this variety of relations, and their interrelations, most rules of IE systems are limited to extract a unique type of interaction relation. Consequently, they face a trade-off between recall and precision. Some favour precision by focusing on very specific and well-defined interactions, like protein-protein interactions (e.g. [4,5,6,7,8]), but neglect other biological phenomenons; whereas other stress on recall by extracting general relations (e.g. [9,10]), but face precision issues originating from the important lexical diversity.

To overcome this trade-off and to be able to model more accurately the biological field, IE systems require more expressive extraction rules. Firstly, it is not sufficient to address one single interaction relation: rules have to involve multiple relations, defined within an arbitrarily complex ontology [3], in order to model, for instance, that GerE *binds to* (first relation) a site *included in* (second relation) the sigK gene. Secondly, syntactico-semantic rules alone are inadequate. Semantic reasoning is needed to express semantic relations dependencies, and to deduce, for instance, that if GerE *binds to* a site *included in* sigK, then GerE *interacts with* sigK. Such a reasoning requires to be able to infer new relations (*interacts with*) from the previously instantiated ones (*binds to*, *included in*),

something beyond the inference capabilities of the current approaches. Thirdly, recursive or mutually recursive rules have to be handled; recursion is indeed intrinsic to natural language (see, for instance, [11,12]), as illustrated by the transitive nature of several relations: if the DNA site A is *included in* another site B itself *included in* C, then A is *included in* C.

We propose an integrated approach to address these three points, in which the logical theory of an ontology generalises regular IE patterns and is responsible for the extraction. We denote by *ontology* both a conceptual and a relational hierarchy (the thesaurus), along with a logical theory (see e.g. [13]), which expresses constraints and dependences between concepts.

The logical theory is able to refer to any concept defined in the ontology, and as such, to handle multiple inter-dependent relations, in accordance with our first point; these dependences may be recursive, in agreement with the third. Furthermore, ontologies exhibit inference capabilities of current knowledge representation languages, like OWL(-DL), Flogic or Datalog (see e.g. [14,15]), which allow to achieve semantic reasoning, as required by the second point. For instance, semantic knowledge may be expressed in Datalog by the following type of rules of the logical theory:

$$interact(A, B) \leftarrow bind\_to(A, C), included\_in(C, B),$$
$$protein(A), gene(B), dna\_site(C)$$

which means that: "A interacts with B, if A binds to a DNA site C, which is included in the gene B".

Extraction rules may be crafted by the domain expert as part as background knowledge, or automatically learnt with machine learning techniques. We choose the latter alternative, which has been well-motivated in IE as a generic component easily adaptable to new domains [16,17]. In our context, rule acquisition becomes part of an ontology learning task: terms, synonyms, the conceptual hierarchy (e.g. [18]), the relational hierarchy (e.g. [2]) and the logical theory of the ontology have to be learnt from a domain corpus. We focus on this latter point which has been seldom addressed, although it is an important prerequisite to complex knowledge-based systems, and we used the multiple predicate learning system Atre [19] to produce recursive rules with the suitable expressiveness. This work extends our previous work [3] in several ways. First, it motivates the use of the logical theory of the ontology as a proper generalization of the extraction rules or patterns. Second, neither relation dependencies nor recursion were taken into account during learning, which limited the expressiveness of the IE system: this is the key element that allows to conduct semantic reasoning and derive new relations from previously extracted ones. Finally, the corpus has been enriched with recursive and interdependent biological phenomena not processed previously, and is made publicly available[1].

The plan of the article is as follows. We discuss related works on IE and machine learning in section 2. We recall our ontology-population based IE plateform in section 3. We present our ontology learning strategy in section 4. In section 5,

[1] http://www-lipn.univ-paris13.fr/~alphonse/IE/genic_interaction

we report and comment our learning results on the bacterium corpus. Finally, in section 6, we discuss our approach and propose some perspectives in IE from text.

2 Related Works

Whereas we aim at automatically acquiring inference rules of an ontology, [20] notes that, in the ontology learning field, very few works are related to this task, as most researches focus on taxonomy and non-hierarchical relations learning. Work of [21] is loosely connected to it, as they learn simple association rules to handle paraphrases; more recently, [22] focus on learning non-domain-specific rules, like inclusion or disjointness statements between concepts, while we acquire domain-specific relations, like binding or regulatory interaction.

These rules cannot be acquired by machine learning techniques usually exploited to learn IE extraction patterns. Binary classification is indeed mostly used (e.g. [23,16]), and is limited to learn a single relation, whereas we need multiple conceptual relations. Furthermore, if multi-class learning is occasionally involved [4,24], this strategy does not yield to the required expressivity level, as they assume independence between target predicates, which forbids recursion. In the same way, a multi-class algorithm is used in [3], which only learns non-recursive syntactico-semantic patterns: in contrast to our approach, recursive clauses or rules based on *previously deduced relations* are not learnable. It was proposed to make use of stratified learning where recursive phenomena were identified, isolated and left to the expert: recursive rules were manually input during the ontology design. However, this approach does not scale well as it is too difficult to identify in the text those phenomena, implicit in the ontology.

To be able to produce recursive or mutually recursive target predicates, we chose to take advantage of a relational learning algorithm in the multiple predicate setting. To the best of our knowledge, the only other IE application of multiple predicate learning is found in [25], but is limited to named entities recognition, whereas we focus on extracting relations between recognized entities.

3 Information Extraction Platform

Our information extraction platform architecture is presented in figure 1. During production (right of the figure), it involves two main stages: firstly, a preliminary ontology population step, during which outputs of NLP modules are normalized in the ontology language by an *ontology population module*, and secondly, inference made by a *query module* based on the logical theory of the ontology in order to derive new instances.

3.1 Ontology Population Module

The first phase, the ontology population, is the extraction from text of instances of concepts and relations defined in the ontology. As it requires complex mappings between expressions in natural language to ontology structures [26], going beyond mere class/label linking (like the `rdf:label` property of RDF[2], or

[2] http://www.w3.org/TR/rdf-schema/

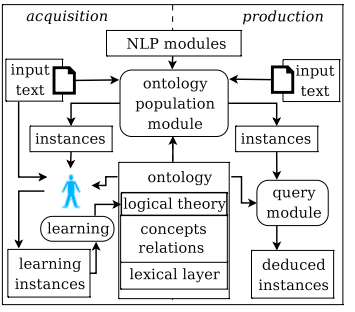


Fig. 1. Ontology-based IE platform

the more complex properties of SKOS[3], some authors introduce lexicon models [26,27] to ground the semantic information to the linguistic domain, even though they do not employ it in an IE context. We follow this approach by providing a so-called *Lexical Layer* (LL) along with the ontology. However, where the previous authors follow a linguistic point of view, by proposing a model to link ontology structures to lexical descriptions, we adopt an application-oriented perspective. Our LL is a task-dependent parameter: it comprises classes and relations required to link the output of NLP modules to the ontology, so it is designed with respect to those NLP modules. Its purpose is to provide a representation with sufficient expressiveness for efficient inference. These classes and relations define normalizations of text in intermediate stages of abstraction, between raw text and conceptual level. For instance, a LL relation may associate a syntactic label with an instance, or a syntactic relation between two instances (subject (*"subj"*) and object (*"obj"*) relations in figure 3). The LL is described in the same language as the ontology, so the inference rules can benefit from it.

Figure 2, in plain lines, exemplifies an output of the ontology population module. Instances of the *protein* concept (GerE, sigmaK) have been instantiated by a terminological module. They have been properly linked with existing domain knowledge, through the *product_of* semantic relation, which states that the protein sigmaK is encoded by the sigK gene. Subject (*subj*), object (*obj*), *comp_from*, and *comp_by* relations belong to the lexical layer, and their instantiations originate from a parser. A fragment of the corresponding ontology is shown in figure 3. Dashed lines exemplify the declarative definition of the lexical layer (e.g. *subj*, *obj*). "stimulate" is an instance of the concept *regulation*, and "use" is an instance of the *dependence* concept. Both are required to understand the presence of a regulation between proteins, and were thus added to the lexical layer. A transcription event occurs from (*t_from*) a promoter, and results from the action (*t_by*) of a protein. Therefore, promoters may be dependent (*p_dep*) of proteins. Finally, a protein complex results from the assembly of several proteins (*"complex_with"*): the protein complex EsigmaK is formed by a RNA polymerase complexed with the protein sigma K.

[3] http://www.w3.org/TR/2005/WD-swbp-skos-core-spec-20051102

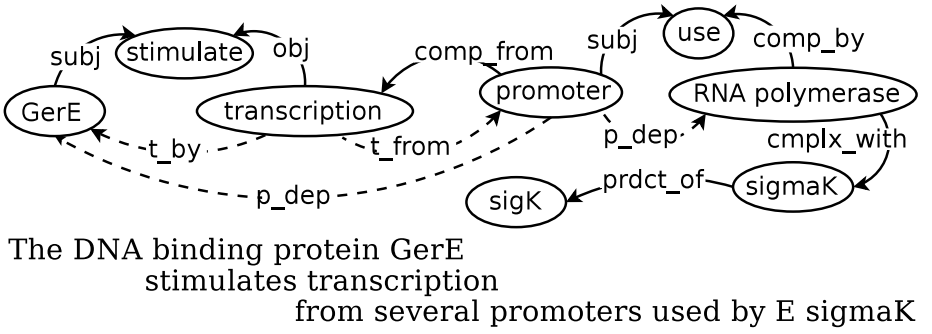


Fig. 2. Ontology Population output (plain lines), and some relations derived from the logical theory (dashed lines)

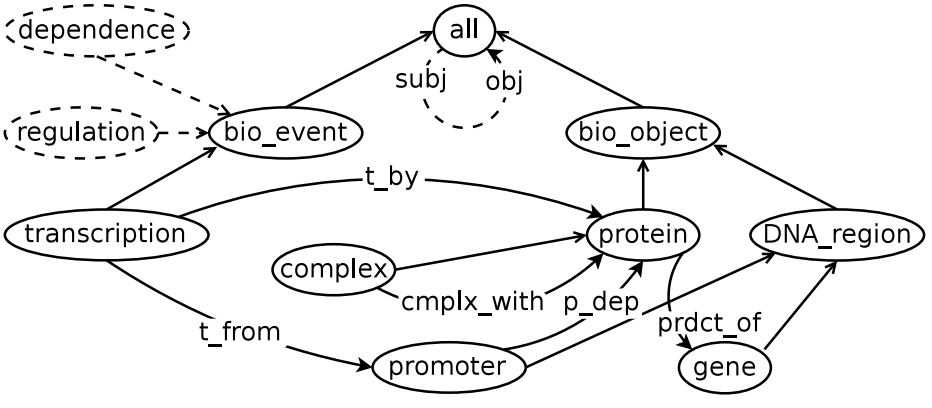


Fig. 3. Fragment of an ontology of biological interactions (lexical layer in dashed lines)

3.2 Query Module

The output of the ontology population module results from NLP modules and from domain knowledge (the latter allows us to know, for instance, that the sigma K protein is the product of the sigK gene, see figure 2). In opposition to traditional IE systems in which new facts are extensively extracted, here, knowledge is intensively encoded into the ontology structure, both within the conceptual hierarchy and within the logical theory, and is available through the mean of user queries. To benefit from the inference capabilities of our system, the logical theory of the ontology is used to derive more instances from those previously extracted. This is done through our *query module*. Figure 2, in dashed lines, exemplifies such deduced instances. Consider the following user query, related to the sentence in figure 2:

```
?- p_dep(A,B).
```

which means "is there a promoter A dependent on a protein B?". An answer of the query module will be:

```
A = promoter,
B = GerE
```

The promoter was inferred to be dependent on the GerE protein thanks to the logical theory of the ontology, encoded as a clausal theory written in Datalog. The following rules was used:

$$p\_dep(A, B) \leftarrow t\_by(C, B), t\_from(C, A)$$

It means that "if a transcription event C is due to a protein B and occurs from a promoter A, then A is dependent on B". In the example, the *promoter dependence* relations between *promoter* and *GerE* (*p_dep(promoter, GerE)*) is true as both the relevant *transcription by* (*t_by(transcription, GerE)*) and *transcription from* relations (*t_from(transcription, promoter)*) are true.

Note that the former rule involves semantic attributes, whereas the other dashed relations have been deduced from syntactico-semantic inference, based on features belonging to the lexical layer.

The *transciption by* relation (*t_by*) was deduced from a rule like this:

$$t\_by(A, B) \leftarrow subj(A, C), obj(B, C), \\ regulation(C), transcription(A), protein(B)$$

which asserts that the transcription A is caused by protein B, if A is subject of a regulation event C, and if an object relation links B to C. In figure 2, *t_by(transcription, GerE)* is true, as *subj(GerE, stimulate)* is true and *obj(transcription, stimulate)* is true.

The *transcription from* relation (*t_from*) was inferred from the following type of rule:

$$t\_from(A, B) \leftarrow comp\_from(B, A), \\ transcription(B), promoter(A)$$

which asserts that the syntactic relation *comp_from* has the semantic value of a *transcription from* relation if the arguments of the relation are respectively a transcription instance and a promoter instance. In the figure, *t_from(transcription, promoter)* is true as *comp_from(promoter, transcription)* is true.

The previous examples illustrate how rules of the logical theory form a major part of our system; the next section describes the approach that allowed to automatically acquire them.

4 Learning the Logical Theory of the Ontology

As opposed to previous approaches (see section 2), learning takes place in the ontology language to produce a logical theory that holds true in the domain ontology and the lexical layer. From a machine learning point of view, the learner

uses the ontology as the hypothesis language and instantiations of the ontology as the example language. During the acquisition of the theory, as illustrated in figure 1 (left part), the domain expert has to provide learning examples defined as instantiations of the ontology. He creates instances of concepts and relations of the ontology from a corpus, some instances being output by the ontology population module. Figure 2 exemplifies such annotation, the dashed lines corresponding to relations to learn.

Learning from such a relational language is known as Inductive Logic Programming (ILP) [28], where the hypothesis and the example languages are subsets of first-order logic. We encode the logical theory as a clausal theory, in Datalog. This is a knowledge representation language expressive enough for the task (as expressive as multi-relational databases), and theoretically well-understood in ILP, that most learners handle as learning language.

To learn from this relational language, we used the ATRE system [19], which handle recursive logical theories. A definition of a recursive theory, founded on the notion of *dependency graph*, is given by [19]. The dependency graph of a theory T is a directed graph $\gamma(T) = \langle N, E \rangle$, in which (i) each predicate of T is a node in N and (ii) there is an arc in E directed from a node a to a node b, iff there exists a clause C in T, such that a and b are the predicates of a literal occurring in the head and in the body of C, respectively.

This notion makes easier the characterization of multiple predicate learning relatively to multi-class learning: the dependency graph of a theory learned in the multi-class ILP setting will only comprise nodes, whereas in the multiple predicate case, it will include nodes and edges. Multiple predicate ILP may allow to learn recursive theory, i.e. a theory T where $\gamma(T)$ will contain at least one cycle.

The main problem to learn such a theory is related to the non-monotonicity property of the normal ILP setting [19]. In normal ILP setting, theories are induced thanks to a *separate-and-conquer* strategy: clauses are learnt one by one, covered examples are removed from the training set, and the process iterates until no more positive examples remained; in the multiple predicates paradigm, whenever two individual clauses are consistent in the data, their conjunction need not be consistent in the same data. ATRE addresses these issues by generating clauses all together, using a *separate-and-parallel-conquer* strategy.

ATRE represents examples as ground multiple-head clauses, called *objects*, which have a conjunction of literals in the head (because of space requirements, we refer the reader to [19] to an extensive description of ATRE). In our case, each sentence matches an object, and negatives examples were generated using a closed-world assumption. For instance, the previous example will be equivalently represented as[4]:

$$t\_by(id2, id1), p\_dep(id4, id1), t\_from(id2, id4),$$
$$\neg t\_by(id1, id2), \neg t\_by(id1, id3), [\ldots] \leftarrow$$
$$subj(id1, id3), obj(id2, id3), comp\_from(id4, id2),$$

[4] Some negative examples have been omitted.

$transcription(id2), protein(id1),$
$regulation(id3), promoter(id4).$

Note that all the ontological knowledge is given as background knowledge to the ILP algorithm, like the generalisation relation between concepts. For instance, specifying that a protein complex is a protein etc. will be represented as a clausal theory:

$protein(A) \leftarrow protein\_complex(A).$
$gene\_product(A) \leftarrow protein(A).$
$gene\_product(A) \leftarrow rna(A).$

Processing an example involving a protein complex or a RNA, the learning algorithm chooses the most relevant generality level (e.g. "protein complex", "protein" or "gene product") to learn the logical theory.

5 Results

As previously stated, extracting a regulation network in other works is mostly restricted to the extraction of a unique binary interaction relation. Consistently, recent trends regarding the application of machine learning to biological IE head toward the development of public annotated corpora, targeting such binary relations to compare systems' performances (e.g. AIMed [29], Bioinfer [30], HPRD50 [10], LLL [9]). In this paper, the ontology does not limit us to the extraction of a single relation, but allows the definition of numerous relations. We present a way to encode extraction patterns in order to infer new knowledge from them. Seemingly, public corpora are inadequate to validate the inference capabilities of the logical theory, as well as the relevance of multiple predicate ILP to acquire it.

We used the ontology of gene transcription in bacteria introduced in [3]. It describes the structural model of a gene, its transcription, and associated regulations, to which biologists implicitly refer in their texts. The ontology includes some forty concepts, mainly about biological objects (gene, promoter, binding site, RNA, operon, protein, protein complex, gene and protein families, etc.), and biological events (transcription, expression, regulation, binding, etc.). We focus on the ten defined conceptual relations: a general, unspecified, interaction relation (i), and nine relations specific to some aspects of the transcription (binding, regulons and promoters). The specific relations are the following: promoter dependence (p_dep), promoter of (p_of), bind to (b_to), site of (s_of), regulon member (rm), regulon dependence (r_dep), transcription from (t_from), transcription by (t_by), event target (et). As an illustration of their semantics, table 1 gives, for each relation, an expression where the relation is needed to normalise it. For instance, the third line in the table states that, in the sentence "GerE binds near the sigK transcriptional start site", the protein "GerE" (in bold font) binds to (b_to) the site "transcriptional start site" (in italics).

The lexical layer encompasses syntactic relations between classes, and syntactico-semantic classes aimed at factorizing entities, which may share the

Table 1. (From [3]) List of relations defined in the ontology, and phrase examples (sub-terms of the relation are shown in italic and bold)

Name	Example
p_dep	*sigmaA* recognizes **promoter elements**
p_of	the *araE* **promoter**
b_to	**GerE** binds near the sigK *transcriptional start site*
s_of	*-35 sequence* of the **promoter**
rm	*yvyD* is a member of sigmaB **regulon**
r_dep	*sigmaB* **regulon**
t_from	**transcription** from the Spo0A-dependent *promoter*
t_by	**transcription** by final *sigma(A)-RNA polymerase*
et	**expression** of *yvyD*
i	**KinC** was responsible for Spo0A~P *production*

Table 2. Results for multiple predicate learning (with recursion). Last column shows the number of examples.

Relation	Recall (%)	Prec. (%)	Number
i	50.2	70.6	225
rm	33.3	41.7	15
r_dep	100.0	100.0	12
b_to	69.6	75.3	79
p_dep	69.8	71.2	53
s_of	61.2	61.2	67
p_of	69.8	55.6	43
et	95.7	96.9	164
t_from	73.3	84.6	15
t_by	52.6	62.5	38
Global	67.7	75.5	711

same syntactical context (gene and protein, gene family and protein family, transcription and expression events).

We validate the interest of multiple predicate ILP in an ontology learning context by reusing the corpus presented in [3]. This corpus is a reannotation of the LLL corpus [9]: 160 sentences, provided with dependency-like parsing with resolved coreferences, have been reannoted with terms, concepts and relations according to the ontology. This corpus have been curated and augmented with new relations that were left out in [3] because they were matching expert rules with recursion or dependencies with other rules. In total, 711 relations were available for learning.

We used a ten-fold cross-validation to evaluate recall and precision of the IE process. In order to evaluate the gain of recursive rules, we ran Atre with and without recursive learning enabled. The results are shown in table 2 and table 3, respectively. Although recursion allows to model more complex interactions, it is interesting to note that the recursive theory also yields better results on this corpus, with a global recall of 67.7%, compared to 65.6%, and a precision

Table 3. Results for multi-class learning (without recursion). Last column shows the number of examples.

Relation	Recall (%)	Prec. (%)	Number
i	57.3	74.5	225
rm	33.3	62.5	15
r_dep	100.0	100.0	12
b_to	67.0	72.6	79
p_dep	67.9	61.0	53
s_of	73.1	54.4	67
p_of	69.7	44.1	43
et	76.8	96.1	164
t_from	60.0	81.8	15
t_by	47.3	69.2	38
Global	65.6	71.7	711

of 75.5%, compared to 71.7%. The scores are satisfactory, and corroborate the relevance of our ontology learning approach. More specific relations (*et*, *t_from*, *r_dep*) have little lexical variability, and reach high scores; on the contrary, more general ones, like *i*, exhibiting greater variability, are noticeably harder to learn. The poor score of *rm* may be due to an unbalanced distribution of this relation through ATRE's objects.

In the following, we will illustrate the benefit of the multiple predicate learning paradigm by outlining a typology of the learned rules. First of all, some rules only exhibit semantic attributes, allowing to exclusively reason on a semantic level.

$$i(X2, X1) \leftarrow t\_by(X2, X3), et(X3, X1). \tag{1}$$

$$s\_of(X2, X1) \leftarrow t\_from(X3, X2), et(X3, X1). \tag{2}$$

For instance, (1) expresses that if X1 is transcribed by X2, then they interact (e.g. "gspA" and "sigma B" in "transcription of gspA is sigma B dependent"); (2) asserts that if the X1 gene is transcribed from the X2 promoter, then X2 is a site included in X1 (e.g. "spoVD" and "promoter" in "spoVD transcription appears to occur from a promoter").

Multiple predicate setting is especially well-fitted to the hierarchical structure of ontologies:

$$s\_of(X2, X1) \leftarrow p\_of(X2, X1). \tag{3}$$

$$p\_of(X2, X1) \leftarrow s\_of(X2, X1), promoter(X2), gene\_entity(X1). \tag{4}$$

Rule (3), given by the expert as domain knowledge, encodes an *is-a* relation between *p_of* and *s_of*, whereas learned rule (4) allows to specialise a *s_of* relation into a *p_of* relation, if X2 is a promoter and X1 a gene. This is illustrated by the last example of the previous paragraph: thanks to (2) and (4), the system will

deduce a *p_of* relation between the promoter and the spoVD gene. Note that the rules (2), (3), (4) constitute a recursive theory.

Previous kind of rules are grounded to NL through predicates that involve LL-defined literals (i.e. syntactico-semantic attributes), like:

$$i(X2, X1) \leftarrow subj\_v\_n(X3, X1), \\ obj\_v\_n(X3, X2), term(X3, require). \tag{5}$$

$$i(X2, X1) \leftarrow subj\_v\_n(X3, X2), \\ obj\_v\_n(X3, X1), regulation(X3). \tag{6}$$

Rules (5) and (6) allow to derive semantic relations from syntactic relations. (5) is related to expressions like "A activates B", while (6) handles phrases like "B requires A" (note the argument order). These two rules show that Atre is able to learn classes of terms not explicitly defined by the expert to derive the argument order.

Our approach has the capacity to combine various abstraction levels in order to deduce new relations. For instance, the recursive rule (7) expresses that if protein X2 binds to (semantic level relation) site X3, included in (semantic level relation) site X4, then a *comp_n_n_of* (syntactic level relation) between X4 and X1 implies that X2 binds to X1 (e.g. "GerE" and "promoter" in "GerE binds to two sites that span the -35 region of the cotD promoter"). Previously inferred semantic relations may also be useful as contextual disambiguation clues. In (8), the *et* relation ensures that a *comp_v_pass_n_from* syntactic relation has the semantic value of a *t_from*.

$$b\_to(X2, X1) \leftarrow b\_to(X2, X3), s\_of(X3, X4), \\ comp\_n\_n\_of(X4, X1). \tag{7}$$

$$t\_from(X2, X1) \leftarrow et(X2, X3), \\ comp\_v\_pass\_n\_from(X2, X1). \tag{8}$$

Moreover, reasoning on multiple abstraction levels allows to factorize various lexical variations into a single semantic label. As a result, the learner produces more compact theories. Rule (9) clarifies this point. It will match expressions either like "the cwlB operon is transcribed by E sigma D" or like "transcription of cotD by sigmaK RNA polymerase", as the two forms "transcription of A" and "A is transcribed" are factorized by rules (10) and (11). In the multi-class ILP setting, two rules would have been required.

$$i(X2, X1) \leftarrow comp\_n\_n\_by(X3, X2), \\ et(X3, X1). \tag{9}$$

$$et(X2, X1) \leftarrow comp\_n\_n\_of(X2, X1), \\ event(X2). \tag{10}$$

$$et(X2, X1) \leftarrow subj\_v\_pass\_n(X2, X1), \\ transcription(X2). \tag{11}$$

6 Conclusion and Perspectives

Automatic extraction of genetic pathways from scientific litterature involves the modelling of a wide variety of semantic relations that are intrinsically interrelated. However, interrelations are neglected by traditional IE approaches, which only focus on the mapping of syntactico-semantic structures and semantic relations, and assume independence between semantic relations. In this paper, we introduced an IE platform that overcomes these limitations and exhibits inference capabilities going beyond existing systems by generalizing traditional IE patterns with the logical theory of an ontology. In particular, it allows to define multiple relations and to derive new relations from previously instantiated ones, when the former depend on the latter. Dependencies and recursive dependencies required by the logical theory are learnt from an annotated corpus by taking advantage of ILP in the multiple predicate setting, using the Atre system, which does not suffer from the independence assumption of usual machine learning approaches. We validated our system by learning a recursive logic theory from a bacterium corpus, and discussed its relevance for IE, especially its capacity to combine syntactic and semantic reasoning, and to benefit from the hierarchical structure of the ontology (specialisation and generalisation rules).

In the future, the declarative nature of our platform will allow its easy extension. Specifically, we plan to handle regulations, like inhibition and activation relations, a very important demand from biologists yet to be fulfilled. It may be due to the fact that these relations are inherently mutually recursive: only when we know that A inhibits B, which in turn inhibits C, that we can derive that A activates (or participates in the activation of) C.

Furthermore, we plan to survey the capacity of ILP tools, learning in the multiple predicate setting, to scale up and to handle noise, as this is a crucial requirement for NLP applications.

References

1. Ananiadou, S., Kell, D.B., Tsujii, J.: Text mining and its potential applications in systems biology. Trends in Biotechnology 24 (2006)
2. Ciaramita, M., Gangemi, A., Ratsch, E., Saric, J., Rojas, I.: Unsupervised learning of semantic relations between concepts of a molecular biology ontology. In: Proceedings of International Joint Conference on Artificial Intelligence (IJCAI 2005), Edinburgh, UK (2005)
3. Manine, A.P., Alphonse, E., Bessiere, P.: Information extraction as an ontology population task and its application to genic interactions. In: 20th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2008, vol. 2, pp. 74–81 (2008)
4. Craven, M., Kumlien, J.: Constructing biological knowledge bases by extracting information from text sources. In: Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology, pp. 77–86. AAAI Press, Menlo Park (1999)
5. Rindflesch, T., Tanabe, L., Weinstein, J., Hunter, L.: EDGAR: extraction of drugs, genes and relations from the biomedical literature. In: Proceedings of the Fifth Pacific Symposium on Biocomputing (PSB 2003), pp. 517–528 (2000)

6. Blaschke, C., Andrade, M.A., Ouzounis, C., Valencia, A.: Automatic extraction of biological information from scientific text: Protein-protein interactions. In: Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology, pp. 60–67. AAAI Press, Menlo Park (1999)
7. Ono, T., Hishigaki, H., Tanigami, A., Takagi, T.: Automated extraction of information on protein-protein interactions from the biological literature. Bioinformatics 17, 155–161 (2001)
8. Saric, J., Jensen, L., Ouzounova, R., Rojas, I., Bork, P.: Large-scale extraction of protein/gene relations for model organisms. In: First International Symposium on Semantic Mining in Biomedicine 2005 (2005)
9. Nédellec, C.: Learning language in logic — Genic interaction extraction challenge. In: Cussens, J., Nédellec, C. (eds.) Proceedings of the Fourth Learning Language in Logic Workshop (LLL 2005), pp. 31–37 (2005)
10. Fundel, K., Küffner, R., Zimmer, R.: RelEx — relation extraction using dependency parse trees. Bioinformatics 23, 365–371 (2007)
11. Hauser, M.D., Chomsky, N., Fitch, W.T.: The faculty of language: What is it, who has it, and how did it evolve? Science 298, 1569–1579 (2002)
12. Boström, H.: Induction of recursive transfer rules. In: Cussens, J., Džeroski, S. (eds.) LLL 1999. LNCS (LNAI), vol. 1925, pp. 52–62. Springer, Heidelberg (2000)
13. Gómez-Pérez, A.: Ontological engineering: A state of the art. Expert Update 2, 33–43 (1999)
14. McGuinness, D., van Harmelen, F.: OWL web ontology language overview: W3C recommendation, February 10, 2004, Technical report, W3C (2004)
15. Kifer, M., Lausen, G., Wu, J.: Logical foundations of object-oriented and frame-based languages. J. ACM 42, 741–843 (1995)
16. Salakoski, T., Rebholz-Schuhmann, D., Pyysalo, S. (eds.): Proceedings of the Third International Symposium on Semantic Mining in Biomedicine (SMBM 2008). Turku Centre for Computer Science (TUCS), Turku (2008)
17. Krallinger, M., Leitner, F., Valencia, A.: Assessment of the second BioCreAtIvE PPI task: Automatic extraction of protein-protein interactions. In: Proceedings of the Second BioCreAtIvE Challenge Evaluation Workshop, pp. 41–54 (2007)
18. Cimiano, P., Hotho, A., Staab, S.: Learning concept hierarchies from text corpora using formal concept analysis. Journal of Artificial Intelligence Research (JAIR) 24 (2005)
19. Varlaro, A., Berardi, M., Malerba, D.: Learning recursive theories with the separate-and-parallel conquer strategy. In: Proceedings of the Workshop on Advances in Inductive Rule Learning in conjunction with ECML/PKDD, pp. 179–193 (2004)
20. Buitelaar, P., Cimiano, P., Magnini, B.: Ontology learning from text: An overview. In: Buitelaar, P., Cimiano, P., Magnini, B. (eds.) Ontology Learning from Text: Methods, Evaluation and Applications. Frontiers in Artificial Intelligence and Applications, vol. 123. IOS Press, Amsterdam (2005)
21. Lin, D., Pantel, P.: DIRT discovery of inference rules from text. In: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 323–328. ACM, New York (2001)
22. Völker, J., Vrandecic, D., Sure, Y., Hotho, A.: Learning disjointness. In: Franconi, E., Kifer, M., May, W. (eds.) ESWC 2007. LNCS, vol. 4519, pp. 175–189. Springer, Heidelberg (2007)
23. Riloff, E.: Automatically generating extraction patterns from untagged text. In: Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI 1996), pp. 1044–1049. AAAI Press / The MIT Press (1996)

24. Rosario, B., Hearst, M.A.: Classifying semantic relations in bioscience texts. In: ACL 2004: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, Morristown, NJ, USA, p. 430. Association for Computational Linguistics (2004)
25. Berardi, M., Malerba, D.: Learning recursive patterns for biomedical information extraction. In: Muggleton, S., Otero, R.P., Tamaddoni-Nezhad, A. (eds.) ILP 2006. LNCS (LNAI), vol. 4455, pp. 79–93. Springer, Heidelberg (2007)
26. Cimiano, P., Haase, P., Herold, M., Mantel, M., Buitelaar, P.: LexOnto: A model for ontology lexicons for ontology-based NLP. In: Proceedings of the OntoLex 2007 Workshop held in conjunction with ISWC 2007 (2007)
27. Buitelaar, P., Sintek, M., Kiesel, M.: A multilingual/multimedia lexicon model for ontologies. In: Sure, Y., Domingue, J. (eds.) ESWC 2006. LNCS, vol. 4011, pp. 502–513. Springer, Heidelberg (2006)
28. Muggleton, S., Raedt, L.D.: Inductive Logic Programming: Theory and methods. Journal of Logic Programming 19, 20, 629–679 (1994)
29. Bunescu, R., Ge, R., Kate, R.J., Marcotte, E.M., Mooney, R.J., Ramani, A.K., Wong, Y.W.: Comparative experiments on learning information extractors for proteins and their interactions. Artificial Intelligence in Medicine 33, 139–155 (2005)
30. Pyysalo, S., Airola, A., Heimonen, J., Bjorne, J., Ginter, F., Salakoski, T.: Comparative analysis of five protein-protein interaction corpora. BMC Bioinformatics 9, S6 (2008)

Ontological Parsing of Encyclopedia Information*

Victor Bocharov, Lidia Pivovarova, Valery Rubashkin, and Boris Chuprin

St. Petersburg State University, Universitetskaya nab. 11, Saint-Petersburg, Russia
victor.bocharov@gmail.com, lidia.pivovarova@gmail.com,
vrubashkin@yandex.ru, boris@vr4591.spb.edu

Abstract. Semi-automatic ontology learning from encyclopedia is presented with primary focus on syntax and semantic analyses of definitions.

Keywords: Ontology Learning, Syntax Analysis, Relation Extraction, Encyclopedia, Wikipedia.

1 Introduction

Ontology Learning is a rapidly expanding area of Natural Language Processing. Many language technologies – from machine translation to speech recognition – should be supported by ontologies that provide conceptual interpretation encompassing the entire corpus vocabulary. However, a formal ontology, which is sufficient to encompass the entire lexis even in a narrow domain, should include a few dozen thousand concepts. Therefore, manual development of an ontology is a very time consuming process that can not be completed at the required level of completeness. Nowadays, this "bottleneck" problem is considered as the main obstacle to using ontologies [1]. This problem becomes even more complex if a universal knowledge base is necessary instead of a domain ontology. Therefore, ontology learning technologies are quite popular now.

It is possible to use different sources (such as natural language texts, machine readable dictionaries, semi-structured data, knowledge bases, etc.; a complete survey is presented in [2]) for ontology learning, which is generally understood as ontology development based on natural language. However, parsing of machine-readable dictionaries seems to be more effective. The main difference between a natural language text and a dictionary is the form of knowledge representation. Knowledge in a dictionary is more structured and compact than in free texts. In some cases, the structure is presented in dictionaries explicitly (as markups, tags, etc.), and otherwise it is expressed only by syntax.

Many efforts are currently underway in this area. (e.g., [3], [4], [5], [6], [7], [8], [9]). Nevertheless, we are unaware of any comparable effort for Russian dictionaries, though certain approaches to ontology learning from Russian free texts are known (e.g., [10], [11], [12]).

* This paper is supported by Russian Foundation For Basic Research, project №09-06-00275-a.

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 564–579, 2010.

2 Problem Statement and Basic Algorithm

We present here ontology learning from machine-readable version of "Russian Encyclopedic Dictionary" [13]. We use the entire dictionary with the exception of toponyms and proper names. A portion of the dictionary taken into consideration includes of 26,375 entries, which describe 21,782 different terms. The difference between these two figures is caused by presence of disambiguated terms (e.g., there are five different definitions for "aberration" in such areas as biology, physics, etc.).

The learned ontology is a universal ontology developed primarily for semantic text analysis. The basic structure for this ontology is represented by an attribute tree where objects alternate with attributes [15]. A small fragment of this tree is presented as an example below:

- TRANSPORT
  - BY ENERGY SOURCE
    - ELECTRIC TRANSPORT
    - ATOMIC TRANSPORT
    - FUEL TRANSPORT
    - WIND-DRIVEN TRANSPORT
  - BY ENVIRONMENT TYPE
    - AIR TRANSPORT
    - WATER TRANSPORT
    - LAND TRANSPORT
    - SPACE TRANSPORT

This structure provides the most natural way to present different links such as correspondence of a value to an attribute (**great color* vs. *great volume*), correspondence of an attribute to an object class (SOLID –> SHAPE vs. *LIQUID –> SHAPE), or a complete set of extension relations between concepts (incompatibility, intersection, inclusion). The ontology provides also representation of different associative relations, which are either unified (PART –> WHOLE, OBJECT –> LOCALIZATION, OBJECT –> FUNCTION, etc.) or specialized (COUNTRY –> CAPITAL, ORGANIZATION –> CHIEF, etc.).

Lexicon is an integral part of a working ontology, which connects a conceptual model with natural language units. Such a lexicon includes words and collocations that can be used to express various concepts. These words and collocations can represent standard terms (i.e., names of concepts used for the ontology) or their synonyms (we use the "synonym" term here in its broad sense as any natural language expression that refers to a respective concept with a reasonable probability).

We use our own ontoedidor [13] with additional tools for encyclopedia information import at the stage of ontology learning. Since the requirements for concept description in natural language processing are very strict, it is hardly possible to populate the ontology from our source in fully automatic fashion. Therefore, ontology learning is broken down into two stages: first, the dictionary entries are pre-classified automatically, and, second, an ontology administrator in given an opportunity to approve, change or cancel a decision made by the program. We discuss here primarily the first stage of this process, which represents automatic linguistic analysis of encyclopedia entries.

This linguistic analysis is based on the following simple hypothesis: usually, a hyperonym for a dictionary term is the first subjective-case noun of its definition (referred to hereafter as "basic word"). Several examples of typical dictionary entries, which correspond to this hypothesis, are shown below[1].

*АГРАФ – нарядная **заколка** для волос, с помощью которой крепили в прическах перья, цветы, искусственные локоны и т. д.*

*HAIRPIN – a **pin** to hold the hair in place.*

*ПЕРИСТИЛЬ – прямоугольный **двор**, сад, площадь, окруженные с 4 сторон крытой колоннадой.*

*PERISTYLE – a **colonnade** surrounding a building or court.*

*ЯТАГАН – рубяще-колющее **оружие** (среднее между саблей и кинжалом) у народов Ближнего и Среднего Востока (известно с 16 в.).*

*YATAGHAN - a long **knife** or short saber that lacks a guard for the hand at the juncture of blade and hilt and that usually has a double curve to the edge and a nearly straight back.*

As was demonstrated in pilot study [17], the structure of most dictionary entries corresponds to our hypothesis; however, its direct usage yields incorrect results occasionally. A list of the most frequent basic words selected at the first step of analysis [17] is shown in Table 1. A very simple lemmatizer was used to determine the first noun in each definition. The total of 4603 different first nouns are were identified using this technique.

The most frequent word here is *Иза*, a Russian woman name. *Из* (the plural form of this name in the genitive case), is a homonym of very frequent Russian preposition *из* (*from*). If this preposition is situated before any noun in the definition, the program selects it as a noun. This situation and some similar cases make it necessary to complete morphological information about grammemes instead of using simple lemmatization.

Table 1. List of the most frequently used basic words (according to pilot study [17])

Rank	Basic Word	Translation	Frequency	Rank	Basic Word	Translation	Frequency
1	ИЗА	IZA	475	18	ЗАБОЛЕВАНИЕ	DISEASE	186
2	ЧАСТЬ	PART	415	19	ПРОЦЕСС	PROCESS	182
3	СОВОКУПНОСТЬ	COMBINATION	406	20	СПОСОБ	APPROACH	169
4	НАЗВАНИЕ	NAME	389	21	БОЛЕЗНЬ	ILLNESS	164
5	СИСТЕМА	SYSTEM	347	22	##не выявлено ##	##undefined##	162
6	РАЗДЕЛ	SECTION	336	23	ЖИДКОСТЬ	LIQUID	154
7	ВИД	KIND	305	24	СОЕДИНЕНИЕ	COMPOUND	153
8	УСТРОЙСТВО	DEVICE	298	25	КРИСТАЛЛ	CRYSTAL	153
9	ПРИБОР	INSTRUMENT	286	26	ПОРОДА	BREED	141
10	МИНЕРАЛ	MINERAL	286	27	НАПРАВЛЕНИЕ	DIRECTION	137
11	ЕДИНИЦА	UNIT	264	28	ОРГАН	ORGAN	134
12	ФОРМА	FORM	232	29	НАУКА	DISCIPLINE	132
13	ГРУППА	GROUP	212	30	ТКАНЬ	TISSUE	132
14	ИНСТРУМЕНТ	TOOL	204	31	ЛИЦО	PERSON	120
15	ВЕЩЕСТВО	SUBSTANCE	202	32	ОБЛАСТЬ	PROVINCE	116
16	ЭЛЕМЕНТ	ELEMENT	198	33	ОТРАСЛЬ	BRANCH	116
17	МЕТОД	METHOD	194	34	КОМПЛЕКС	COMPLEX	109

[1] Relevant definitions taken from Webster dictionary (`http://www.merriam-webster.com/`) or English Wikipedia (`http://en.wikipedia.org/`) are shown here instead of translations of respective Russian definitions.

Then, there are such frequent words as *part*, *complex*, *name, kind*, *sort,* etc. These words cannot be used as basic words; they are more like links that mark relationship between a dictionary term and a proper basic word. The high frequency of using such words makes it necessary to apply additional logical-linguistic rules for extracting relations of different kind.

Finally, some other words are noticeable in this list. For example, *единица* is a part of Russian phrases *единица измерения* (*unit of measurement*) or *денежная единица* (*monetary unit*), which are very frequent in encyclopedic dictionary. Similarly, such frequently used words as *элемент (element)* and *лицо (person)* are parts of such phrases as *химический элемент* (*chemical element*) and *должностное лицо* (*official)* respectively. This fact justifies extraction of noun groups (in addition to single nouns) as basic words, and, therefore, it becomes necessary to use certain elements of syntactic analysis.

Very frequent occurrence of undefined basic words can be explained in two different ways. First, this phenomenon can be caused by certain errors, which are

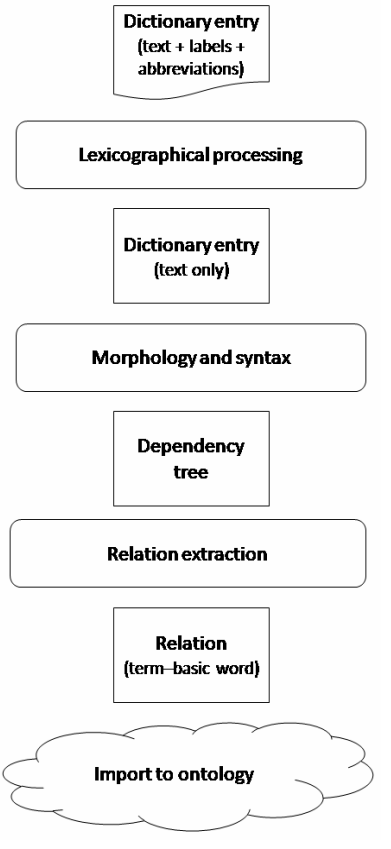


Fig. 1. The general framework of linguistic analysis

partly corrected herein. Second, it can indicate an unusual dictionary definition. For example: *МОРСКАЯ АРТИЛЛЕРИЯ – состоит на вооружении кораблей и береговых ракетно-артиллерийских войск* (*NAVAL ARTILLERY – is in service with naval ship or coastal defense troops*) – no noun in subjective case is present in this definition.

The general framework of linguistic analysis is shown in Figure 1. The rest of this paper describes every stage of this framework in more details.

3 Lexicographic Processing

Lexicographic processing is a preliminary step aimed to prepare a dictionary entry for morphology and syntax analyses. AOT (http://www.aot.ru/) open source tool is used for morphology and syntax analyses. Input text for this instrument should consist of well-formed Russian sentences. However, a dictionary is not written using exactly natural language text since it includes certain labels, abbreviations and extra punctuation.

Thus, lexicographic processing consists of the following steps:

- term recognition;
- recognition of domain labels, e.g., *в медицине* (*medical*), *в антропологии* (*anthropological)*, etc.;
- bracket text elimination;
- replacement of abbreviations by full forms of words.

The first three steps are executed for regular expressions. The last one is possible only if a context hints for an unambiguous form of an abbreviated word. Only most frequent abbreviations in certain already known contexts are replaced with full words.

Here are some examples:

на Сев. Кавказе → *на Северном Кавказе* (*at N. Caucasus* → *at the North Caucasus*). Russian adjectives have to agree grammatically with nouns. In the list of abbreviations, *Сев.* is associated with *Северный* (*North*) adjective. The form of the adjective can be copied from the respective noun form;

в 18 в. → *в 18 веке* (*in 18 c.* → *in the 18th century*). In this example, we use the prepositional government to determine the noun case.

If context is ambiguous, abbreviations are just eliminated.

4 Morphology and Syntax

At this step, we use context-free grammar to analyze first sentences of dictionary entries. The output of this step is represented by dependency trees. Since dictionary definitions usually start with a noun group that includes the base word, full syntax analysis is unnecessary. The grammar is very simple and aimed to recognize noun groups only. The grammar consists of the following rules:

```
[NP] -> [NOUN];
```

A noun group may consist of a single noun.

```
[NP] -> [ADJ] [NP root]
    : $0.grm:= case_number_gender($1.grm, $2.type_grm, $2.grm);
```

An adjective stays at the left side of a noun (this is a standard word order in Russian). The second line determines gender, number and case agreement between a noun and an adjective.

```
[NP] -> [NP root] [NP grm="рд"];
```

A noun group may be added at the right-hand side to another noun group in the genitive case (indicated by `"рд"` grammeme).

```
[PP] -> [PREP root] [NP];
```

A preposition and a noun group may be combined into a prepositional group.

```
[NP] -> [NP root] [PP];
```

A noun group may be added to a prepositional group at the right-hand side.

We use AOT tool to compile this grammar. The AOT output is an immediate constituent structure where roots of constituents are marked. An example of the constituent structure for phrase *верхняя одежда у некоторых азиатских народов* (*outdoor clothes of some Asian nations*), which is a definition for *халат* (*oriental robe*), is shown in Figure 2.

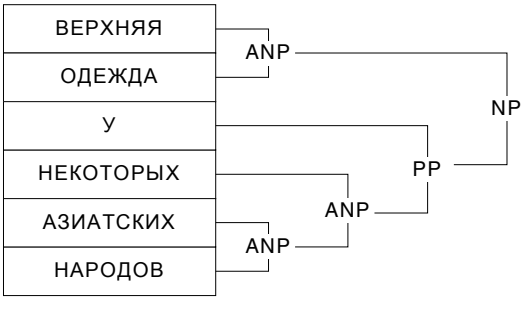


Fig. 2. An example of immediate constituents structure

Since a dependency tree is necessary for the subsequent steps of analysis, it is transformed using the following rules:

- a root governs other elements of the constituent;
- a constituent root is governed by the root of the immediate constituent of the upper level;

An example of dependency tree for the same phrase is shown in Figure 3.

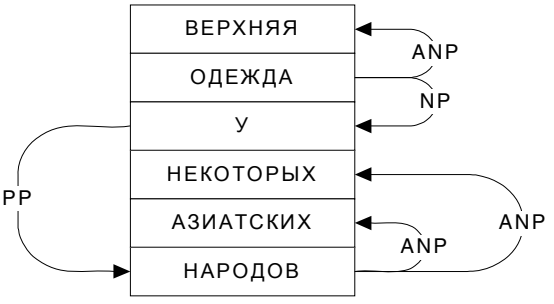


Fig. 3. Dependency tree

Morphology analysis is applied just ahead of syntax analysis. The result of morphology analysis is a set of morphological analysis outputs. Availability of multiple outputs for one word represents a very frequent situation as Russian is an inflectional language and the level of homonymy between different forms is very high. Conducting syntax analysis, we are able to avoid some "unproductive" forms that are not implemented in the dependency tree (the similar approach for French is presented in [18]). We discuss now *о Чукотском море* (*about Chukchee Sea*) phrase. There are three outputs of morphological analysis for *море (sea)*: *мор* (*pestilence*), prepositional case, singular, masculine gender; *море (sea)*, prepositional case, singular, neuter gender; and *мора (mora)*, prepositional case, singular, feminine gender. There are two outputs of morphological analysis for *чукотском* (*Chukchee*) word: *чукотский* (*Chukchee*) adjective in prepositional case and masculine or neuter gender. Only two outputs are agreed by gender (singular/plural forms and cases are identical), and thereby the third lemma – *мора (mora)* – has to be rejected. Unfortunately, two other outputs *мор* (*pestilence*) and *море (sea)* are still possible and, therefore, certain ambiguity is unavoidable here. However, dramatic decrease of ambiguity in Russian language can be achieved by applying syntax analysis. Our numerical results are presented in Table 2.

Table 2. Applying syntax for disambiguation

	Before syntax analysis	**After syntax analysis**
Average number of lemmas for one word form	1.27	1.06
Average number of morphological analysis outputs for one word form	2.26	1.64

5 Relation Recognition

Relation recognition is based here on logical-linguistic rules relevant to a dependency tree. Six types of semantic relations currently used in the ontology are extracted. These relation types are listed in Table 3.

Table 3. Extracted relation types

Relation Description	Notation
GENERALIZATION (IS-A) – default value	Gen
INSTANCE (reverse to Gen)	Spec
IDENTITY	Same
PART	Part
WHOLE (reverse to Part)	Whole
FUNCTION	Func
OTHER	Other

A specific rule is attached to a certain word. Our software parses the dependency tree and searches the first nouns in the definitions. Then, the rule attached to this word (if any) is executed. Each rule describes, first, the type of relation indicated by this word and, second, a directive of saving this word as a basic, or rejecting it and obtaining the next basic word candidate according to the rule. Two examples of rules for GENERALIZATION relation are presented in Table 4 as examples.

Table 4. Examples of GENERALIZATION relation rules

Basic word	Example	Rule	Result of application
род, *вид*, *сорт*, *тип*, … *kind*, *sort*, *type*, *class*, etc.	ФИЛЬДЕПЕРС – *высший сорт* ***фильдекоса***. PERSIAN THREAD – *the first class of* ***lisle***. ПИДЖИНЫ – *тип* ***языков****, используемых как средство межэтнического общения в среде разноязычного населения.* PIDGINS – a *sort of* ***languages****, used for communication between people with different languages*.	1. Save default type of relation (<Gen>) 2. Save next noun as a basic word (“next” means the next node in the dependency tree, which does not necessarily represent the next word in linear context).	ФИЛЬДЕПЕРС фильдекос GEN PERSIAN THREAD lisle GEN ПИДЖИН язык GEN PIDGIN language GEN
жанр *genre*	МИСТЕРИЯ – ***жанр*** *средневекового западноевропейского религиозного* ***театра***. MYSTERY – *a* ***genre*** *of the religious medieval* ***theatre***.	1. Save word as a basic word with default relation type 2. Save default type of relation (<Gen>) 3. Save the next noun as a basic word context.	МИСТЕРИЯ жанр GEN МИСТЕРИЯ театр GEN MYSTERY genre GEN MYSTERY theatre GEN

We discuss now these two rules in more details. The difference between them is that such words as *kind*, *sort,* etc. are eliminated while *genre* is saved. Therefore, there are two relations in the resulting output if *genre* is a basic word (in some cases, it is possible to extract even a larger number of different relations and save them as the result). We have two reasons to save *genre*: first, it is intuitively clear that this word is more sensible than *sort* and other similar words; second, in some cases the

definition may be too complicated for correct syntax analysis, and the program extracts at least one basic word in such cases.

Generally there are two main types of logical-linguistic rules:

1. Save the first basic word – change the type of relation – save the next basic word (the notation for this type is `save word - <relation name> - next noun`)
2. Reject the first basic word – change the type of relation – save the next basic word (`<relation name> - next noun`)

Choosing either of these types depends on the frequency of a particular structure and authors' introspection. Two additional examples for IDENTITY relation are presented in Table 5.

Table 5. Examples of IDENTITY relation rules

Basic word	Example	Rule	Result of application
обозначение nomination	СОЦИОСФЕРА – *обозначение **человечества**, общества, а также освоенной человеком природной среды, в совокупности составляющих часть географической оболочки.*	`<Same> - next noun`	СОЦИОСФЕРА человечество SAME
	SOCIOSPHERE – *a nomination of **humanity** as well as human assimilated environment arranged together in a part of geographical envelope.*		SOCIOSPHERE humanity SAME
явление phenomena	СИНЕСТЕЗИЯ – ***явление восприятия**, когда при раздражении данного органа чувств наряду со специфическими для него ощущениями возникают и ощущения, соответствующие другому органу чувств.*	`Save word - <Same> - next noun`	СИНЕСТЕЗИЯ явление GEN СИНЕСТЕЗИЯ восприятие SAME
	SYNESTHESIA – *a **perception phenomenon** with subjective sensation or image of a sense other than the one being stimulated.*		SYNESTHESIA phenomenon GEN SYNESTHESIA perception SAME

We have an additional reason to save *явление (phenomenon)* as a basic word: it is a part of such Russian phrases as *атмосферное явление* (*atmospheric phenomenon*), *физическое явление (physical phenomenon),* and so on. Our syntax analysis yields all grammatical information about noun phrases and this information has to be saved at the relation recognition step. The final choice between a single basic word and a basic collocation should be done by an ontology administrator.

More complicated rules, which can not be reduced to the two previous types, are used sometimes. An example of such a rule for FUNCTION relation is presented in Table 6.

Table 6. An example of complicated rule

Basic word	Example	Rule	Result of application
инструмент, прибор, аппарат, … *instrument, tool, device,* etc.	ФЕН – *электрический* ***аппарат*** *для* ***сушки*** *волос.* HAIRDRYER – *an electric* ***device*** *for hair* ***drying****.*	Save word – move to the next preposition If it is *для* (*for*): - change relation type to <Func> - save next noun	ФЕН аппарат GEN ФЕН сушка FUNC HAIRDRYER device_GEN HAIRDRYER drying FUNC

This rule factors in such a fact that functional relations in Russian are usually formed by preposition *для*, while dependent noun without preposition can not indicate a functional relation: *прибор темной окраски* (*darkly colored device*) vs. *прибор для окраски* (*device for coloring*).

The "Other" type of relation is very significant as it can result in modifications of the ontology model. Some examples are presented in Table 7.

Table 7. Examples of OTHER relation rules

Basic word	Example	Rule	Result of application
прерывание *termination*	АБОРТ – ***прерывание беременности*** *в сроки до 28 недель (то есть до момента, когда возможно рождение жизнеспособного плода).* ABORTION – *the* ***termination*** *of a* ***pregnancy*** *after, accompanied by, resulting in, or closely followed by the death of the embryo or fetus.*	`Save word - <Other> - next noun`	АБОРТ прерывание GEN АБОРТ беременность OTHER ABORTION termination GEN ABORTION pregnancy OTHER
способность *ability*	ХОМИНГ – ***способность животного*** *возвращаться со значительного расстояния на свой участок обитания, к гнезду, логову и т. д.* HOMING – *the* ***ability*** *of* ***animals*** *to come back from the considerable distance to their home range, nest, lie etc .*	`Save word - <Other> - next noun`	ХОМИНГ способность GEN ХОМИНГ животное OTHER HOMING ability GEN HOMING animal OTHER

These rules represent the intuitively recognized fact that abortion is relevant to pregnancy and homing is relevant to animals – even if it is difficult to specify such relevance. Such rules are applied to approximately 30 basic words. It is found unexpectedly that these 30 words can be broken down into the following two groups: (i) words, which indicate a certain feature of a term defined (e.g., *ability*), and (ii) words, which indicate certain transformation (e.g., *termination*).

The first group includes the following basic words: *характеристика* (*characteristic*), *признак* (*attribute*), *свойство* (*property*), *число* (*number*), *показатель* (*index*), *степень* (*degree*), *количество* (*quantity*), *характер* (*character*), *масса* (*mass*), *состояние* (*condition*), *способность* (*ability*), *место* (*place*), *источник* (*source*).

The second group includes primarily verbal nouns: *переход* (*transition*), *извлечение* (*extraction*), *превращение* (*transformation*), *введение* (*introduction*), *выделение* (*emission*),

возникновение (*origination*), *нарушение* (*deviation*), *прерывание* (*termination*), *развитие* (*evolution*), *образование* (*formation*), *увеличение* (*increase*), *уменьшение* (*decrease*).

A genitive noun is used very frequently after these words in Russian (e.g., *прерывание беременности* in the first example in Table 7). The equivalent English form is *of* + noun (such as *termination of a pregnancy* in this example). Sometimes these words form relatively long genitive chains: *увеличение показателя состояния…* (*an increase of an index of condition…*). Therefore, the rules are applied recursively.

The fact that a group of unrelated words is clustered with such ease is very significant. Therefore, further extension of the ontology model by adding these two types of relations deserve additional consideration.

In total, there are 18 different rules for 91 basic words. The list of most frequent basic words from pilot study [17] (a representative excerpts from this list is shown in Table 1) is used to develop specific rules. In particular, we formulate rules for 51 of 100 most frequently used basic words. These rules are applicable to a relative minority of all entries. We currently apply them to just 8484 of 26,375 entries. Probably, this number can be increased; however, no rules shall be attached to a majority of entries since the main hypothesis is valid for them. After the relation recognition step, the total number of different basic words grows slightly to 4679 (while 4603 possible basic words were found in pilot study [17]); however, these words are much more informative. The new list of the most frequent basic words obtained by applying our rules is presented in Table 8.

Table 8. Most frequent basic words

Rank	Basic Word	translation	frequency	rank	basic word	translation	frequency
1	УСТРОЙСТВО	DEVICE	332	18	РАСТЕНИЕ	PLANT	146
2	МИНЕРАЛ	MINERAL	322	19	ТКАНЬ	TISSUE	146
3	ЕДИНИЦА	UNIT	293	20	СООРУЖЕНИЕ	STRUCTURE	138
4	ПРИБОР	INSTRUMENT	292	21	МАТЕРИАЛ	MATERIAL	134
5	ВЕЩЕСТВО	SUBSTANCE	277	22	ЛИЦО	PERSON	133
6	ПРОЦЕСС	PROCESS	243	23	ОБЛАСТЬ	PROVINCE	121
7	ИНСТРУМЕНТ	TOOL	235	24	ИЗМЕРЕНИЕ	MEASUREMENT	117
8	ЭЛЕМЕНТ	ELEMENT	228	25	ИЗМЕНЕНИЕ	MODIFICATION	117
9	ЗАБОЛЕВАНИЕ	DISEASE	210	26	ВЕЛИЧИНА	MAGNITUDE	116
10	НАУКА	DISCIPLINE	199	27	ОБРАЗОВАНИЕ	FORMATION	114
11	СОЕДИНЕНИЕ	COMPOUND	184	28	ПРОДУКТ	PRODUCT	110
12	БОЛЕЗНЬ	ILLNESS	174	29	ДВИЖЕНИЕ	MOVEMENT	104
13	ПОРОДА	BREED	170	30	ВОСПАЛЕНИЕ	INFLAMMATION	98
14	ОРГАН	ORGAN	168	31	МЕРА	MEASURE	98
15	ЖИДКОСТЬ	LIQUID	166	32	УЧАСТОК	SITE	97
16	КРИСТАЛЛ	CRYSTAL	164	33	ПРОИЗВЕДЕНИЕ	CREATION	94
17	МАШИНА	ENGINE	158	34	АППАРАТ	MECHANISM	93

We evaluate our relation recognition approach by comparing its output with opinion of an expert who reads 200 dictionary entries and extracts basic words from them. For 90% of entries (179 of 200), the results obtained by the expert and our sofware are identical.

We analyze now those 21 dictionary entries, which are incorrectly processed by the program. Most of these errors (16 of 21) are caused at different steps of analysis by specific algorithm inaccuracies that can be eliminated by minor modifications. We expect to correct these inaccuracies in the nearest feature and achieve the theoretical level of accuracy of (179 + 16 = 195) / 200 = 97.5% by applying the proposed approach to this source.

However, for each of the other 5 of 200 dictionary entries, a basic word is missing from the definition text. These entries are inconsistent with the basic hypothesis that the basic word is the first subjective-case noun of the definition, and the proposed approach is unsuitable for processing such entries. There are also three dictionary entries where definition starts with a verb while the defined term is a subject. For example:

АБРАЗИВНЫЙ ИНСТРУМЕНТ – *служит для механической обработки (шлифование, притирка и другие).*
ABRASIVE TOOL – *is designed for mechanical processing (grinding, reseating, etc.).*

The grammar has to be dramatically expanded for processing such definitions. Another way is to analyze the entire dictionary entry (including the defined term) for recognizing *инструмент* (*tool*) as a basic word in this example.

Another type of unusual entries is represented by statements of natural laws, theorems, etc. where a definition represents an extended description of the respective defined object. For example:

АВОГАДРО ЗАКОН – *в равных объемах идеальных газов при одинаковых давлении и температуре содержится одинаковое число молекул.*
AVOGADRO'S LAW – *equal volumes of ideal or perfect gases, at the same temperature and pressure, contain the same number of particles, or molecules.*

This case is very similar to the previous one because it is possible to extract a basic word from the defined term: *закон* (*law*).

Another type of difficulties is represented by omission of a basic word. There is one such example in the evaluation set:

АБИТУРИЕНТ – *оканчивающий среднее учебное заведение.*
COLLEGE APPLICANT – *a person graduating from high school.*

The translation does not reflect difficulties with this example because there is a subject noun (*person*) in the English phrase, while in the Russian phrase (which represents a well-formed Russian sentence) it is absent. Such a word as *человек* (*person*) is probably missing from the Russian definition. An approach enabling us to reconstruct the eliminated word is necessary to overcome this difficulty. However, only one such case is found in the evaluation set (corresponding to less then 1% of the set scope), whereas a modified algorithm necessary to remedy this deficiency would be very complex and inaccurate.

6 Import to Ontology

The final step of import is manual; however, it can be simplified by using our ontoeditor. A table with all dictionary terms is listed under a respective tab of the ontoeditor. Each term is matched with its definition (using syntax markup), and, then, the first sentence of the definition and a basic word (extracted automatically as described in the previous sections) are shown in individual columns. This is the right part of ontoeditor in Figure 4.

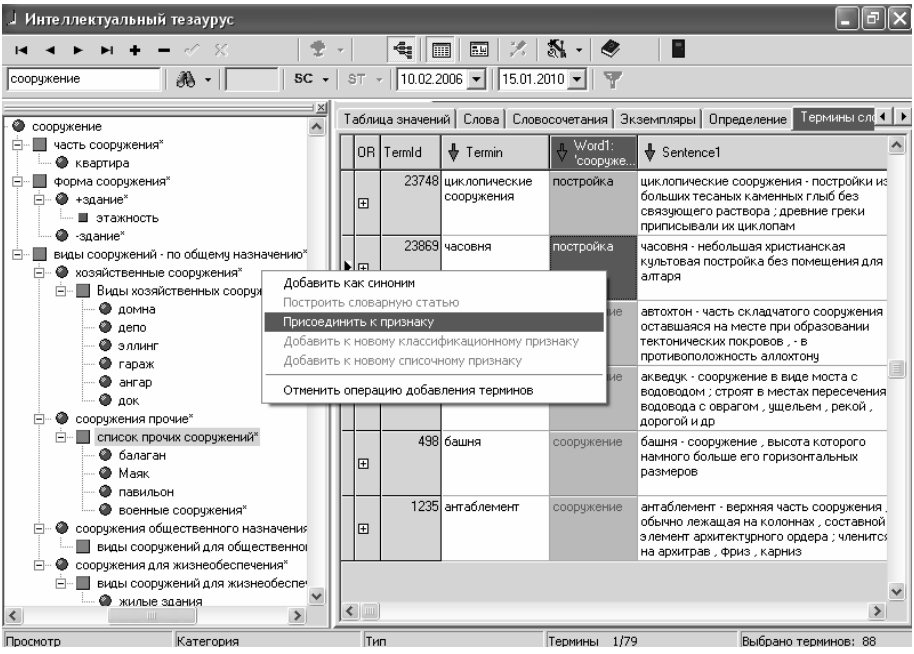


Fig. 4. The encyclopedia import into the ontoeditor

An ontology administrator selects a subset of dictionary entries for each individual import operation. There are three ways to make such a selection: (i) to specify a base word (factoring in all synonyms from the ontology lexicon); (ii) to specify a base word and all dependent concepts; and (iii) to specify a certain word in the definition governed by the base word. The ontology administrator may exclude irrelevant terms from the selection or include other terms. Then, the selection is imported to the ontology in one of the following ways:

- added as a synonym to the existing concept (which corresponds to the basic word);
- added as a new concept positioned in the taxonomy factoring in the basic word (the ontology administrator may add extra information to clarify these concepts); or
- added as an unsorted list of concepts (the ontology administrator may sort it later using the drug-and-drop interface)

Currently, rules for GENERATION relation (IS-A) only are added to our ontology. Processing of rules of other aforementioned types will be added to our software soon.

7 Wikipedia Parsing

The proposed approach is designed as scalable and applicable to other dictionary resources. We discuss now its trial application to Russian Wikipedia.

As Wikipedia is a free encyclopedia developed through community efforts, its content is larger than the content of any other dictionary, and the information and

execution quality of its individual articles varies. Wikipedia includes a lot of natural language information as well as its own taxonomy and templates. Using Wikipedia for ontology learning is quite popular ([19], [20], [21], [22], etc.); however, we are unaware of any comparable effort for Russian Wikipedia.

For our experiment we use the Russian Wikipedia dump of November 13, 2009, which includes 506,504 entries. We use Zemanta Wikiprep program (http://sourceforge.net/apps/mediawiki/wikiprep/) to convert articles from wiki markup to a plain text format.

Wikipedia includes different types of terms:

- abstract concepts;
- terms for different narrow domains;
- proper names (of persons, cities, streets, etc.); and
- lists (of dates, events, etc.)

Proper names and lists are out of scope of our study, and we filter them out by Wikipedia categories. The portion of Wikipedia taken into consideration includes 196,349 entries. The first sentence of each Wikipedia article, which includes a dash symbol, is used for analysis as a definition since this format is recommended by Wikipedia guidelines. A dash symbol is used as an equivalent of English *is-a* structure in Russian sentences with the predicate represented by a noun.

We apply an algorithm described in the previous sections to these first sentences. We evaluate the results against opinion of an expert who reads 500 Wikipedia entries and extracts basic words from them. For 82% of entries (410 of 500), the results obtained by the expert and our software are identical. We attribute approximately 40% of the errors (36 of 90 entries) to irregularities in the article texts. It is necessary to add some extra syntax and lexical-logical rules to our algorithm in order to remedy the other errors, and it will be a subject of future studies. Nevertheless, our approach is generally applicable to Wikipedia also.

8 Conclusion

We present an approach for discerning semantic relations automatically from the text of an encyclopedic dictionary. This approach is designed for semi-automatic addition of terms to ontologies. The original hypothesis that the first subjective-case noun of the definition represents a base term yields correct results for more than 90% of entries of Russian Encyclopedic Dictionary [10]. This hypothesis is applicable to practical extension of ontologies. The original hypothesis is clarified by developing methods for selection of a proper base word (when it is not represented by the root of the first noun group) and determining types of those semantic relations that do not fall under IS-A category. Certain definition structures, which can not be properly processed using our algorithm of automatic processing, are revealed. Such definitions are rare (1% of entries) in the encyclopedic dictionary that we processed.

The presented approach is generally applicable to other dictionary resources. We expect to apply it in future to Wikipedia and traditional explanatory dictionaries.

References

1. Gomez-Perez, A., Fernando-Lopez, M., Corcho, O.: Ontology Engineering. Springer, Heidelberg (2004)
2. Gomez-Perez, A., Manzano, D., Mancho, A.: Survey of Ontology Learning Methods and Techniques, IST Project IST-2000-29243 OntoWeb, Technical Report (2003)
3. Jannink, J.: Thesaurus Entry Extraction from an On-line Dictionary. In: Proceedings of Fusion 1999, Sunnyvale, CA (1999)
4. Rigau, G., Rodríguez, H., Agirre, E.: Building Accurate Semantic Taxonomies from Monolingual MRDs. In: Proceedings of the 17th International Conference on Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics COLING-ACL 1998, Montreal, Canada (1998)
5. Lee, C., Lee, G., Yun, S.J.: Automatic WordNet Mapping Using Word Sense Disambiguation. In: 38th Annual Meeting of the Association for Computational Linguistics (2000)
6. Agirre, E., et al.: Extraction of semantic relations from a Basque monolingual dictionary using Constraint Grammar. In: Euralex Sttutgart, Germany (2000)
7. Litkowski, K.C.: Digraph Analysis of Dictionary PrepositionDefinitions. In: Proceedings of the Asscociation for Computational Linguistics Special Interest Group on the Lexicon, Philadelphia, July 11 (2002)
8. Nichols, E., et al.: Multilingual Ontology Acquisition from Multiple MRDs. In: Proceedings of the 2nd Workshop on Ontology Learning and Population, Sydney, pp. 10–17 (2006)
9. Morita, T., Fukuta, N., Izumi, N., Yamaguchi, T.: DODDLE-OWL: A Domain Ontology Costruction Tool with OWL. In: Mizoguchi, R., Shi, Z.-Z., Giunchiglia, F. (eds.) ASWC 2006. LNCS, vol. 4185, pp. 537–551. Springer, Heidelberg (2006)
10. Ермаков А. Е. Автоматизация онтологического инжиниринга в системах извлечения знаний из текста // Труды международной конференции «Диалог 2008» - М.:Наука (2008), Ermakov, A.E.: The Atomation of Ontology Engineering for Knowledge Acquisition Systems. In: Proceedings of Dialogue 2008 International Meeting, Nauka, Moscow (2008) (in Russian)
11. Минаков И. А. Системный анализ, онтологический синтез и инструментальные средства обработки информации в процессах интеграции профессиональных знаний // Автореферат диссертации на соискание ученой степени доктора технических наук – Самара (2007), Minakov, I.A.: System analysis, ontological synthesis and tools for information processing for professional knowledge integration. PhD thesis, Samara (2007) (in Russian)
12. Пекар В.И. Автоматическое пополнение специализированного тезауруса // Труды международной конференции «Диалог 2002» - М.:РГГУ (2002), Pekar, V.I.: The Domain Thesaurus Learning. In: Proceedings of Dialogue 2002 International Meeting. RGGU, Moscow (2002) (in Russian)
13. Российский энциклопедический словарь - Гл. ред.: А. М. Прохоров — М.: Большая Российская энциклопедия (2001), Russian Encyclopedic Dictionary. Prohorov, A.M. (ed.) Russian Encyclopedic Dictionary, Moscow (2001) (in Russian)
14. Рубашкин В. Ш. Семантический компонент в системах понимания текста // КИИ-2006. Десятая национальная конференция по искусственному интеллекту с международным участием. Труды конференции. – М.: Физматлит, pp. С455– С463 (2006), Rubashkin, V.S.: Semantic Component in Understanding Text Systems. In: Proceedings of Tenth National Meeting on Artificial Intelligence (KII 2006), Fizmatgiz, Moscow, pp. 455 – 463 (2006) (in Russian)

15. Рубашкин В. Ш. Онтологии - проблемы и решения. Точка зрения разработчика. // Труды международной конференции «Диалог 2007» - М.:Наука (2007), Rubashkin, V.S.: Ontologies: Problems and Solutions. Developer's Point of View. In: Proceedings of Dialogue 2007 International Meeting, Nauka, Moscow, pp. 456 – 458 (2007) (in Russian)
16. Рубашкин В. Ш., Пивоварова Л. М Онторедактор как комплексный инструмент онтологической инженерии. // Труды международной конференции «Диалог 2008» - М.:Наука (2008), Rubashkin, V.S., Pivovarova, L.M.: Ontoeditor as a Complex Tool for Ontology Engineering. In: Proceedings of Dialogue 2008 International Meeting, Nauka, Moscow, pp. 456 – 458 (2008) (in Russian)
17. Рубашкин В.Ш., Капустин В.А. Использование определений терминов в энциклопедических словарях для автоматизированного пополнения онтологий - XI Всероссийская объединенная конференция "Интернет и современное общество" – СПб (2008), Rubashkin, V.S., Kapustin V.A.: Ontology Learning from Encyclopedia Entries Definitions. In: Proceedings of Internet and Modern Society International Meeting, Saint-Petersburg (2008) (in Russian)
18. Fernandez, M., Clergerie, E., Vilares, M.: Mining Conceptual Graphs for Knowledge Acquisition. In: Procedings of Workshop on Improving Non-English Web Searching Inews 2008, Napa Valley, USA, pp. 25–32 (2008)
19. Kassner, L., Nastase, V., Strube, M.: Acquiring a Taxonomy from the German Wikipedia. In: Proceeding of the 6th International Conference on Language Resources and Evaluation, Marakech, Morocco (2008)
20. Ruiz-Casado, M., Alfonseca, E., Okumura, M., Castells, P.: Information Extraction and Semantic Annotation of Wikipedia. In: Proceeding of the 2008 conference on Ontology Learning and Population: Bridging the Gap between Text and Knowledge, pp. 145–169 (2008)
21. Ponzetto, S.P., Strube, M.: WikiTaxonomy: A Large Scale Knowledge Resource. In: Proceedings of the 18th European Conference on Artificial Intelligence, Patras, Greece, July 21-25, pp. 751–752 (2008)
22. Wu, F., Hoffmann, R., Weld, D.S.: Information Extraction from Wikipedia: Moving down the Long Tail. In: Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 731–739 (2008)

Selecting the N-Top Retrieval Result Lists for an Effective Data Fusion

Antonio Juárez-González[1], Manuel Montes-y-Gómez[1], Luis Villaseñor-Pineda[1], David Pinto-Avendaño[2], and Manuel Pérez-Coutiño[3]

[1] Laboratory of Language Technologies,
National Institute of Astrophysics, Optics and Electronics (INAOE), Mexico
{antjug,mmontesg,villasen}@inaoep.mx
[2] Faculty of Computer Science,
Autonomous University of Puebla (BUAP), Mexico
dpinto@cs.buap.mx
[3] Vanguard Engineering Puebla (VEng), Mexico
mapco@v-eng.com

Abstract. Although the application of data fusion in information retrieval has yielded good results in the majority of the cases, it has been noticed that its achievement is dependent on the quality of the input result lists. In order to tackle this problem, in this paper we explore the combination of only the *n*-top result lists as an alternative to the fusion of all available data. In particular, we describe a heuristic measure based on redundancy and ranking information to evaluate the quality of each result list, and, consequently, to select the presumably *n*-best lists per query. Preliminary results in four IR test collections, containing a total of 266 queries, and employing three different DF methods are encouraging. They indicate that the proposed approach could significantly outperform the results achieved by fusion all available lists, showing improvements in mean average precision of 10.7%, 3.7% and 18.8% when it was used along with Maximum RSV, CombMNZ and Fuzzy Borda methods.

1 Introduction

The great amount of available digital content has motivated the development of several information retrieval (IR) approaches, which help users to locate useful documents for their specific information needs. All these approaches differ one from another in several issues such as the preprocessing process, the data representation, the weighting scheme and the similarity measure [3]. Evaluation exercises (see for instance [1, 23]) have evidenced that there is not a leading IR approach from all this variety, and, furthermore, that the performance of IR is highly affected by the nature and complexity of collections and queries. In particular, they have shown that different methods may achieve the best performance scores for different queries as well as they may retrieve distinct relevant documents for the same query.

The above situation explains why data fusion (DF), which goal is to enhance the retrieval results by taking advantage of the strengths of several methods, has become one of the most used strategies in IR. Particularly, the last two decades have

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 580–589, 2010.

witnessed a lot of work concerning the design and development of different DF methods specially suited for IR tasks [4, 8, 12, 13, 16, 21].

Although the application of DF in IR has yielded good results in the majority of the cases, it has been noticed that its achievement is dependent on the quality of the input result lists [5, 8, 17, 22, 24]. This dependence is mainly because the widespread use of DF consists in combining all available results lists obtained for a specific query without considering any information about them. Evidently, under this scenario, the presence of some poor-quality lists (containing very few relevant documents) may cause a significant drop in the fusion performance.

In order to tackle the above problem, in this paper we consider the combination of only the *n*-best result lists as an alternative to the fusion of all available data. In particular, we describe a heuristic measure to evaluate the quality of each result list, and, consequently, to select the presumably *n*-top lists per query. The proposed measure attempts to estimate the quality of result lists based on the assumption that a document occurring in several lists has more probability for being relevant, and, therefore, that the lists containing the major number of likely relevant documents at the very first positions are the ones more suitable for being combined.

Preliminary results in four data sets, considering a total of 266 queries, and employing three different DF methods are encouraging. They indicate that in scenarios including lists of diverse qualities, the proposed approach could significantly outperform the results achieved by fusion all result lists, showing improvements in mean average precision that range from 6% to 62.2%.

The rest of the paper is organized as follows. Section 2 describes the related work in DF applied to IR. It mainly discusses some efforts regarding the improvement of DF results. Section 3 introduces the method proposed for estimating the quality of the result lists and for their subsequent selection. Section 4 describes the experimental setup, whereas, Section 5 shows the results regarding the fusion of only the *n*-top result lists per query, obtained using four different data sets. Finally, Section 6 presents our conclusions and exposes further research directions.

2 Related Work

Broadly speaking, data fusion (DF) is the process of combining information gathered by multiple agents (sources, schemes, sensors or systems) into a single representation or result [10]. In IR it has been used to combine results from several retrieval approaches into a "better" single result list. In particular, in this area DF methods differ one from another in the way they compute the final score of documents. Some methods directly use the retrieval status values of the documents across the lists [4, 11, 21], other consider their rank [8, 13], and others their probability of occurring in a predefined segment of the lists [12, 14]. In addition, some recent methods are based on the Social Choice Theory [16, 18], and use pair wise contests of documents to determine their final score.

The application of DF in IR has shown relevant results in the majority of the cases; nevertheless, it has been noticed that it is sensitive to several factors. On the one hand, its performance is affected by the quality of the input lists, and, on the other hand, the

selection of the appropriate DF method depends on characteristics such as the redundancy and complementarity of the lists.

Regarding these problems, Gopalan and Batri [9] proposed a supervised method for selecting the m-best retrieval approaches and the best DF method for a given target document collection, and Diamond and Liddy [5] introduced the idea of learning a different linear weighted fusion function for each query instead of using the same static function to all queries.

More recently, some works have focused on investigating the feasibility of predicting the performance of the fusion of a given set of result lists [17, 22, 24]. To some extent, they have demonstrated that an appropriate selection of the input lists may result in a significant improvement of the DF process. However, given that these works consider the relevance judgments as central information for their predictions, they can only be considered as insightful studies about this phenomenon, but cannot be applied as automatic selection procedures.

Supported on the results of these studies, in this paper we consider the combination of only the *n*-top result lists as an alternative to the fusion of all available data, and, going a step forward, we propose an unsupervised method for selecting the presumably *n*-best lists per query. The major differences of our method in comparison to previous approaches are that it considers the selection the *n*-best lists for each individual query, and it does not depend on user relevance judgments nor on a priori information about the used IR methods.

3 Selecting the N-Top Result Lists

As we previously mentioned, the performance of DF is commonly affected by the quality of the input lists. Motivated by this situation, in this paper we explore the idea of combining only the *n*-top result lists as an alternative to the fusion of all available data. Under this proposal, the major problem is the selection of the *n*-top result lists for each query, which can be defined as the problem of determining the set of lists having the greatest relevance values in accordance to a specified measure.

More formally, given a set of *m* result lists $R = \{L_1, L_2, \ldots, L_m\}$, where L_i indicates a list of documents (i.e., $L_i = \{d_1, d_1, \ldots, d_{|L_i|}\}$), and a relevance measure Q, the problem of selecting the *n*-top result lists consists in identifying the set of n lists $T \subset R$ with the greatest relevance values, such that:

$$\forall(L_i \in T, L_j \notin T)\ \ Q(L_i) > Q(L_j) \tag{1}$$

Due to our intention about developing a fusion strategy that does not depend on the user relevance judgments nor consider information of the IR methods, we decided to design a measure that evaluates the relevance of the lists according to their intersimilarities, by using information about the redundancy and ranking of documents across them. In particular, we relied on the idea that the relevance of a list must be incremented by the presence of common documents at the very first positions.

Formula 2 shows the proposed relevance measure, where $q(d_k, L_i)$ denotes the contribution of document d_k to the relevance or quality of list L_i, and $r(d_k, L_i)$ indicates the position (rank) of d_k in the list L_i.

$$Q(L_i) = \sum_{\forall d_k \in I} q(d_k, L_i) \tag{2}$$

$$q(d_k, L_i) = 1 - \frac{ln(r(d_k, L_i))}{ln(|L_i|)} \tag{3}$$

It is important to comment that our first attempt to measure the value of q was $q(d_k, L_i) = 1/r(d_k, L_i)$. Nevertheless, using this direct formula was not possible to achieve satisfactory results, since it severely castigated the contribution of most documents to the global relevance value. In order to reduce the enormous differences in the values of consecutive documents in the lists, especially at the very first positions, we modified this formula by including a smoothing factor as showed in Formula 3. With this modification the values of the first five documents are 1, 0.9, 0.85, 0.8 and 0.77 respectively[1], instead of 1, 0.5, 0.33, 0.25 and 0.2.

Section 5 presents the DF results achieved in four different data sets when the proposed measure was used to select the *n*-top result list for each query.

4 Experimental Setup

In order to evaluate the proposed DF approach, we used four different data sets from the CLEF[2]. In particular, we considered a total of 189,477 documents, 266 queries, and three different DF methods. The following sections give further details about these data sets and the used evaluation measure.

Table 1. Data sets used in our experiments

Data set	Queries	Supported Queries	Number of Documents	Relevant docs per query (*average*)
Ad-hocCLEF	50	50	169,477	39
GeoCLEF	25	24	169,477	26
ImageCLEF	39	39	20,000	60
RobustCLEF	160	153	169,477	28

4.1 Data Sets and Result Lists

We used four data sets corresponding to the following CLEF tracks: 2005 Ad-hoc English retrieval [6], 2008 Geographic IR [15], 2008 Image Retrieval [2], and 2008 Robust IR [1]. Table 1 describes some data about these collections. It is important to clarify that in the experiments we only considered the set of supported queries, that is, the queries that have at least one relevant document in the reference collection.

Given that our goal was to evaluate the DF process, we consider five retrieval result lists per query for each data set. In all cases, five different retrieval systems were

[1] This values were calculated under the assumption that $|L_i| = 1000$.
[2] Cross-Language Evaluation Forum (*www.clef-campaign.org*).

used to retrieve the result lists. In particular, for the GeoCLEF data set, we used some IR systems developed in [20], which differ one from another in the use of different relevance feedback and ranking refinement techniques. For the ImageCLEF data set, the result lists were retrieved using different combinations of visual and textual features [7]. Finally, for the ad-hoc English track and RobustCLEF data sets, we used five distinct retrieval strategies implemented in the Lemur IR toolkit[3]; these strategies considered different retrieval models as well as different weighting schemes, such as the vector space model and the probabilistic model with Boolean and Frequency-based weightings.

4.2 Data Fusion Methods

In order to obtain general conclusions about the proposed method, we considered three different DF methods: *Maximum RSV* (from linear combination methods), *CombMNZ* (from positional methods), and *Fuzzy Borda Count* (from social choice theory methods). We did not consider probabilistic-based fusion methods because they imply a previous training and our approach is aimed to be fully unsupervised.

Following we present a brief description of the used DF methods. For more details on linear combination fusion methods refer to [4, 11, 21], on CombMNZ go to [8, 13], and on Fuzzy Borda consult [18].

4.2.1 Maximum RSV

This method sorts all documents in the lists by their normalized retrieval status value (RSV), computed independently from each IR system. In the case of repeated documents, the one with the highest value is considered for the final list.

Formally, let $R = \{L_1, L_2, \ldots, L_m\}$ be the set of m result lists, $L_i = \{d_1, d_1, \ldots, d_{|L_i|}\}$ a list of retrieved documents, and $D = \bigcup_i L_i$ the set of all different documents in the lists. Then, the final score for each document $d_k \in D$ is computed as defined in (4), where $v(d_k, L_i)$ is the normalized RSV of document d_k in the list L_i.

$$MaxRSV(d_k) = \max_{\forall L_i \ni d_k} \left(v(d_k, L_i)\right) \tag{4}$$

4.2.2 CombMNZ

Using the same notation from the previous section, this DF method sorts documents from D in decreasing order according to the following score.

$$combMNZ(d_k) = \left(\sum_{\forall (L_i \in R)} e(d_k, L_i)\right)\left(\sum_{\forall (L_i \in R, L_i \ni d_k)} |L_i| - r(d_k, L_i) + 1\right) \tag{5}$$

$$e(d_k, L_i) = \begin{cases} 1 & \text{if } L_i \ni d_k \\ 0 & \text{if } L_i \not\ni d_k \end{cases}$$

where $e(d_k, L_i)$ indicates the existence of document d_k in the list L_i, and $r(d_k, L_i)$ its rank in the list.

[3] www.lemurproject.org

4.2.3 Fuzzy Borda Count

This DF method considers the set of lists R as a set of experts that establish their preference for different alternatives (i.e., documents) by means of pairwise contests. It mainly sorts documents from D in decreasing order according to the following score:

$$FuzzyBorda(d_k) = \sum_{\forall (L_i \in R, L_i \ni d_k)} p(d_k, L_i) \tag{6}$$

$$p(d_k, L_i) = \sum_{\forall d_j \in L_i} c_{L_i}(d_k, d_j)$$

$$c_{L_i}(d_k, d_j) = \begin{cases} \dfrac{v(d_k, L_i)}{v(d_k, L_i) + v(d_j, L_i)} & \text{if } v(d_k, L_i) \geq v(d_j, L_i) \\ 0 & \text{Otherwise} \end{cases}$$

where $v(d_k, L_i)$ is the normalized retrieval status value of d_k in the list L_i, $c_{L_i}(d_k, d_j)$ indicates how much expert i (list L_i in this case) prefers d_k to d_j, $p(d_k, L_i)$ corresponds to the degree of preference of d_k by L_i, and finally, the total score indicates the general preference of d_k by all lists.

4.3 Evaluation Measure

The evaluation of results was carried out using a measure that has demonstrated its pertinence to compare IR systems, namely, the Mean Average Precision (MAP). It is defined as the norm of the average precisions (AveP) obtained for each query. The AveP for a given query is calculated as follows:

$$AveP = \frac{\sum_{r=1}^{m} P(r) \times rel(r)}{n} \tag{7}$$

where $P(r)$ is the precision at the first r documents, $rel(r)$ is a binary function which indicates if document at position r is relevant or not for the query; n is the number of relevant documents for the query that exist at the entire document collection; and m is the number of relevant documents retrieved. In all the experiments, we computed the MAP taking into account the first 1000 retrieved documents.

In addition, in all experiments we evaluated the statistical significance of results by means of the *paired student's t-test* considering a confidence level $\alpha = 0.05$, which is extendedly used in IR tasks [19].

5 Experimental Results

5.1 Baseline Results

As we previously mentioned, the traditional DF approach consists in combining all available results lists obtained for a specific query without considering any information about them. Based on this fact, Table 2 presents the MAP results corresponding to the combination of the entire set of five result lists per query, using three different DF methods and

four different data sets. In general, these results indicate that methods taking advantage of the document's redundancies, such as CombMNZ and Fuzzy Borda, are more robust than the ones based on information complementarities, such as MaxRSV.

In addition, the last row of Table 2 shows the average performance rate (i.e., the average MAP results) from the five individual IR methods considered for fusion. The comparison of these results against those from DF reveals that in many cases, but not all, DF results are higher. This is an important result since it indicates that in a real IR scenario, where there is no a priori information about the available IR methods, it is a better alternative to apply a DF method, particularly the CombMNZ method, than randomly select one IR method.

Table 2. Baseline results obtained by combining all results lists

DF Method	Ad hoc 2005	GeoCLEF 2008	ImageCLEF 2008	RobustCLEF 2008
MaxRSV	0.231	0.180	**0.251**	0.231
CombMNZ	**0.275**	**0.244**	**0.302**	**0.341**
Fuzzy Borda	**0.267**	**0.251**	**0.321**	0.167
IR systems *Average Performance*	0.250	0.233	0.238	0.265

5.2 Results of the Proposed Approach

The proposal of this paper is the combination of only the n-top result lists per query. Therefore, our experiments were designed to confirm the hypothesis that the combination of only the presumably n-best list per query (determined by a proposed heuristic quality measure) allows achieving better results than the combination of all available data. In order to carry out these experiments we proceed as follows:

1. Calculate the quality value (Q) for each one of the given result lists as described by Formula 2.
2. Select the set of n list having the greatest quality values. In particular, given that our interest was to combine the selected set of lists, we considered the following cases: $2 \leq n < |R|$.
3. Perform the DF process using the three methods, namely, Maximum RSV, CombMNZ and Fuzzy Borda.

The results from these experiments are shown in Tables 3 to 5. These tables also include in the last row the baseline results obtained by the combination of all lists (traditional DF approach). In them, the numbers in bold indicate that our method could outperform the baseline results, and the asterisks (*) next to the MAP scores indicate that the achieved improvement was statistically significant.

Table 3. Data fusion results using the n-top lists and the Maximum RSV method

Number of selected lists	Ad hoc 2005	GeoCLEF 2008	ImageCLEF 2008	RobustCLEF 2008
$n = 2$	**0.245***	**0.214**	**0.310***	**0.288***
$n = 3$	0.229	**0.188**	**0.303***	**0.263***
$n = 4$	0.225	0.177	**0.287***	**0.246***
Combining all lists	0.231	0.180	0.251	0.231

Table 4. Data fusion results using the *n*-top lists and the CombMNZ method

Number of selected lists	Ad hoc 2005	GeoCLEF 2008	ImageCLEF 2008	RobustCLEF 2008
$n = 2$	**0.300***	0.233	**0.333***	0.334
$n = 3$	**0.281**	**0.274***	**0.340***	0.328
$n = 4$	0.274	**0.261***	**0.323***	0.324
Combining all lists	0.275	0.244	0.302	0.341

Table 5. Data fusion results using the *n*-top lists and the Fuzzy Borda method

Number of selected lists	Ad hoc 2005	GeoCLEF 2008	ImageCLEF 2008	RobustCLEF 2008
$n = 2$	**0.295***	**0.266**	**0.341***	**0.271***
$n = 3$	**0.285***	**0.288***	**0.345***	**0.261***
$n = 4$	**0.278***	**0.286***	**0.335**	**0.223***
Combining all lists	0.267	0.251	0.321	0.167

In general, we consider that these results are encouraging, because they show that in all cases, except one configuration, the proposed method could outperform the baseline results. This behavior was particularly clear for the ImageCLEF data set, where we obtained very good results using all DF methods and considering any number of lists. We believe this was because this dataset contains more relevant documents per query (60 as showed in Table 1) than the other three collections.

On the other hand, we cannot formulate a definitive conclusion about the adequate value of *n*, since its selection depends on the used DF method and on the characteristics of the target document collection. However, from the results, it is possible to observe that $n = 3$ and $n = 2$ tended to generate the best results, indicating somehow that it is better to select the best lists than eliminate the worst(s).

6 Conclusions and Future Work

This paper proposed a new DF approach based on the combination of only the *n*-top result lists per query as an alternative to the fusion of all available data. The selection of the top result lists relies on an unsupervised quality measure that uses information about the redundancy and ranking of the documents across the lists. This approach differs from previous proposals in that it does not depend on any a priori knowledge about the IR methods nor on the user relevance judgments.

The evaluation results in four IR test collections, considering a total of 266 queries, and employing three different DF methods are encouraging. They indicate that the proposed approach could significantly outperform the results achieved by fusion all result lists, considering the MAP scores. They also show that the approach may be successfully used in conjunction with several DF methods given that it could achieve average improvements of 10.7%, 3.7% and 18.8% when was used along with Maximum RSV, CombMNZ and Fuzzy Borda respectively. In addition, we could observe relevant results with several data sets of different characteristics, obtaining average improvements over the baseline of 3.9%, 7.9%, 11.8% and 20.7% for the Ad-hoc, GeoCLEF, ImageCLEF and RobustCLEF collections.

Finally, supported by the presented experimental results, we plan to focus our future work in two main issues. On the one hand, the selection of the most appropriate DF method for a given set of lists and, on the other hand, the dynamic choice of the value of n (number of lists to be combined) based on the redundancy and complementarity characteristics of the given result lists.

Acknowledgments. This work was done under partial support of CONACYT (project grants 83459 and 82050, and scholarship 165499). We would also like to thank the CLEF organizing committee for the resources provided.

References

1. Agirre, E., Di Nunzio, G.M., Ferro, N., Mandl, T., Peters, C.: CLEF 2008: Ad Hoc Track Overview. In: Working Notes for the CLEF 2008 Workshop, Aarhus, Denmark (2008)
2. Arni, T., Clough, P., Sanderson, M., Grubinger, M.: Overview of the ImageCLEFphoto 2008 Photographic Retrieval Task. In: Working Notes for the CLEF 2008 Workshop, Aarhus, Denmark (2008)
3. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley, Reading (1999)
4. Bartell, B.T., Cottrell, G.W., Belew, R.K.: Automatic Combination of Multiple Ranked Retrieval Systems. In: Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, Dublin, Ireland (1994)
5. Diamond, T., Liddy, E.D.: Dynamic data fusion. In: Proceedings of the TIPSTER Text Program: Phase III. Annual Meeting of the Association for Computational Linguistics (ACL), Baltimore, Maryland, USA (1998)
6. Di Nunzio, G.M., Ferro, N., Jones, G.J.F., Peters, C.: CLEF 2005: Ad Hoc Track Overview. In: Working Notes for the CLEF 2005 Workshop, Vienna, Austria (2005)
7. Escalante, H.J., González, J.A., Hernández, C.A., López, A., Montes, M., Morales, E., Sucar, L.E., Villaseñor, L.: TIA-INAOE's Participation at Image CLEF 2008. In: Working Notes for the CLEF 2008 Workshop, Aarhus, Denmark (2008)
8. Fox, E.A., Shaw, J.A.: Combination of Multiple Searches. In: Proceedings of The Second Text REtrieval Conference (TREC-2), Gaithersburg, Maryland, USA (1994)
9. Gopalan, N.P., Batri, K.: Adaptive Selection of Top-m Retrieval Strategies for Data Fusion in Information Retrieval. International Journal of Soft Computing 2(1), 11–16 (2007)
10. Hsu, D.F., Taksa, I.: Comparing Rank and Score Combination Methods for Data Fusion in Information Retrieval. Information Retrieval 8(3), 449–480 (2005)
11. Kantor, P.B.: Decision level data fusión for routing of documents in the TREC3 context: A best case analysis of worst case results. In: Proceedings of The Third Text REtrieval Conference (TREC-3), Gaithersburg, Maryland, USA (1995)
12. Lebanon, G., Lafferty, J.: Cranking: Combining rankings using conditional probability models on permutations. In: Proceedings of the Nineteenth International Conference on Machine Learning, Sydney, Australia (2002)
13. Lee, J.H.: Analyses of Multiple Evidence Combination. In: Proceedings of the 20th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, Philadelphia, PA, USA (1997)

14. Lillis, D., Toolan, F., Collier, R., Dunnion, J.: A probabilistic approach to data fusion. In: Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, Washington, USA (2006)
15. Mandl, T., Carvalho, P., Gey, F., Larson, R., Santos, D., Womser-Hacker, C.: GeoCLEF 2008: the CLEF 2008 Cross-Language Geographic Information Retrieval Track Overview. In: Working Notes for the CLEF 2008 Workshop, Aarhus, Denmark (2008)
16. Montague, M., Aslam, J.A.: Condorcet fusion for improved retrieval. In: Proceedings of the 11th International Conference on Information Knowledge and Management (CIKM-ACM), McLean, VA, USA (2002)
17. Ng, K.B., Kantor, P.B.: Predicting the effectiveness of naive data fusion on the basis of system characteristics. Journal of American Society for Information Science 51, 1177–1189 (2000)
18. Perea, J.M., Ureña, L.A., Buscaldi, D., Rosso, P.: TextMESS at GeoCLEF 2008: Result Merging with Fuzzy Borda Ranking. In: Working Notes for the CLEF 2008 Workshop, Aarhus, Denmark (2008)
19. Smucker, M.D., Allan, J., Carterette, B.: Agreement Among Statistical Significance Tests for Information Retrieval Evaluation at Varying Sample Sizes. Poster session for The 32nd Annual ACM SIGIR Conference (SIGIR 2009), Boston, MA, USA (2009)
20. Villatoro-Tello, E., Montes-y-Gómez, M., Villaseñor-Pineda, L.: INAOE at GeoCLEF 2008: A Ranking Approach based on Sample Documents. In: Working Notes for the CLEF 2008 Workshop, Aarhus, Denmark (2008)
21. Vogt, C., Cottrell, G., Belew, R., Bartell, B.: Using relevance to train a linear mixture of experts. In: Proceedings of The Fifth Text REtrieval Conference (TREC-6), Gaithersburg, Maryland (1997)
22. Vogt, C.C., Cottrell, G.W.: Predicting the performance of linearly combined IR systems. In: Proceedings of the 21st ACM-SIGIR International Conference on Research and Development in Information Retrieval, Melbourne, Australia (1998)
23. Vorhees, E.M.: Overview of TREC 2007. In: Proceedings of the sixteenth Text Retrieval Conference (TREC 2007), Gaithersburg, Maryland, USA (2007)
24. Wu, S., McClean, S.: Performance prediction of data fusion for information retrieval. Information Processing and Management 42(4), 899–915 (2006)

Multi Word Term Queries for Focused Information Retrieval

Eric SanJuan[1] and Fidelia Ibekwe-SanJuan[2]

[1] LIA & IUT STID, Université d'Avignon
339, chemin des Meinajaries, Agroparc BP 1228,
84911 Avignon Cedex 9, France
eric.sanjuan@univ-avignon.fr
[2] ELICO, Université de Lyon 3
4, Cours Albert Thomas, 69008 Lyon, France
ibekwe@univ-lyon3.fr

Abstract. In this paper, we address both standard and focused retrieval tasks based on comprehensible language models and interactive query expansion (IQE). Query topics are expanded using an initial set of Multi Word Terms (MWTs) selected from top n ranked documents. MWTs are special text units that represent domain concepts and objects. As such, they can better represent query topics than ordinary phrases or n-grams. We tested different query representations: bag-of-words, phrases, flat list of MWTs, subsets of MWTs. We also combined the initial set of MWTs obtained in an IQE process with automatic query expansion (AQE) using language models and smoothing mechanism. We chose as baseline the Indri IR engine based on the language model using Dirichlet smoothing. The experiment is carried out on two benchmarks: TREC Enterprise track (TRECent) 2007 and 2008 collections; INEX 2008 Ad-hoc track using the Wikipedia collection.

1 Introduction

Previous experiments carried out within the framework of TREC [1] tended to conclude that retrieval performance has not been enhanced by adding NLP, especially syntactic level of processing. The problem lies in determining the level of NLP needed, on which text units to implement it, whether to implement NLP on both queries and documents and at what stage (whole collection or only on an initial set of returned documents). Previous research also concluded that a deep syntactic representation of queries and documents is not useful to achieve a state-of-the-art performance in IR [2]. It may on the contrary degrade results. On the other hand, performance can be boosted by better representing queries and documents with longer phrases using shallow NLP. In some cases, even a well-tuned n-gram approach can approximate the extraction of phrases and may suffice to boost retrieval performance.

Up until 2004, the dominant model in IR remained the bag-of-words representation of documents which continued to show superior performances in IR.

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 590–601, 2010.

However, a series of experiments carried out on several document collections over the past years are beginning to show a different picture. Nothwithstanding the apparent success of the bag-of-word representation in some IR tasks, it is becoming clear that certain factors related mostly to query length and document genre (general vs technical) influence the performance of IR systems. For instance, [1,3] showed that representing queries and document by longer phrases can improve systems' performances since these text units are inherently more precise and will better disambiguate the information need expressed in the queries than lone words.

Furthermore, [1] concluded that the issue of whether or not to use NLP and longer phrases would yield better results if focused on query representation rather than on the documents themselves because no matter how rich and elaborate the document representation, a poor representation of the information need (short queries of 1-2 words) will ultimately lead to poor retrieval performance.

Based on these earlier findings, we wish to investigate the issue of representing queries with a particular type of phrase which are Multiword Terms (MWTs). MWTs is understood here in the sense defined in computational terminology [4] as textual denominations of concepts and objects in a specialized field. Terms are linguistic units (words or phrases) which taken out of context, refer to existing concepts or objects of a given field. As such, they come from a specialized terminology or vocabulary [5]. MWTs are thus terms of length >1. MWTs, alongside noun phrases, have the potential of disambiguating the meaning of the query terms out of context better than single word terms or statistically-derived n-grams and text spans. In this sense, MWTs cannot be reduced to words or word sequences that are not linguistically and terminologically grounded. An initial selection of MWTs from queries is used in an Interactive Query Expansion (IQE) process to acquire more MWTs from top n-ranked documents. The expanded set is submitted to standard IR Language Models for document ranking. Our approach is tested on two corpora: the TREC Enterprise track 2007 and 2008 collections, and INEX 2008 Ad-hoc track. We chose as baseline against which to compare our IQE approach, an IR engine based on the language model using Dirichlet smoothing. The Indri IR system [6] in its default mode applies this language model. Indri was also used as baseline in TREC terabyte[1]. The idea was to test our IQE approach against a strong baseline that competes favorably with the best systems in current IR evaluation campaigns. The results obtained on the Wikipedia corpus in the INEX Ad-hoc track are particulary promising.

The rest of the paper is structured as follows. Section §2 presents our language model and its application to the IR tasks. Section §3 describes the application of our IR model to the TREC Enterprise track 2007 and 2008 collections for document search task. Section §4 presents the focused retrieval tasks on the Wikipedia collection in the INEX 2008 Ad-hoc track. Finally, section §5 discusses lessons learned from these experiments.

[1] `http://stefan.buettcher.org/trec-tb/`

2 Combining Automatic and Interactive Query Expansion

2.1 Language Model

Language models are widely used in NLP and IR applications.In the case of IR, smoothing methods play a fundamental role [7]. We shall first describe the probability model that we use.

Document Representation: probabilistic space and smoothing. Let us consider a finite collection $\mathcal{D}$ of documents, each document D being considered as a sequence $(D_1, ..., D_{|D|})$ of $|D|$ terms D_i from a language $\mathcal{L}$, i.e. $\mathcal{D}$ is an element of $\mathcal{L}^\star$, the set of all finite sequences of elements in $\mathcal{L}$. Our formal framework is the following probabilistic space $(\Omega, \wp(\Omega), P)$ where Ω is the set of all occurrences of terms from $\mathcal{L}$ in some document $D \in \mathcal{D}$ and P is the uniform distribution over Ω LMs for IR rely on the estimation of the a priori probability $P_D(q)$ of finding a term $q \in \mathcal{L}$ in a document $D \in \mathcal{D}$. We chose the Dirichlet smoothing method because it can be viewed as a maximum *a priori* (MAP) document probability distribution. Given an integer μ, it is defined as:

$$P_D(q) = \frac{f_{q,D} + \mu \times P(q)}{|D| + \mu} \tag{1}$$

Query Representation and ranking functions. Our purpose is to test the efficiency of MWTs in standard and focused retrieval compared to a classic bag-of-word model and statistically-derived phrases. For that, we shall consider phrases (instead of single terms) and a simple way of combining them. Given a phrase $s = (s_0, ..., s_n)$ and an integer k, we formally define the probability of finding the sequence s in the corpus with at most k insertions of terms in the following way. For any document D and integer k, we denote by $[s]_{D,k}$ the subset of $D_i \in D$ such that: $D_i = s_1$ and there exists n integers $i < x_1, ..., x_n \leq i+n+k$ such that for each $1 \leq j \leq n$ we have $s_j = D_{x_j}$.

We can now easily extend the definition of probabilities P and P_D to phrases s by setting $P(s) = P([s]_{.,k})$ and $P_D(s) = P_D([s]_{D,k})$. Now, to consider queries that are set of phrases, we simply combine them using a weighted geometric mean for some sequence $w = (w_1, ..., w_n)$ of positive reals. Unless stated otherwise, we shall suppose that $w = (1, ..., 1)$, i.e. the normal gometric mean. Therefore, given a sequence of weighted phrases $Q = \{(s_1, w_1), ..., (s_n, w_n)\}$ as query, we shall rank documents acording to the following scoring function $\Delta_Q(D)$ defined by:

$$\Delta_Q(D) \stackrel{\text{rank}}{=} \sum_{i=1}^{n} \left(\frac{w_i}{\sum_{j=1}^{n} w_j} \times \log(P_D(s_i)) \right) \tag{2}$$

This plain document ranking can easily be computed using any passage information retrevial engine. We chose for this purpose the Indri engine since it combines a language model (LM) with a bayesian network approach which can handle complex queries.

2.2 Query Expansion

We propose a simple QE process starting with an approximative short query $Q_{T,\mathcal{S}}$ of the form $(T, \mathcal{S})$ where $T = (t_1, ..., t_k)$ is an approximative document title consisting of a sequence of k words, followed by a possibly empty family of sets of phrases: $\mathcal{S} = \{S_1, ..., S_{|S|}\}$ where for each $1 \leq i \leq |S|$, S_i is of the form $\{S_{i,1}, ..., S_{i,l_i}\}$ for some $l_i \geq 0$. If $l_i = 0$ then S_i is considered to be the empty set. In our case, each $S_{i,j}$ will be a MWT.

Baseline document ranking function. By default, we shall rank documents acording to $\Delta_{T,\mathcal{S}} = \Delta_T \times \prod_{i=1}^{|\mathcal{S}|} \prod_{j=1}^{|l_i|} \Delta_{S_{i,j}}$. Therefore, the larger $\mathcal{S}$ is, the less the title part T is taken into account. Indeed, $\mathcal{S}$ consists of coherent subsets of MWTs defined by the user. If the user can expand the query by finding coherent clusters of terms, then we are no more in the situtation of a vague information need and documents should be first ranked according to precise MWTs. For our baseline, we shall generally consider $\mathcal{S}$ to be empty or made of phrases automatically generated from T.

Interactive Multiword Term Selection. The IQE process works in the following manner. We consider the top twenty ranked documents of Δ_Q ranking. The user selects a family $\mathcal{S}'$ of several subsets $S'_1, ..., S'_s$ of MWTs appearing in these documents. This leads to acquiring sets of synonyms, abbreviations, hypernyms, hyponyms and associated terms with which to expand the original query terms. We also let the user check that these terms do not introduce noise by adding them individually to the initial query and observing the top ranked documents. The selected multiword terms S'_i are added to the initial set $\mathcal{S}$ to form a new query $Q' = Q_{T,\mathcal{S}\cup\mathcal{S}'}$ leading to a new ranking $\Delta_{Q'}$ computed as previously in §2.2. We emphasize that $\mathcal{S}'$ is more than a flat list of MWTs. In our experiments we also evaluate if the structure of $\mathcal{S}'$ (i.e., grouping the MWTs into subsets) is relevant or not.

Automatic Query expansion. We also experimented with the automatic query expansion (AQE). In our model, it consists in the following. Let $D_1, ..., D_K$ be the top ranked documents by the initial query Q. Let $C = \cup_{i=1}^{K} D_i$ be the concatenation of these K top ranked documents. Terms c occuring in D can be ranked according to $P_C(c)$ as defined by equation (1). We consider the set E of the N terms $\{c_1, ..., c_N\}$ having the highest probability $P_C(c_i)$. We then consider the new ranking function Δ'_Q defined by $\Delta'_Q = \Delta_Q^{\lambda} \times \Delta_E^{1-\lambda}$ where $\lambda \in [0, 1]$.

Unless stated otherwise we shall take $K = 4$, $N = 50$ and $\lambda = 0.1$. We now explore in which context IQE based on MWTs is efficient. Our baseline is automatic document retrieval based on equation 2 in §2.1.

3 Enterprise Search

The goal of the TREC enterprise track (TrecEnt) was "*to conduct experiments with enterprise data that reflect the experiences of users in real organizations*" [8].

This track ran from 2004 to 2008. We participated in the 2008 edition but "trained" our search strategies beforehand on the 2007 data. Hence, we will indicate performances obtained on data from both years.

3.1 Document Collection and Tasks

In 2007, the TrecEnt track chose the CSIRO Enterprise Research Collection (CERC) which is a crawl of all the `*.csiro.au` public websites performed in march 2007[2]. The collection consists of 370,715 documents totaling 4.2 gigabytes. The search topics used in the TrecEnt tasks were furnished by employees of CSIRO in charge of science communication. These topics correspond to real world information needs received by the CSIRO staff from the public. Thus participating IR systems were judged on real life information needs and not on artificially contrived queries. The submitted runs were evaluated by the community based on the final answer furnished by CSIRO staff to the original requester. An example of a topic from TrecEnt 2008 is *Weatherwall* with the following narrative: "*Have been trying to access the CSIRO weatherwall site to check on weather in Melbourne over the last 24 hours. It seems to be off line at present. Any idea why? When might it be back on line?*"

We designed four basic search strategies, called "runs" in the TREC terminology. These four runs were applied on the 2007 and 2008 TrecEnt collections as well on the INEX Ad-hoc tasks albeit with some variations. The first run is the baseline defined in §2.2 using only the query fields. The second is a boosting of this baseline by simply repeating queries in the $\mathcal{S}$ component as phrases. Clearly, instead of leaving $\mathcal{S}$ empty,$\mathcal{S}$ is the singleton $\{\{q\}\}$ made of the query phrase q. The last two runs are based on the IQE process described in 2.2. We give below the precise details of each run:

- **baseline bag-of-words (baseline-B)**: we set $T = \{q_1, ..., q_n\}$ where the q_i are the terms in topic query field q. $\mathcal{S}$ is left empty. This is the usual multinomial bag-of-word approach.
- **baseline phrases (baseline-P)**: we keep the same T but $\mathcal{S}$ is set to the singleton $\{\{(q_1, ..., q_n)\}\}$ whenever the query contains at least two words, i.e. in addition to the bag-of-words approach, we also consider the query q as a phrase.
- **IQE MWT-groupings (IQE-C)**: this run corresponds to the IQE approach described in §2.2 except that the user creates sub-groups of MWTs, hence providing a hierarchy of sorts among MWTs. We set $\mathcal{S}$ to $\mathcal{S}(t)$ for each topic. The T component is unchanged.
- **IQE MWTs flat list (IQE-L)**: we consider as $\mathcal{S}$ a flat version of each $\mathcal{S}_t$ where all the selected MWTs are considered at the same level, the internal structure of $\mathcal{S}(t)$ is ignored.

The **IQE − L** run evaluates the impact of $MWTs$ on document ranking while the **IQE-C** run, also based on MWTs, evaluates the impact on the retrieval

[2] The Australian 'Commonwealth Scientific and Industrial Research Organization'.

effectiveness of forming subsets of *MWTs* by the user. We illustrate these two representations of MWTs on the same topic.For the **IQE-C run**, the user formed these subsets of MWT queries:

1. {weatherwall}
2. {(weatherwall site), weather, Melbourne}
3. {(CSIRO weatherwall site), weatherwall, (weather in Melbourne)}

In this representation, the particular angle by which the MWT is sought is reflected by a facet term placed to the right of it, e.g. *((weatherwall site), weather, Melbourne))*. In the **IQE-L run**, the expanded query is represented by this flat list of MWTs: *((weatherwall site), (CSIRO weatherwall site), (weather in Melbourne), weatherwall, weather, Melbourne*). This is a simplified version of the same MWTs used in the IQE-C run in which the facet terms have been removed. All terms are weighted equally here.

3.2 Results Based on Usual Average Precision

The official measure for the TrecEnt 2007 edition was Average Precision (AP). This was changed to *inferred* Average Precision (*inf*AP) for TRECEnt 2008. However, we can compute AP on both tracks.

Document search on the TrecEnt 2007 collection. 50 topics were provided and all were judged. On the resulting document qrels, our baseline reaches a mean average precision (MAP) of 0.441 which outperforms all reported runs in [8], the highest MAP being 0.422. However, based on the query by query average precision (AP) score, there is no statistical evidence (t-test with a 95% confidence interval) that our baseline has a true mean not equal to 0.422. Since TrecEnt queries were short phrases most of which had the appearance of MWTs like *"solve magazine, selenium soil"*, the question was to ascertain if our baseline can be boosted by considering phases as suggested by [3]. It seems the answer is yes, but only slightly since the *phrases* run reaches the MAP score of 0.448.

Document search on the TrecEnt 2008 collection. 77 topics were made available to participants of which 67 were judged. Four had no judged relevant documents and were dropped. The same IQE process was implemented in which a user selected for each topic t, subsets $\mathcal{S}(t)$ of MWTs following the methodology described in §2.2.

We first computed the AP measures used in TrecEnt 2007 in other to compare our baseline to its performance on this data. Confirming its good perfomance in 2007, our *baseline-B* run implementing the bag-of-word approach outperformed all our other approaches. The good performance of our *baseline-B* here confirms that it is indeed a strong one since it reaches similar precision scores at 10% of recall and even higher at 20% of recall. The 2008 curves then drop because TrecEnt 2008 qrels are based on a more complex pooling process that handicaps low ranked documents in participant runs. In fact, it appears that our two baseline runs ranked first the "easiest to find" relevant documents among these qrels. These are documents found by most participants.

3.3 Results Based on Inferred Average Precision

The inferred AP (infAP) measure used in TRECEnt 2008 is similar to the original infAP used in the TREC Terabyte track, except that it has been modified to work on stratified samples. Both versions of infAP take into account the fact that the measurement is based on a pool of relevant documents and not on an exhaustive list of all relevant documents. Indeed, AP relies on the knowledge of the complete set of relevant documents which on a large corpus is not generally known. According to NIST organizers of the TrecEnt 2008, *"two runs were pooled out from each group to depth 100. The documents were selected for judging by taking a stratified sample of that pool based on document ranks: documents retrieved at ranks 1-3 were sampled at 100% depth, documents of ranks 4-25 at depth 20%, and document between 25-75 rank were sampled at 10% depth. The rank of a document for sampling purposes is the highest rank over all pooled runs. "* The evaluation script and relevance judgments are available from the TREC website[3]. The script also allows us to estimate the usual Normalized Discounted Cumulated Gain (NDCG) that gives more importance to elements at higher ranks. Figure 1 shows the inferred AP and NDCG of our baseline and IQE runs.

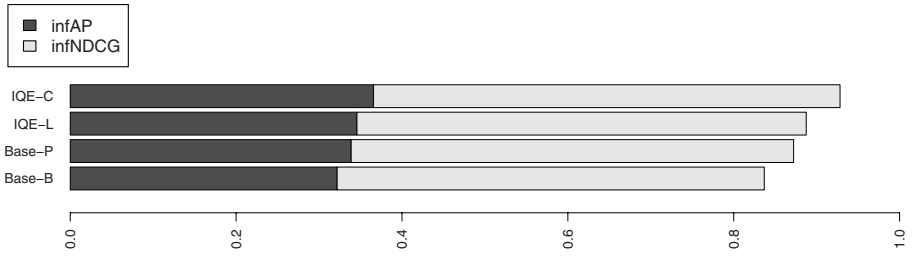


Fig. 1. Inferred Average Precision and Normalized Discounted Cumulated Gain on TrecEnt 2008 qrels using available sampling information

On the resulting 2008 stratified qrels, our *baseline-B* run attains an infAP score of 0.3218 thus placing itself among the six best runs submitted to TrecEnt 2008. In contrast with previous results on absolute AP, the infAP goes up to 0.3387 when considering phrases in *baseline-P* run, 0.345 when considering *IQE-L* run based on the flat list of additional terms and 0.3657 for *IQE-C* run using the grouped set $\mathcal{S}(t)$ of MWTs. Therefore, using the infAP measure, our IQE-MWTs runs outperform the baseline bag-of-word and phrase runs.

However, only the difference between the first *baseline-B* and other runs is statistically significant (t-test at 95% of confidence). Other differences are not significant. Since the *baseline-P* run is in fact the *baseline-B* boosted by adding the whole topic query as a phrase to the initial bag of words query, these results show that [3]'s observations that document retrieval performance can be

[3] `http://http://trec.nist.gov/data/t17_enterprise.html/`

boosted on large web collections by considering phrases, are also true on smaller enterprise web corpus.

4 Focused Retrieval

The focused retrieval experiment was carried out in the framework of INEX 2008 Ad-hoc track which is the main forum for researchers working on the extraction of information from structured documents, mostly XML [9].

4.1 INEX 2008 Ad-Hoc Track

Corpus and topics. The official INEX 2008 corpus was the 2006 version of the English Wikipedia comprising 659,388 articles without images [10]. On average, an article contains 161 XML nodes, where the average depth of a node in the XML tree of the document is 6.72. From this corpus, participants were asked to submit query topics corresponding to real life information needs. A total of 135 such topics were built, numbered from 544-678. 70 out of them were judged by the community and thus used in the official evaluation. A topic consists of four fields: content only field (<CO> or <Title>) with a multi-word term expression of the topic; a content only + structure version of the topic (<CAS>) which is the title with indication of XML structure where the relevant elements may be found; a <description> field which is a slightly longer version of the title field; and a <narrative> field comprising a summary with more details about the expected answers.

Ad-Hoc Retrieval Tasks. The 2008 Ad-Hoc track had 3 tasks: Focused retrieval, Relevant-in-Context (RiC), Best-in-Context (BiC).

1. The focused task requires systems to return a ranked list of relevant non-overlapping elements or passages. This is called the "fetching phase".
2. The Relevant-in-Context (RiC) task builds on the results of the focused task. This task is based on the assumption that a relevant article will likely contain relevant information that could be spread across different elements. This is called the "browsing phase". Systems are therefore asked to select, within relevant articles, several non-overlapping elements or passages that are specifically relevant to the topic.
3. The Best-in-Context (BiC) task is aimed at identifying the best entry point (BEP) to start reading a relevant article. This task is based on the assumption that "even an article completely devoted to the topic of request will only have one best starting point from which to read (even if that is the beginning of the article)" [11].

Extended qrels and evaluation measures. The evaluation procedure establishes an extended qrel file similar to those used in TREC against which all participating systems are evaluated. Like in TREC Terabyte and Ad-hoc tracks,

the procedure consists in selecting for each query a pool of documents from participant runs. Topics and documents are then randomly distributed to assessors from the INEX community. Using an ergonomic java on-line interface, each assessor has to mark-up for each document, the relevant passages with regard to a topic. It is important to emphasize that query terms are highlighted in the display of documents. Moreover, in 2008, the interface offered the facility of selecting the whole document using a simple radio button. The assessor had also to point out the BEP. These result in a qrel file that gives for each evaluated pair of topic and document, the total length of relevant passages, the document length, the offset of the BEP and the list of relevant passages. Lengths are computed as number of characters in the text version of the corpus (without XML tags). The 2008 qrel file required the evaluation of 36, 605 articles. Among them, only 4, 773 were judged to contain at least one relevant passage for at least one topic. However, it appears that 40% of these 4, 773 documents have at least 95% of their content marked as relevant by assessors. These highly relevant documents only cover 0.02% of the total length of evaluated documents but almost 25% of the total length of relevant passages. These facts are important to estimate the upper AP bound for systems retrieving full document instead of passages or XML elements.

The RiC and BiC are also evaluated based on these qrels but using graded document scores whereas in the focused task, scores are based on the sole relevant passages no matter their co-occurrence in documents. Given a document score function S into $[0, 1]$, both RiC and BiC evaluations are based on generalized precision gP at some rank r which is the average score S over the r scores documents. Given a document d, the score $S(d)$ is in the case of:

- RiC, the F-score of the retrieved passages from d by the system among all relevant passages in d.
- BiC, a normalized distance in number of caracters between the BEP found by the system and the real one.

The consequence is that these measures favour even more full document retrival strategies against passage retrieval since for 40% of relevant documents, full document retrieval strategies will obtain the maximal score whenever they retrieve relevant documents. We refer to [11] forfurther discussion of these measures.

4.2 Results

We first present our search strategies, then analyze results by tasks in the INEX Ad-hoc track.

Runs. We consider the same four basic strategies as in the TREC Enterprise search track: *baseline bag-of-words (baseline-B), baseline phrases (baseline-P), IQE MWTs subsets (IQE-C)* and *IQE MWT flat list (IQE-L)*. Like in the TrecEnt experiment, the two first runs are automatic, the last two rely on the sets of MWTs manually gathered when browsing the top ranked 20 documents based on an initial query. Table 1 gives an example of such expansion.

Table 1. Selected multiword terms for the INEX 2008 topic "dna testing forensic maternity paternity"

IQE-LC with subsets of MWTs	resulting flat list for IQE-C
{(dna testing) disease}	(dna testing)
{(dna testing ancestry)}	(dna testing ancestry)
{(genetic disease), (dna testing) ancestry}	(genetic disease)
{(hereditary disease) (dna testing) ancestry}	(hereditary disease)

Compared to the TrecEnt runs, there are two differences in the way that we apply these runs here: 1) we do not use any stemmer, nor lemmatization and we index all the text (no stop word list), 2) we systematically apply AQE to all runs.

Indeed, Wikipedia articles are well written, with very few spelling errors, thus any stemming will induce a loss of information whereas on the CSIRO web pages, stemming tended to reduce the noise. AQE on the non lemmatized Wikipedia corpus was able to automatically capture synonyms and some grammatical variants of the query term. On the CSIRO corpus used in TrecEnt, AQE just added more noise.

Focused task. The INEX 2008 official measure for focused task was average interpolated Precision at 1% of recall (iP[0.01]). Figure 2 shows the Recall/Precision curves of our baseline and IQE runs. The best score for all runs in the official evaluation was 0.6896. Our *baseline-B* score (automatic run with AQE) obtains a significantly much lower score at 0.5737. The *baseline-P* run did not benefit from the same boosting effect as in TRECEnt experiment, hence its much lower score of 0.5732. The *IQE-L* run obtained a much higher score of 0.7016, even higher than the best participating system. This score is further improved to 0.7137 when we consider the *IQE-C* run in which MWTs had been grouped to reflect more complex query representations (see table 1 for an example).

The differences between IQE-based runs are not statistically significant, whereas the difference between *baseline* runs and the IQE runs is this time clearly significant. Indeed, using the Welch Two Sample paired t-test, we find a p-value of 0.02302. Moreover, other participants' best runs submitted at INEX 2008 were optimal for very low recall values but then drop down fast for higher recall values. One might put forward the argument that the good score of our IQE runs may be due to the fact that the user found one or two completely relevant documents with some specific MWTs which were then re-introduced in the expanded query. The Precision/Recall curves in Figure 2 show that this was not the case. In fact, mean average iP for the *baseline* runs is only 0.28 while that of both both IQE runs reach 0.34. The difference is again statistically significant at 95% of confidence with an estimated p-value of 0.03966. Therefore, this experiment clearly demonstrates that representing queries with MWTs corresponding to real concepts instead of n-grams or bag-of-words, can dramatically improve

IR when dealing with a high quality collection such as the Wikipedia. We now present results for the other two tasks of the Ad-hoc track.

Relevant-in-Context and Best-in-Context tasks. The official measure for these tasks was MAgP (Mean Average generalized Precision). By considering that we only retrieve articles that are completely relevant, and that the best entry point is the first character of the document, the same four runs can be evaluated with regard to the RiC and BiC measures.

Our runs maintained the same order as it can be observed in figure 2. Among all submitted runs to INEX 2008, the best score was 0.228 for RiC and 0.224 for BiC. Our *baseline* already reaches a score of 0.197 for RiC and 0.20 for BiC. This places our baseline among the six best runs and our group among the three best teams. The baseline is slightly improved by considering *phrases*: 0.2 for RiC, 0.206 for BiC. The scores of IQE outperform the best scores in the official evlauation. Indeed, the *IQE-L* run reaches a score of 0.236 for RiC and 0.248 for BiC. Surprisingly, *IQE-C* run does not improve these score since it obtains a score of 0.235 for RiC and 0.246 for BiC. However, none of these differences are statiscally significant at 95% of confidence, the Welch Two Sample t-test p-value between the *baseline* and the *IQE-L* runs being 0.08739 for RiC and 0.05981 for BiC. Classical MAP was also computed at INEX 2008 by considering as relevant any document involving at least one relevant passage, whatever its length. There, we also find that IQE runs also outperform all other runs, but the difference with the baseline is even less significant.

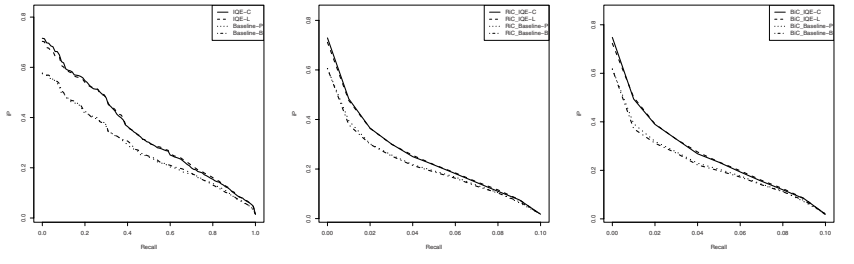


Fig. 2. Interpolated generalized precision curves on INEX 2008 topics for Focused (left) Relevant in Context (center) and Best in Context (right)

5 Conclusions

We have presented in this paper a methodology that relies on meaningful text units (multiword terms) to represent queries. These multiword terms are used alternatively with interactive query expansion and automatic query expansion, the two are also combined in order to determine the combination that best boosts retrieval effectiveness. The experimentation has been carried out on two different document collections: a web collection consisting of the CSIRO domain and

the Wikipedia corpus within TREC Enterprise track and INEX Ad-hoc track respectively. While the results obtained on the TrecEnt collection are not conclusive due perhaps to poor corpus quality and a change of evaluation measures in the TrecEnt campaigns, the results on the Wikipedia collection show that multi-word term query representation and interactive query expansion are a promising combination for both standard document and focused retrieval.

References

1. Perez-Carballo, J., Strzalkowski, T.: Natural language information retrieval: progress report. Information Processing and Management 36(1), 155–178 (2000)
2. Smeaton, A.F.: Using nlp and nlp resources for information retrieval tasks, pp. 99–109. Kluwer Academic Publishers, Dordrecht (1999)
3. Mishne, G., de Rijke, M.: Boosting web retrieval through query operations. In: Losada, D.E., Fernández-Luna, J.M. (eds.) ECIR 2005. LNCS, vol. 3408, pp. 502–516. Springer, Heidelberg (2006)
4. Kageura, K.: The dynamics of Terminology: A descriptive theory of term formation and terminological growth. John Benjamins, Amsterdam (2002)
5. Ibekwe-SanJuan, F.: Constructing and maintaining knowledge organization tools: a symbolic approach. Journal of Documentation 62, 229–250 (2006)
6. Metzler, D., Strohman, T., Turtle, H., Croft, W.B.: Indri at trec 2004: Terabyte track (2005); electronic proceedings only
7. Zhai, C., Lafferty, J.: A study of smoothing methods for language models applied to information retrieval. ACM Trans. Inf. Syst. 22(2), 179–214 (2004)
8. Bailey, P., de Vries, A.P., Craswell, N., Soboroff, I.: Overview of the trec 2007 enterprise track. In: Voorhees, E.M., Buckland, L.P. (eds.) TREC. Volume Special Publication 500–274, National Institute of Standards and Technology, NIST (2007)
9. Lalmas, M., Tombros, A.: Evaluating XML retrieval effectiveness at inex. SIGIR Forum 41(1), 40–57 (2007)
10. Denoyer, L., Gallinari, P.: The wikipedia XML corpus. In: SIGIR Forum p.6 (2006)
11. Kamps, J., Geva, S., Trotman, A., Woodley, A., Koolen, M.: Overview of the INEX 2008 ad hoc track. In: Geva, S., Kamps, J., Trotman, A. (eds.) INEX 2008. LNCS, vol. 5631, pp. 1–27. Springer, Heidelberg (2009)

Optimal IR: How Far Away?

Xiangdong An[1], Xiangji Huang[2], and Nick Cercone[1]

[1] Department of Computer Science and Engineering
York University, Toronto, ON M3J 1P3, Canada
xan@cs.yorku.ca, ncercone@yorku.ca
[2] School of Information Technology
York University, Toronto, ON M3J 1P3, Canada
jhuang@yorku.ca

Abstract. There exists a gap between what a human user wants in mind and what (s)he could get from the information retrieval (IR) systems by his/her queries. We say an IR system is *perfect* if it could always provide the users with what they want in their minds if available in corpus, and *optimal* if it could present to the users what it finds in an optimal way. In this paper, we empirically study how far away we are still from the optimal IR or the perfect IR based on submitted runs to TREC Genomics track 2007. We assume perfect IR would always achieve a score of 100% for given evaluation methods. The optimal IR is simulated by optimized runs based on the evaluation methods provided by TREC. Then the average performance difference between submitted runs and the perfect or optimal runs can be obtained. Given annual average performance improvement made by reranking from literature, we figure out how far away we are from the optimal or the perfect IRs. The study indicates we are about 7 and 16 years away from the optimal and the perfect IRs, respectively. These are absolutely not exact distances, but they do give us a partial perspective regarding where we are in the IR development path. This study also provides us with the lowest upper bound on IR performance improvement by reranking.

1 Introduction

An IR system automatically finds the information that matches the information needs of users expressed through their queries. We say an IR system is *perfect* if it could always find the information, if available in corpus, that matches the information needs of the users, and *optimal* if it could always present to the users what it finds in an optimal way (w.r.t. the relevancy). A critical difference between a perfect IR system and an optimal IR system is that an optimal system may not find all relevant information and may present irrelevant information. A perfect system involves much more sophisticated techniques than an optimal system. To be perfect, an IR system needs to understand natural languages well since natural languages are generally used to write queries expressing users' needs in their minds, and based on such understanding, the system needs to find *all* relevant information *exactly* and presents them in an optimal way. To be optimal, an IR system only needs to present the results in an optimal way. We say the retrieval results are presented in an optimal way if the results are ranked properly based on their relevancy to the query.

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 602–613, 2010.

None of the currently existing IR systems could be considered optimal in general. An existing system could be made closer to optimal by re-ranking. In this paper, we empirically study how far away we are from the optimal and the perfect IRs based on submitted runs to TREC (Text REtrieval Conference) Genomics track. We assume perfect IRs always achieve a performance of 100% for given evaluation methods. We simulate optimal IRs with optimized runs over TREC evaluation. Then, the performance difference between submitted runs and the optimal runs or perfect IRs can be calculated. Based on annual average performance improvement made by re-ranking from literature, we can figure out how far away we are from optimal or perfect IRs. The study might give us some ideas about where we are in the IR development path and a partial perspective about future IR development. On the other hand, some researchers have tried to improve their retrieval results by re-ranking [1,2,3]. How much could they potentially improve their retrieval results via re-ranking or say what is the lowest upper bound for such improvement by re-ranking? This empirical study provides us with an answer that may help understand re-ranking.

There might not be an agreed standard on the level of relevancy and the optimality of the ranking, getting optimal ranking may be intractable [4,5], and the queries and information expressed in natural languages could be ambiguous, but we assume the queries and evaluation used by TREC (Text REtrieval Conference) Genomics track [6] are fair and proper. We assume a re-ranking obtained by always selecting the most relevant information unit with ties broken arbitrarily is *optimal*, which has been shown generally reasonable [4,5].

The rest of the paper is organized as follows. In Section 2, we provide the details of our empirical study method. Experimental results are given and described in Section 3. In Section 4, we discuss and analyze the results. Section 5 concludes.

2 Method

2.1 Dataset

We make the study based on the 63 runs submitted by 26 groups to the TREC 2007 genomics track[1]. The genomics track of TREC, running from 2003 to 2007, provided a common platform to evaluate the methods and techniques proposed by various research groups for biomedical IR.

From 2003 to 2005, the track focused on document-level retrieval for question answering. The document-level retrieval was scored based on the document mean average precision (MAP) (called the document measure in this paper). In its last two years (2006 & 2007), the track implemented and focused on a new task, called passage retrieval, where a passage could range from a phrase to a sentence or a paragraph of a document and must be continuous [7]. The task was evaluated based on two performances: the passage-level retrieval performance and the aspect-level retrieval performance. The passage-level retrieval performance, in 2006 was rated based on the amount of overlap between the returned passages and the passages deemed relevant by the judges (called

[1] A total of 27 groups submitted 66 runs, but we only got 63 runs from 26 groups for the study.

the passage measure in this paper), and in 2007 was scored by treating each character in each passage as a ranked document (called the passage2 measure) to address the "doubling score by breaking passages in half" problem of the passage measure [6]. The aspect-level performance was rewarded by the amount of relevant aspects reached and penalized by the amount of non-relevant passages ranked higher than the novel passages (called the aspect measure). The relevant aspects related with each topic (question) in 2006 were a set of MeSH terms (entities) assigned by the judges and in 2007 were a set of answer entities picked from the pool of nominated passages deemed relevant by the judges.

A passage is *novel* if it contains relevant aspects not existing in the passages ranked higher than it. Note by the aspect measure, no penalty would be applied if an irrelevant passage is ranked higher than a redundant (i.e., relevant but unnovel) passage. In IR field, some researchers consider the relevance judgment to encompass both the topical match between an information need and a document (or an information unit) and the novelty of the document such as [8,9]. They use "topicality" to refer to the subjective judgment of whether a document is related to the subject area of the user's information need, and "novelty" the degree to which the content of the document is new and beyond what the user has known. Some other researchers simply consider the relevance judgment to be topicality judgment and the relevant documents could be either novel or redundant such as [4,5]. We follow the usage of the latter in the paper.

For the question-answering task of TREC 2007 genomics track, there was a list of 36 topics in the form of questions that needed to be answered based on a collection of 162,259 HTML formatted documents collected from the electronic distribution of 49 genomics-related journals from the Highwire Press (www.highwire.org). All the 36 questions were selected from the information needs statements provided by the surveyed working biologists after being screened against the corpus to ensure that the relevant passages were present. The desired entity type, such as genes, proteins, diseases, mutations, etc., for answering each question was designated. For example, for the question "What [GENES] are genetically linked to alcoholism?", the answers would be the passages that relate one or more entities of type GENE to alcoholism.

To answer a question, up to 1000 passages could be nominated. Each nominated passage should include the document ID, the starting byte offset of the passage in the document, passage length, rank, and rank value. A set of 3 runs, each including the nominated passages to all 36 topics, could be submitted by one research group. Eventually a total of 66 runs from 27 groups were submitted. The judges identified the gold passages and located the answer entities for each topic from the pool of nominated passages. The performance of each run was then evaluated based on the gold passages and the assigned answer entities. A total of 4 measures were used to evaluate each run: the passage2 measure, the aspect measure, the passage measure and the document measure. The document-level performance was evaluated based on the passage retrieval results: a document was considered relevant for a topic if it had a relevant passage for the topic. Since the document retrieval was evaluated via the passage retrieval, and the passage2 measure was an improvement to the passage measure, we make our empirical study based on the two passage retrieval performance measures: the passage2 and the aspect measures.

2.2 Measure Passage2 and Optimization

Algorithm 1, summarized from the TREC 2007 genomics track scoring program, shows how the passage2 measure works.

Algorithm 1. Passage2 evaluation

```
   Input: {nominatedPassageSet[topic]}, {goldPassageSet[topic]}.
   Output: Average passage2 precision by topic.
 1 for each Topic do
 2     nume=0; deno=0; sumPrecision=0.0;
 3     for each nominated Passage do
 4         if no any relevant characters then
 5             deno += passageLength;
 6         else
 7             for each character do
 8                 if irrelevant or novel then
 9                     deno +=1;
10                     if novel then
11                         nume +=1;
12                         sumPrecision+=nume/deno;
13                     end
14                 end
15             end
16         end
17     end
18     count=numCharactersInGoldPassages[Topic];
19     averagePrecision[Topic]=sumPrecision/count;
20 end
```

From Algorithm 1, the nominated list of passages for each topic is processed from the top (ranked highest) to the bottom (ranked lowest). If a nominated passage is not relevant (i.e., does not contain any relevant characters), it is penalized by increasing *deno* by the passage length (line 5) since *deno* would be used as the denominator when calculating the sum of precision *sumPrecision* (line 12). Otherwise, the passage would be processed character by character (line 7). If the character is not within the corresponding gold passage range (*irrelevant*), only *deno* is increased by 1 (lines 8-9); if the character is within the corresponding gold passage range and has not been used for scoring (*novel*), both *deno* and *nume* would be increased by 1 (lines 8-11), which is considered a reward since *nume* would be used as the numerator in calculating *sumPrecision*. Nothing would be done if the character is within the corresponding gold passage range but has been used for scoring before. From the equation at line 12, if all relevant passages are ranked higher than irrelevant ones, and all relevant passages are within the respective gold passage ranges, *sumPrecision* would be equal to the sum of the lengths of all relevant passages:

$$sumPrecision = \sum_{p_i} len(p_i),$$

where p_i is a relevant passage. This is because in that case the penalties applied at line 5 would never be used in calculating *sumPrecision* and the penalty applied at line 9 would always be paired with the reward at line 11. Therefore, if all gold passages for *Topic* are exactly nominated, *averagePrecision[Topic]* would equal 100%.

From the analysis above, to get (an approximation to) the optimal passage2 ranking for a nominated run, we first need to push all irrelevant passages back behind all the relevant ones. This can be done by checking if a nominated passage has any overlap with any gold passages. Secondly, we should rank relevant passages such that the highest performance score is achieved. However, this might be a hard problem. We take heuristics by ranking higher the relevant passages that contain higher ratio of relevant characters. That is, we would order all nominated relevant passages based on

$$r - ratio = \frac{numRelevantCharacters(p_i)}{len(p_i)},$$

where p_i is a relevant passage. The higher its r-ratio is, the higher the passage should be ranked with ties broken arbitrarily.

2.3 Measure Aspect and Optimization

Algorithm 2, summarized from the genomics track scoring program, shows the details of the aspect-level performance evaluation.

Algorithm 2. Aspect evaluation

Input: {nominatedPassageSet[topic]}, {goldPassageSet[topic]}.
Output: Average aspect precision by topic.

```
1  for each Topic do
2      nume=0; deno=0; sumPrecision=0.0;
3      for each nominated Passage do
4          if there are any relevant aspects then
5              if numNewAspects > 0 then
6                  nume +=1; deno +=1;
7                  sumPrecision+= numNewAspects*nume / deno ;
8              end
9          else
10             deno +=1;
11         end
12     end
13     count=numUniqueAspects[Topic];
14     averagePrecision[Topic]= sumPrecision / count ;
15 end
```

From Algorithm 2, the nominated list of passages for each topic is processed from the top (ranked highest) to the bottom (ranked lowest). Any nominated passages containing relevant aspects are considered relevant (line 4); all other passages are considered irrelevant and are simply penalized by increasing *deno* by 1 (line 10). Any relevant passages

that contain new aspects are considered novel and would be rewarded by increasing both *nume* and *deno* by 1 (line 6). Nothing would be done for relevant passages that only contain previously seen aspects. That is, the redundant passages would have no impacts on the score. The variable *sumPrecision* is updated only upon novel passages (line 7). The aspect-level performance, *averagePrecision*, is finally calculated for each topic (line 14). The *averagePrecision* is actually equivalent to the *recall* of the relevant aspects [4]. From the equation at line 7, if all irrelevant passages are put after the novel ones, *sumPrecision* would equal the number of aspects all passages nominated for the topic contain:

$$\begin{aligned} sumPrecision &= \sum_{p_i} numNewAspects(p_i) \\ &= numUniqueAspects[Topic], \end{aligned}$$

where p_i is a novel passage. This is because in that case

$$\frac{nume}{deno} = 1$$

would always hold. Therefore, the maximum value for *sumPrecision* is 100%. From the analysis, we can get an approximation to the optimal aspect ranking for each nominated run by always selecting the passage with the most new aspects with ties broken arbitrarily.

2.4 Perfect Runs

We assume the perfect passage2 results for each topic are obtained by ordering all gold passages based on their r-ratio and the perfect aspect results for each topic are obtained by ordering the gold passages based on the number of new aspects they have with ties broken arbitrarily. The perfect result based on either the passage2 or the aspect measure would produce an MAP of 100%.

3 Experimental Results

We first re-rank all the 63 submitted runs to get their respective passage2 and aspect optimal runs. We then compare the performances of the two sets of optimal runs with that of the 63 runs.

Figure 1 shows the passage2 performances of the set of submitted and the respective set of passage2 optimal runs. It is indicated that all the passage2 optimal runs perform better than the respective submitted runs and mostly much better. Figure 2 shows the respective relative improvements of the passage2 optimal runs over the respective submitted runs from as little as 162% to as much as 1165% with the mean 475%. Figure 3 shows the distribution of these relative improvements. It is indicated that most relative improvements are located between 300% and 500%.

Figure 4 shows the aspect performances of the set of submitted and the respective set of aspect optimal runs. It is indicated that all the aspect optimal runs perform better than

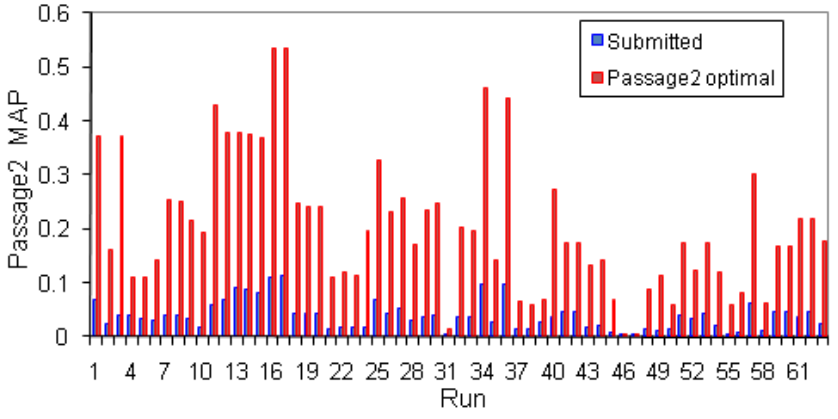


Fig. 1. Performances of the submitted and the passage2 optimal runs on the measure Passage2

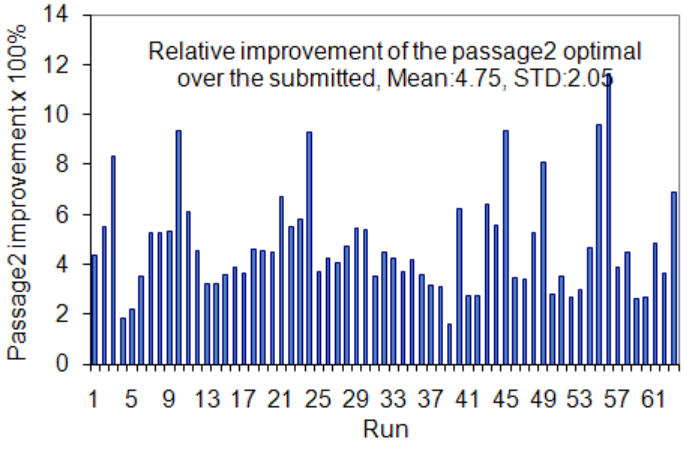


Fig. 2. Relative improvements of the passage2 optimal runs over the respective submitted runs

the respective submitted runs and mostly much better. Figure 5 shows the relative improvements of the aspect optimal runs over the respective submitted runs from as little as 63% to as much as 1252% with the mean 359%. Figure 6 shows the distribution of these relative improvements. It is indicated that most relative improvements are located between 100% and 300%.

Finally, we show how much the optimal runs need to be improved to be perfect (i.e., their distances). Figure 7 shows the relative improvements of the perfect runs over the respective passage2 optimal runs. It is indicated besides the 3 runs (31, 46, 47) that need exceptionally significant improvements, all other runs need improvements up to 1639%. This is consistent with Figure 8 where the relative improvements of the prefect runs over the respective aspect optimal runs are shown. Figure 8 shows the same 3 outliers need exceptionally outstanding improvements and all others need improvements up to 413%.

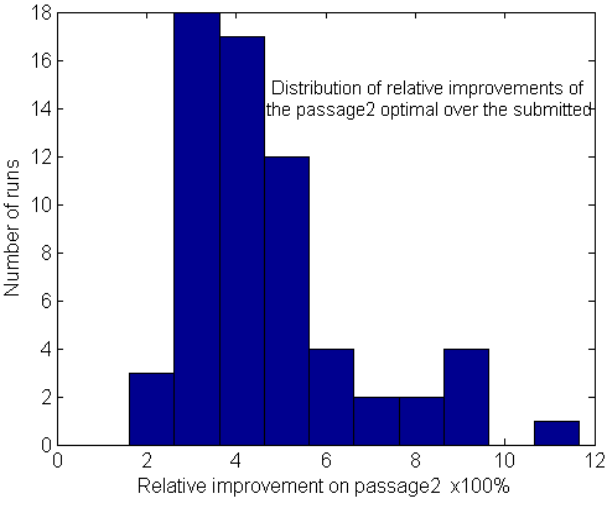


Fig. 3. Histogram of relative improvements of the passage2 optimal runs over the respective submitted ones

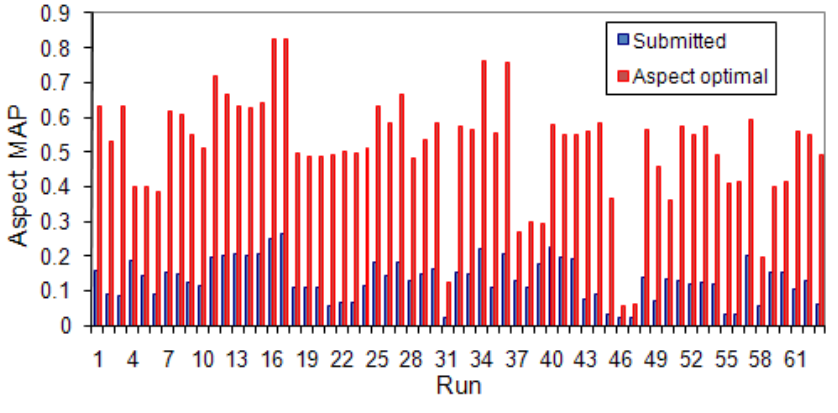


Fig. 4. Performances of the submitted and the aspect optimal runs on the measure Aspect

Refer to Figures 1 and 4, the 3 outlier runs perform not only worst but also exceptionally bad among all. This also indicates some very bad retrieval results are neither ranked well nor contain enough relevant information.

4 Discussion and Analysis

From the experimental results presented above, the optimal runs generally perform much better than the respective submitted runs on either passage2 or aspect measure. These optimal runs actually represent the lowest performance upper bound the

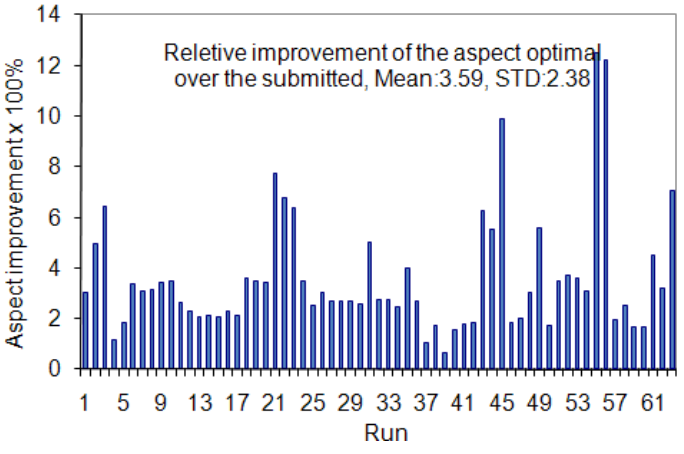


Fig. 5. Relative improvements of the aspect optimal runs over the respective submitted runs

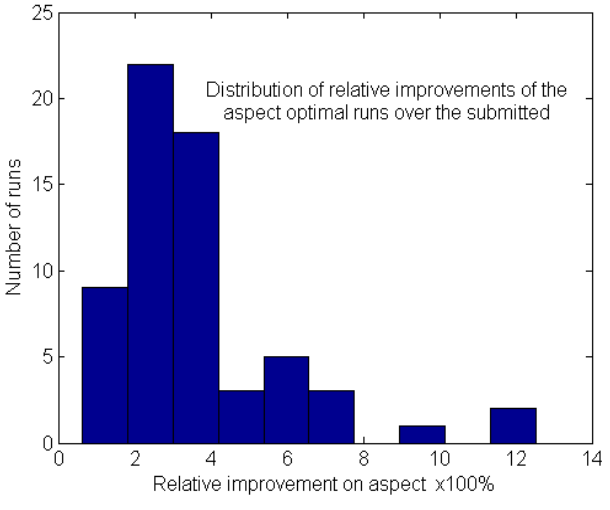


Fig. 6. Histogram of relative improvements of the aspect optimal runs over the respective submitted runs

respective submitted runs could achieve by re-ranking. If we assume all submitted runs as a whole represent the current average IR technology in genomics, we may take the average mean improvements on both passage2 and aspect

$$(475\% + 359\%)/2 = 417\%$$

as the average improvement we could reach by re-ranking over various measures. Therefore, the average relative improvement 417% may be considered the gap between

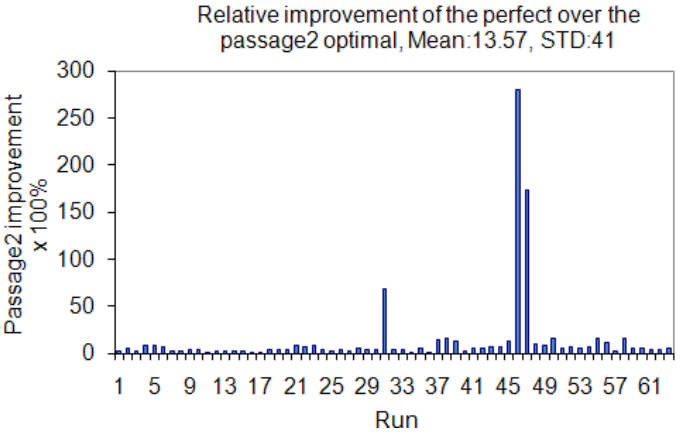


Fig. 7. Relative improvements of the perfect runs over the respective passage2 optimal runs

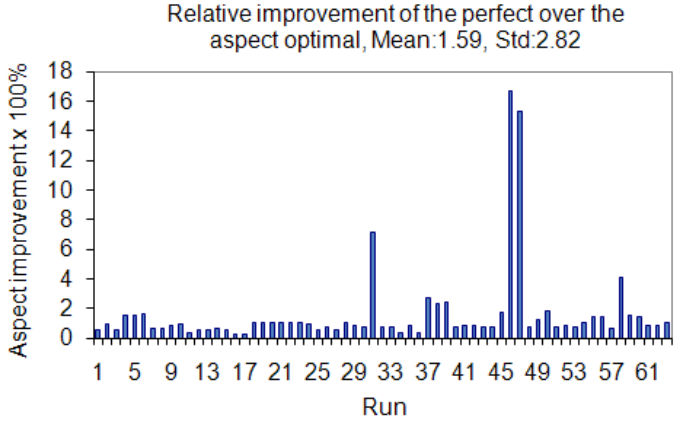


Fig. 8. Relative improvements of the perfect runs over the respective aspect optimal runs

the current average IR technology and the optimal IR in genomics that could be bridged via re-ranking. Different levels of IR performance improvements through re-ranking have been reported [1,2,3]. Some improvements are quite small (2%-5%) [2] and some improvements could only be made on a small subset of predefined topics [1]. It is reported [3] that a performance improvement of 27.09% was once achieved on TREC 2007 genomics track data. If we assume this is the best performance improvement we could make within one year, we may compute how many years we are away from the optimal IR with the following equation:

$$1 + 417\% = (1 + 27.09\%)^x,$$

where x is the number of years we are away from the optimal IR. We can get

$$x = \frac{ln5.17}{ln1.2709} = 6.85.$$

This definitely would not be the accurate distance we are away from the optimal IR, and so far there are still too many crucial factors that are quite uncertain to determine the development progress of the IR. However, it does tell us there is probably still a long way to go before the optimal IR even from an optimistic view.

We could similarly calculate how far away the perfect IR is from the optimal IR. The distance between the optimal IR and the perfect IR might be considered the amount of efforts we need to make on the techniques other than re-ranking such as query understanding and matching. We use the average mean perfect improvements on both passage2 and aspect measures to represent the average improvement made by perfect runs over optimal ones:

$$(1357\% + 159\%)/2 = 758\%.$$

We could get the distance x in years by the following equation:

$$1 + 758\% = (1 + 27.09\%)^x.$$

It turns out $x = 8.97$. This indicates we need more time from the optimal to the perfect than from the current to the optimal. In other words, from now on, we need about

$$6.85 + 8.97 = 15.82$$

years to get to the perfect IR. It can be easily shown exactly the same result would be obtained if the average relative improvement needed to make the submitted runs perfect is directly used in calculating the distance between the perfect IR and the submitted runs.

5 Conclusion

It has been about half a century since the automated information retrieval systems were first implemented in 1950s. Where are we now and how far away are we from the optimal or the perfect IR? We absolutely could not give an exact answer for such questions since there exist too many crucial factors that are still highly uncertain to determine IR development progress. In this paper, based on some assumptions, we empirically studied how much effort we might still need to make to get to the optimal and the perfect IRs. The study indicated we still have a long way to go to make the existing systems optimal and even a much longer way to go to make systems perfect even from an optimistic view.

This work first experimentally studied lowest upper bound regarding performance improvement by re-ranking, which might not have been realized by relevant researchers. The study indicated the improvements made by work in the literature are quite marginal relative to the big room provided by optimal rankings. How to make the performance improvement close to the upper bound may deserve more studying.

References

1. Yang, L., Ji, D., Tang, L.: Document re-ranking based on automatically acquired key terms in chinese information retrieval. In: COLING 2004, pp. 480–486 (2004)
2. Shi, Z., Gu, B., Popowich, F., Sarkar, A.: Synonym-based query expansion and boosting-based re-ranking: A two-phase approach for genomic information retrieval. In: TREC 2005 (2005)
3. Hu, Q., Huang, X.: A reranking model for genomics aspect search. In: SIGIR 2008, pp. 783–784 (2008)
4. Zhai, C., Cohen, W.W., Lafferty, J.: Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In: SIGIR 2003, pp. 10–17 (2003)
5. Clarke, C.L.A., Kolla, M., Gormack, G.V., Vechtomova, O., Ashkan, A., Büttcher, S., MacKinnon, I.: Novelty and diversity in information retrieval evaluation. In: SIGIR 2008, pp. 659–666 (2008)
6. Hersh, W., Cohen, A., Roberts, P.: TREC 2007 genomics track overview. In: TREC 2007, pp. 98–115 (2007)
7. Hersh, W., Cohen, A., Roberts, P., Rekapalli, H.K.: TREC 2006 genomics track overview. In: TREC 2006, pp. 68–87 (2006)
8. Boyce, B.: Beyond topicality: a two stage view of relevance and the retrieval process. Information Processing & Management 18, 105–109 (1982)
9. Xu, Y., Yin, H.: Novelty and topicality in interactive information retrieval. Journal of the American Society for Information Science and Technology 59, 201–215 (2008)

Adaptive Term Weighting through Stochastic Optimization

Michael Granitzer[1,2]

[1] Graz University of Technology, Graz, Austria
mgranitzer@tugraz.at
http://kmi.tugraz.at/
[2] Know-Center Graz, Graz, Austria
mgrani@know-center.at
http://www.know-center.at/

Abstract. Term weighting strongly influences the performance of text mining and information retrieval approaches. Usually term weights are determined through statistical estimates based on static weighting schemes. Such static approaches lack the capability to generalize to different domains and different data sets. In this paper, we introduce an on-line learning method for adapting term weights in a supervised manner. Via stochastic optimization we determine a linear transformation of the term space to approximate expected similarity values among documents. We evaluate our approach on 18 standard text data sets and show that the performance improvement of a k-NN classifier ranges between 1% and 12% by using adaptive term weighting as preprocessing step. Further, we provide empirical evidence that our approach is efficient to cope with larger problems.

1 Introduction

Term weighting schemes - statistical models for estimating the importance of a term - directly impact the information retrieval ranking and text mining accuracy; a fact shown by information retrieval evaluation initiatives like for example TREC[1], CLEF[2] or INEX[3]. Most of todays weighting techniques rely on parametric approaches, which have a defined functional form and a (usually small) number of parameters. Examples are *tfidf* weighting [1], *BM25* [2], language modeling [3], axiomatic weighting [4] etc. Parametric approaches suffer from the drawback that they do not adapt to particular domains and user needs. For example, searching for research publications requires a different weighting scheme than searching for restaurants.

To overcome the rigidity of parametric weighting schemes Metzler and Zaragoza [5] introduced semi-parametric and non-parametric weighting schemes recently. They extended Anh and Moffats [6] approach to binned ranking as a general form of dimensionality reduction. Binned ranking employs a partially order on terms and groups

[1] Text REtrieval Conference (http://trec.nist.gov/)
[2] Cross-Language Evaluation Forum (http://clef-campaign.org/)
[3] The INitiative for the Evaluation of XML retrieval http://www.inex.otago.ac.nz/

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 614–626, 2010.

them into k bins. A weight is assigned to every bin and subsequently used for ranking/similarity calculations[4]. Semi-parametric schemes now estimate bin weights similar to parametric schemes, i.e. in a functional form. More interestingly, non-parametric weighting schemes directly modify the bin weights in a supervised manner in order to optimize the ranking. As argued by Metzler and Zaragoza [5], this supervised approach has promising properties under the assumption that enough supervision can be provided. In particular it allows adapting ranking functions to particular domains/user needs, while binning reduces the dimensionality of the vector space and therewith keeps the parameter estimation tractable.

A related approach of adapting weight parameters is presented in [7]. Ernandes et. al. developed a context sensitive, adaptive weighting scheme. Here the weight of a word depends on the words statistical distribution and the words context defined by surrounding words. With resilient parameter adaptation gradient descent the actual influence factors among words and their context are estimated in a batch like manner.

Adapting distance metrics for classification has been described in [8]. Shwartz et. al. optimized the Mahalanobis distance via linear transformations in order to boost the accuracy of a k-NN classification algorithm, which can be seen as implicit application of an weighting scheme.

In this paper, we follow this line of research by learning a linear transformation to optimize a metric space. This linear transformation approximates the corpus dependent component of a term weight in order to optimize the expected similarities among documents. Since optimization is based on pair wise drawn data elements - yielding a quadratic complexity in terms of data set size - we use stochastic on-line gradient descent algorithms. Our approach allows adapting dot product based text mining algorithms to particular user and/or domain needs by providing expected similarities among documents, while keeping the algorithms itself unchanged.

With our work we carry on the ideas provided by Metzler and Zaragoza [5], but do not rely on binning for dimensionality reduction. Instead we use online stochastic gradient descent methods to keep the problem tractable. In contrast to Shwartz et. al. [8], we focus on inner product based similarity measures rather than on distances and, different than Ernandes et. al. [7], we adapt well known retrieval models. Hence, our approach is directly applicable in existing systems. Therefore, we contribute to adaptive, non-parametric weighting schemes by

- presenting a stochastic on-line approach to approximate the corpus dependent component of a term weight in order to optimize the expected similarities among documents
- showing that using pair wise training in such a setting is efficient
- showing that the adapted term weights improve nearest neighbor search as well as k-Nearest Neighbor classification.

The remainder of this paper is structured as follows: section 2 shows that linear transformations of a vector space correspond to some well known weighting schemes. This property is exploited in section 3 by approximating the linear transformation through

[4] With having only one bin, parametric weighting schemes can be seen as special case to binned retrieval.

on-line stochastic gradient descent. Experimental results are outlined in section 4, while section 5 concludes our work and points to future topics.

2 Term Weighting as Linear Transformation

Our approach relies on the observation that most weighting schemes are linear transformations $L : \Re^d \rightarrow \Re^d$ of a vector space $D = \{d_1 \ldots d_N\}, d_j \in \Re^d$ where d_j denotes the term vector of document v and $|D|$ the dimensionality of the vector space. One example for such an linear transformation is the well known *tfidf* family of weighting schemes. In its most basic form, the weight $w_{k,j}$ of the term k (corresponding to the k^{th} dimension of the vector space D) in document j is calculated as

$$w_{k,j} = tf_{k,j} \cdot idf_k = occ_{k,j} \cdot \log(\frac{N}{n_k}) \tag{1}$$

with $occ_{k,j}$ being the number of occurrences of term k in document j, N being the number of documents and n_k being the number of documents containing term k[5]. Obviously the weighting scheme has a document dependent part, the term frequency *tf*, and a corpus dependent part, the inverse document frequency *idf*. Rewriting the *idf*-part as diagonal matrix $L = diag(idf)$ - with idf being a d-dimensional vector representing the idf values of each dimension as $idf_k = \log(\frac{N}{n_k})$ and assuming that the document vectors contain the document specific part, i.e. $d_{j,k} = occ_{k,j}$ - the *tfidf* weighting scheme can be written as

$$\breve{d}_i = Ld_i \tag{2}$$

Similarly to *tfidf* , the Okapi *BM25* [2] consists of a document and a corpus specific term. Herein, the weight $w_{k,j}$ is given as

$$w_{k,j} = tf_{k,j} \cdot idf_k = \frac{(k_1+1)occ_{k,j}}{k_1(1-b+b\frac{l_d}{l_a})+occ_{k,j}} \cdot \log(\frac{N-n_k+0.5}{n_k+0.5}) \tag{3}$$

with l_d being the length of the documents (i.e. number of total terms), l_a being the average length of documents in a corpus, and b and k_1 being positive constants to tune the impact of document length and/or the term frequency. Again, by separating the document from the corpus specific part *BM25* -weighting can be seen as linear transformation of the according vector space.

The TREC evaluation series[6] as well as several research groups have shown that different weighting schemes, and therefore different linear transformations, have a dramatic impact on the performance of different algorithms. Moreover, differences depend strongly on the underlying corpus as well as on the task at hand such that no weighting scheme outperforms other weighting schemes in general.

Our approach targets this problem by automatically adapting the linear transformation to optimize expected similarities among documents. We focus on cosine based similarity measures since they are commonly used in information retrieval and text mining

[5] There are a number of different *tfidf* functions with slightly different calculations of the term frequency *tf* and inverse document frequency *idf*. For details see for example [9].

[6] Text REtrieval Conference (`http://trec.nist.gov/`)

algorithms like vector space retrieval, clustering or classification and, as we will show, allow the integration of linear transformation for their optimization. In particular, we start with the cosine similarity defined as

$$s_{i,j} = \frac{d_i' \cdot d_j}{\|d_i\| \|d_j\|} \tag{4}$$

which in case that $\|d_i\| = 1$ and $\|d_j\| = 1$ is reduced to a simple dot product ($'$ denotes the transpose of the vector). Hence, applying a weighting scheme as linear transformation, the resulting similarity becomes

$$s_{i,j} = \frac{(Ld_i)' \cdot Ld_j}{\|Ld_i\| \|Ld_j\|} \tag{5}$$

In order to optimize the linear transformation, we assume that the target similarity is given and that we have to adapt the linear transformation to approximate the expected target similarity. Such similarities could be provided for example by relevance judgments, user feedback or information visualization techniques [10]. Assuming that a sufficient number of target similarities are given, we expand the linear transformation in equation 5, rearrange coefficients and, under Euclidean norm we get

$$s_{i,j} = \frac{\sum_k l_k^2 d_{i,k} d_{j,k}}{\sqrt{\sum (l_k \cdot d_{i,k})^2} \sqrt{\sum (l_k \cdot d_{j,k})^2}} \tag{6}$$

The impact of term k on the similarity is defined by the transformation coefficients l_k, which have to be adapted over all terms in order to return the expected similarity. This adaption via on-line learning is shown in the following section.

3 On-Line Learning for Term Weighting

Given a target similarity between two documents, denoted as $\hat{s}_{i,j}$, our goal is now to find a linear transformation L to approximate $\hat{s}_{i,j}$ as good as possible. Formally we want

$$\hat{s}_{i,j} \hat{=} \frac{\sum_k l_k^2 d_{i,k} d_{j,k}}{\sqrt{\sum (l_k \cdot d_{i,k})^2} \sqrt{\sum (l_k \cdot d_{j,k})^2}} \tag{7}$$

yielding an equation system consisting of N^2 equations with $|D| - 1$ degrees of freedom.

Solving this equation system exhibits two problems: First, the existence of an exact solution is not guaranteed. This can be easily shown by assuming that two documents share no common term, but must have a similarity larger than 0. Second, the number of equations may be too large for practical problems. In most application scenarios a corpus contains several thousand documents, yielding a large set of equations through the quadratic dependency on the number of documents.

We favor on-line approaches over batch approaches due to several reasons. Typical text data sets have a high number of documents and correspondingly a high dimensionality of the vector space. So batch optimization would become infeasible since

it requires to store a $n^2 \times |D|$ sparse matrix to represent all inner document products $d_j \cdot d_i$. On-line approaches allow to randomly draw examples and optimize the cost function on a per-example basis, thereby overcoming the storage problem. Regarding convergence, which directly translates to runtime complexity, stochastic approaches can achieve higher convergence rates than batch algorithms in case of redundant data sets [11], i.e. data sets where examples share similar properties. Redundancies will likely occur in our problem setting since we can assume that document vectors share similar term distributions while having similar target similarities. Hence, stochastic on-line gradient descent seems to be suited best for the task.

For formulating eq. 7 as optimization problem we use a quadratic loss function. Formally we have

$$E(L) = \sum_{\forall i,j,i \neq j} (\hat{s}_{i,j} - s_{i,j})^2 \tag{8}$$

with $s_{i,j}$ begin the similarity as defined in equation 6. By being differentiable, the quadratic loss function allows us to estimate the gradient and to use gradient descent approaches for determining a good solution L^*.

For performing stochastic on-line gradient descent, we randomly draw two documents and their corresponding target similarity. Afterwards, the loss function $E(L)$ and its gradient w.r.t. the current data examples is calculated as

$$\frac{\partial E(L,i,j)}{\partial l_k} = 4\delta l_k d_{i,k} d_{j,k} \tag{9}$$

with $\delta = \hat{s}_{i,j} - s_{i,j}$ and $E(L,i,j)$ being the loss for the document pair i, j.

To avoid negative weights, updates dropping a dimension below zero are ignored. Instead the current weight of this dimension is made smaller by dividing it by two. Hence, the new weight is $l_k = l_k + \Delta_k$ with

$$\Delta_k = \begin{cases} \eta \delta l_k d_{i,k} d_{j,k} & if l_k + \Delta_k > 0 \\ l_k/2 & if l_k + \Delta_k < 0 \end{cases} \tag{10}$$

and $\eta > 0$ being the learning rate resp. step size. Convergence depends strongly on the chosen learning rate. A low learning rate slows down convergence, while a too large learning rate may avoid convergence towards the optimal solution. Therefore, we choose to anneal the learning rate logarithmically via

$$\eta_t = \eta / \log(t+2) \tag{11}$$

with t being the current step.

Clearly, this setting allows us to easily add new examples and adapt the linear transformation on the fly. So if new target similarities are provided to the algorithm, the linear transformation can be adapted easily and with rather low computational complexity. Hence, user feedback can be integrated straightforward.

4 Experiments

In order to evaluate our approach to adaptively weight terms we conducted two sets of experiments on 18 standard text data sets. In the first set of experiments we evaluated

the correctness of our approach by learning a known *tfidf* scheme (see section 4.2). The second set of experiments targets the practical case of text classification. We show that our approach, used as preprocessing step, improves a k-NN classifier in its classification accuracy (see section 4.3).

4.1 Data Sets

For our experiments we used the Cluto data sets [12]. Those data sets have been explored mostly for document clustering and provide a collection of standard text data sets given as document term matrix. The datasets - compiled from well known benchmark data sets like TREC, Cranfield and the Reuters-21578 corpus - have been already preprocessed by removing stop words and are stemmed using Porter's algorithm. All terms that occur only in one document were eliminated (see [12] for further details)[7]. Table 1 depicts the different data sets and their properties.

Table 1. Data Set Overview

Data Set	# Exa.	# Terms	# Classes
Classic	7094	41681	4
Cranmed	2431	41681	2
Fbis	2463	2000	17
Hitech	2301	126373	6
K1a	2340	21839	20
K1b	2340	21839	6
La1	3204	31472	6
La2	3075	31472	6
New3	9558	83487	44

Data Set	# Examples	# Terms	# Classes
Re0	1504	2886	13
Re1	1657	3758	25
Reviews	4069	126373	5
Sports	8580	126373	7
Tr11	414	6429	9
Tr12	313	5804	8
Tr31	927	10128	7
Tr41	878	7454	10
Wap	1560	8460	20

4.2 Learning a Given Weighing Scheme

In order to evaluate that we can efficiently approximate a weighting scheme based on provided similarities, our first experiment reconstructs the *tfidf* weighting scheme. The target similarity values are estimated by applying a *tfidf* weighting scheme (see equation 1) to each document and by calculating the dot product similarity matrix (see equation 6) accordingly. The linear transformation L is initialized uniformly as diagonal matrix with $1/d$ in each diagonal element. While doing the online gradient descent, we evaluated the Pearsons correlation coefficient ρ between the approximated transformation L and the original *tfidf* transformation after every 25 examples drawn. The correlation coefficient determines how good both transformations match independent of their actual scaling. Hence, we measure whether our approach correctly approximates the differences of term importance. Table 2 depicts the results for all data sets.

[7] One exception is the FBIS data set, who also had been pruned to the 2000 most important features according to a forum entry of George Karypis `http://glaros.dtc.umn.edu/gkhome/node/353`.

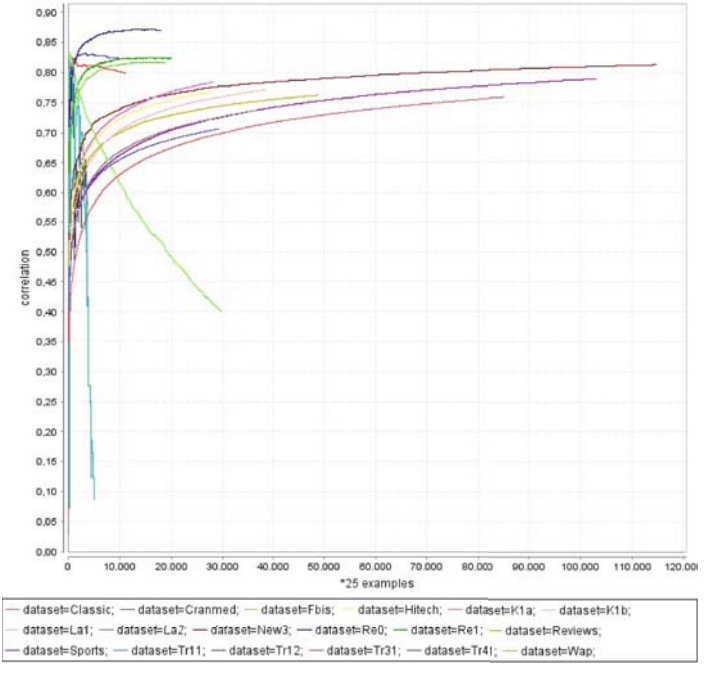
Table 2. Highest correlation coefficient ρ between the optimized and the original tfidf vector. The number of examples drawn to achieve the correlation are shown for the different learning rates η.

Data Set	η=0.1		η=1.0		dataset	#terms
	ρ	# examples drawn	ρ	# examples drawn	size	
Classic	*0.759*	85,128	0.845	85,126	7,094	41,681
Cranmed	*0.706*	29,172	0.782	29,172	2,431	41,681
Fbis	*0.832*	1,151	0.828	197	2,463	2000
Hitech	*0.767*	27,612	0.824	27,609	2,301	126,373
K1a	*0.784*	28,080	0.847	28,075	2,340	21,839
K1b	*0.784*	28,080	0.847	28,075	2,340	21,839
La1	*0.771*	38,448	0.834	23,067	3,204	31,472
La2	*0.737*	36,900	0.826	32,612	3,075	31,472
New3	*0.812*	114,696	0.829	114,695	9,558	83,487
Re0	*0.872*	18,021	0.876	5,358	1,504	2,886
Re1	*0.824*	19,808	0.814	9,422	1,657	1,657
Reviews	*0.762*	48,828	0.823	48,824	4,069	126,373
Sports	*0.789*	102,954	0.867	102,958	8,580	126,373
Tr11	*0.822*	958	0.855	53	414	6,429
Tr12	*0.820*	354	0.819	17	313	5,804
Tr31	*0.824*	11,109	0.850	1,504	927	10,128
Tr41	*0.832*	10,525	0.839	1,104	878	7,454
Wap	*0.817*	18,720	0.824	17,023	1,560	8,460

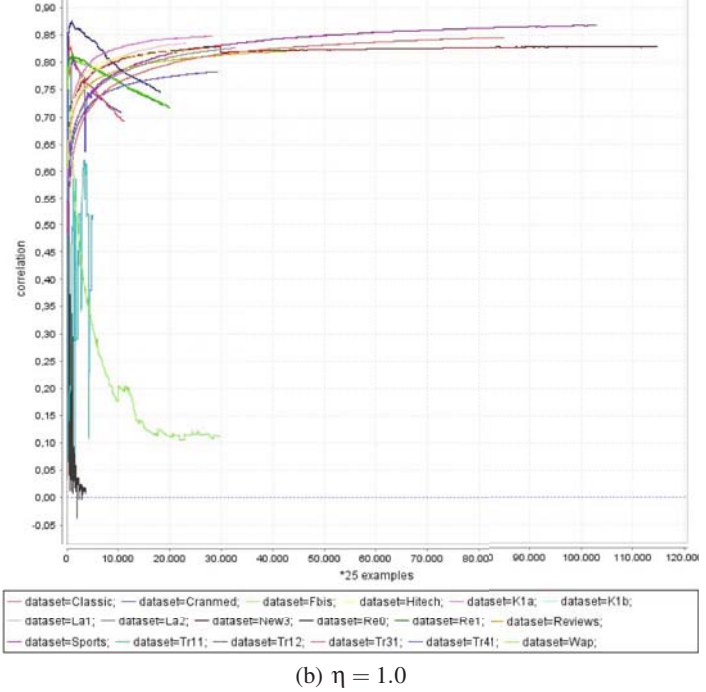
Accuracy: Two important properties regarding accuracy can be observed: Firstly, we achieve high correlation coefficients for every data sets, indicating that our approach is capable to identify the linear transformation accompanying a given similarity distribution. Second, the solution could be achieved within a small numbers of iterations. While it depends on the learning rate used, the highest correlation coefficients could be achieved in the range of around 2-10 times the number of term vectors in the data set. Hence, every run did only see a fraction of all possible N^2 examples. Intuitively this seems to be valid since the number of parameters is much smaller than the number of examples. Thus, it can be assumed that *the data set is highly redundant* which is *optimal for stochastic approaches and allows to find good transformations efficiently*.

Convergence: Another important aspect to discuss is convergence. Figure 1 shows the convergence for $\eta = 1.0$ and $\eta = 0.1$ respectively. As expected, the different learning rates influence convergence stability and $\eta = 0.1$ achieves a more stable accuracy over the different data sets. Exceptions are the fbis and the Tr11 data sets. They rapidly achieve high correlation values but drop afterwards. We see the reason therefore in the rather different numbers of dimensions (Fbis) and examples (Tr11). However, for the majority of data sets the solution is improved with every iteration.

We conclude that all data sets achieve high correlation values rather fast and continue to improve them slowly. *Hence, on-line gradient descent allows to efficiently approximate a given weighting scheme.*



(a) $\eta = 0.1$



(b) $\eta = 1.0$

Fig. 1. Correlation per 25 examples for different learn rates η

4.3 Weight Adaption for k-NN Classification

To estimate the impact on nearest neighbor search and classification, we considered the improvement of k-NN classifiers via adapting the term weighting scheme based on class information in the training set. k-NN classifiers assign a document to the class where the k nearest neighbors of this document belong to. In adapting the weighting scheme such that documents of the same class become more similar while documents belonging to different classes become more dissimilar, we assume that k-NN classification performance has to increase. Hence, for this experiment the target similarity among two documents is set to 1.0 if they belong to the same class and 0.0 otherwise. Note that we are moving from a continuous spectrum of target values to a discrete one. Besides the actual improvement in classification accuracy, we also evaluate whether stochastic gradient descent can improve a particular parametric weighting scheme. Therefore, we distinguish whether documents are already weighted with a corpus specific part (denoted *tfidf* & *BM25* in the result tables) or not (denoted with *tf* in the result tables). In both cases the found linear transformation adds a multiplicative scaling term to each dimension of the vector space.

In our experiments, three folded cross-validation splits the data set into a training and test data set for each fold[8]. During the learning phase, which usually involves no calculations in classical k-NN classifiers, stochastic gradient descent updates the weighting scheme iteratively. Every time after randomly drawing half of the total number of term vectors n, the F_1 measure is evaluated on the test set. Results for all data sets are depicted in table 3 showing the improvements for the different weighting schemes. The learning rate η has been set to 2.0.

Improvement: While the quadratic loss function may not be optimal for the discrete classification task in this use case, results show that a significant improvement can be achieved. In general, every run seems to improve the F_1 measure. The improvements range from around 12% to around 1%, depending on the data set and weighting scheme. For example, the Cranmed data set already achieved a F_1 measure of 0.9933 without adapting the weighting scheme. In this case only marginal improvements are possible. The average improvement over all data sets is about 3.6%, whereas improvement rates for *BM25* vary more than for *tfidf* and *tf* in terms of standard deviation. *Overall, our approach significantly improves classification accuracy on average.*

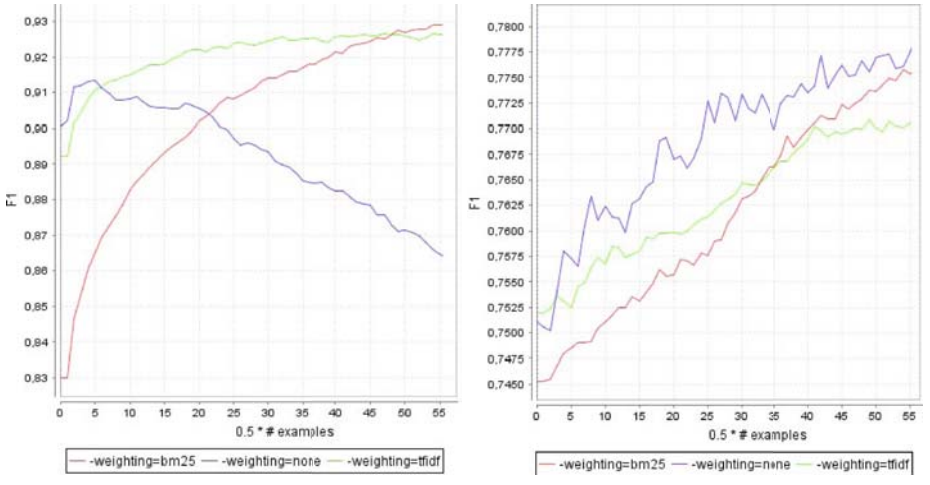
Convergence: Besides the actual improvement, convergence behavior plays an important role. For all data sets and experiments, we analyzed the convergence behavior of the improvement. It turned out that improvements in F_1 measure differ strongly among the different weighting schemes. For the *BM25* and *tfidf* weighting scheme nearly every iteration increased the F_1 measure for all data sets (see figure 2(b) as an example on the LA1 data set). Since convergence depends strongly on the chosen learning rate, one can argue that *BM25* and *tfidf* weighted document vectors yield to more robust convergence behavior. *tf* weighted examples show a different pictures (see figure 2(a) as an example on the Classic data set). Around half of the data sets show an increase

[8] We choose three folded cross-validation over ten-folded cross validation in order to reduce the required computation time for our experiments.

Table 3. Improvement & achieved F_1 value for 3-folded k-NN Classification. F_1^{max} depicts the maximal achieved F_1 value, F_1^{ori}, the original F_1 value without adaptive weighting and $\frac{F_1^{max}}{F_1^{ori}}$ the corresponding improvement

	BM25			*tf*			*tfidf*		
	F_1^{max}	F_1^{ori}	F_1^{max}/F_1^{ori}	F_1^{max}	F_1^{ori}	$\frac{F_1^{max}}{F_1^{ori}}$	F_1^{max}	F_1^{ori}	F_1^{max}/F_1^{ori}
Classic	0.9305	0.8293	*1.122*	0.9150	0.9004	1.016	0.9287	0.8914	1.042
Cranmed	0.9962	0.9933	1.003	0.9703	0.9617	1.009	0.9892	0.9802	1.009
Fbis	0.7421	0.7262	1.022	0.7146	0.7033	1.016	0.7053	0.6891	1.023
Hitech	0.5879	0.5827	1.009	0.6360	0.6100	1.043	0.6020	0.5764	*1.045*
K1a	0.3620	0.3574	1.013	0.6502	0.6121	1.062	0.5728	0.5581	1.026
K1b	0.4467	0.4124	1.083	0.8516	0.8226	1.035	0.7886	0.7822	1.008
La1	0.7804	0.7437	1.049	0.7801	0.7510	1.039	0.7716	0.7518	1.026
La2	0.7513	0.7178	1.047	0.7900	0.7663	1.031	0.8006	0.7829	1.023
New3	0.6813	0.6726	1.013	0.7095	0.6971	1.018	0.6819	0.6697	1.018
Re0	0.6666	0.6193	1.076	0.7233	0.6903	1.048	0.7290	0.6658	1.095
Re1	0.6742	0.6199	1.088	0.6306	0.5855	1.077	0.7266	0.6745	1.077
Reviews	0.8904	0.8772	1.015	0.9064	0.8815	1.028	0.9050	0.8673	1.043
Sports	0.9438	0.9304	1.014	0.8937	0.8932	1.001	0.9144	0.8975	1.019
Tr11	0.6762	0.6642	1.018	0.7685	0.7403	1.038	0.7544	0.7039	1.072
Tr12	0.6405	0.6313	1.015	0.7938	0.7495	1.059	0.8197	0.7944	1.032
Tr31	0.8837	0.8757	1.009	0.8865	0.8626	1.028	0.8999	0.8880	1.013
Tr41	0.9031	0.8819	1.024	0.8925	0.8722	1.023	0.9159	0.9048	1.012
Wap	0.4347	0.4299	1.011	0.6686	0.6183	*1.081*	0.6255	0.6041	1.035
Average			1.035			1.036			1.034
Std. Dev.			0.035			0.022			0.025

of the F_1 measure after the first few iterations, but decreases then. The behavior could be interpreted as overfitting on the training set. A further explanation would be a too slow annealing of the learning rate, so that training does not converge to a good solution. However, since improvements do not steadily increase, it would be required to select a particular solution. Without using a validation set, this remains problematic since selecting a solution in stochastic on-line optimization is hard to solve in general. Particularly, we could not identify a good heuristic therefore in our task. In terms of number of drawn examples we observe that improvements can be achieved after only doing a few steps. This is important since the number of total examples is quadratic. In particular it seems that similarity calculations among documents yield to a redundant set of training examples for approximating similarities. One reason therefore may be seen in the fact that the number of possible training examples is much larger than the degree of freedom of the optimization problem, i.e. $n^2 \gg |D|$, such that not all examples have to be seen. Hence, stochastic online approaches have an advantage over batch techniques. *Overall, our approach achieves stable convergence at least for* BM25 *and* tfidf *weighted vector spaces.*



(a) Classic dataset (b) La1 data set

Fig. 2. Convergence rates after every $0.5 * |D|$ randomly drawn examples

Influence of the weighting scheme: One interesting observation made in our experiments is the difference in accuracy among the different weighting schemes. Surprisingly, *tf* outperforms *idf* on 10 data sets and *BM25* on 12 data sets; *tfidf* outperforms *BM25* also on 12 data sets. In terms of absolute difference, *tf* and *tfidf* outperform *BM25* by a large degree. Hence, ignoring the corpus specific weighting component achieves the best performance on average. Partially stemming and stop word filtering during preprocessing may be responsible for this effect, but does not explain it fully. However, since our optimization approach improves every weighting scheme, the relative differences among weighting schemes stay the same. Also, those differences do not influence our results and thus can be ignored in our work.

5 Conclusion

In this paper we introduced a new approach to automatically estimate the corpus related component for weighting schemes. Our experiments on 18 data sets showed that our approach (i) can approximate a given weighting scheme rather efficiently using online stochastic gradient descent, (ii) improves classification accuracy in a range between 1% and 12% with an average of 3.6%, (iii) steadily increases improvements on F_1 measures for *BM25* and *tfidf* weighted document vectors and (iv) can achieve significant improvements after drawing only a fraction of all possible $O(n^2)$ examples.

While our results are promising, several issues remain open. The loss function is not optimally suited for the classification case. Here, cross entropy would define a better loss function [13]. Second, stopping criterion's and solution selection are potential problems in our approach. Another interesting extension would be to consider non-linear

transformations by adding weights between different terms that co-occur together in a similarity calculation. Hence, the diagonal matrix L would become a symmetric matrix reflecting term-term relationships.

Future work will address those issues, as well as the application to more complex scenarios like for example learning to rank retrieval results or to integrate a-priori knowledge into content based ranking functions. Moreover, since we can optimize the representation of documents towards an expected similarity measures, it would be possible to map between data set of different characteristics or to adapt data set similarity through visualization.

Acknowledgments

The Know-Center GmbH Graz is funded within the Austrian COMET Program - Competence Centers for Excellent Technologies - under the auspices of the Austrian Ministry of Transport, Innovation and Technology, the Austrian Ministry of Economics and Labor and by the State of Styria. COMET is managed by the Austrian Research Promotion Agency FFG.

References

1. Salton, G., McGill, M.J.: Introduction to Modern Information Retrieval. McGraw-Hill, Inc, New York (1986)
2. Robertson, S., Walker, S., Jones, S., Hancock-Beaulieu, M.M., Gatford, M.: Okapi at trec-3. In: Proceedings of the Third Text REtrieval Conference (TREC 1994), pp. 109–126 (1996)
3. Ponte, J.M., Croft, W.B.: A language modeling approach to information retrieval. In: SIGIR 1998: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, pp. 275–281. ACM, New York (1998)
4. Fang, H., Zhai, C.: An exploration of axiomatic approaches to information retrieval. In: SIGIR 2005: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 480–487. ACM, New York (2005)
5. Metzler, D., Zaragoza, H.: Semi-parametric and non-parametric term weighting for information retrieval. In: Azzopardi, L., Kazai, G., Robertson, S., Rüger, S., Shokouhi, M., Song, D., Yilmaz, E. (eds.) ICTIR 2009. LNCS, vol. 5766, pp. 42–53. Springer, Heidelberg (2009)
6. Anh, V.N., Moffat, A.: Simplified similarity scoring using term ranks. In: SIGIR 2005: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 226–233. ACM, New York (2005)
7. Ernandes, M., Angelini, G., Gori, M., Rigutini, L., Scarselli, F.: Adaptive context-based term (re)weighting an experiment on single-word question answering. Frontiers in Artificial Intelligence and Applications 141, 1 (2006)
8. Shwartz, S.S., Singer, Y., Ng, A.Y.: Online and batch learning of pseudo-metrics. In: ICML 2004: Proceedings of the twenty-first international conference on Machine learning, p. 94+. ACM, New York (2004)
9. Manning, C.D., Schutze, H.: Foundations of Statistical Natural Language Processing, vol. 8. MIT Press, Cambridge (2000, 2002)

10. Granitzer, M., Neidhart, T., Lux, M.: Learning term spaces based on visual feedback. In: Proc. 17th International Conference on Database and Expert Systems Applications DEXA 2006, pp. 176–180. IEEE, Los Alamitos (2006)
11. Bottou, L.: Stochastic learning. In: Bousquet, O., von Luxburg, U., Rätsch, G. (eds.) Machine Learning 2003. LNCS (LNAI), vol. 3176, pp. 146–168. Springer, Heidelberg (2004)
12. Zhao, Y., Karypis, G.: Evaluation of hierarchical clustering algorithms for document datasets. In: Proc. of CIKM 2002, McLean, Virginia, pp. 515–524 (2002)
13. Bishop, C.M.: Neural networks for pattern recognition. Oxford University Press, Oxford (1996)

Enhancing Text Classification by Information Embedded in the Test Set

Gabriela Ramírez-de-la-Rosa, Manuel Montes-y-Gómez, and Luis Villaseñor-Pineda

Laboratory of Language Technologies
National Institute of Astrophysics, Optics and Electronics
Luis Enrrique Erro No. 1, Sta. María Tonantzintla, Pue., 72840, Mexico
{gabrielarr,mmontesg,villasen}@inaoep.mx

Abstract. Current text classification methods are mostly based on a supervised approach, which require a large number of examples to build models accurate. Unfortunately, in several tasks training sets are extremely small and their generation is very expensive. In order to tackle this problem in this paper we propose a new text classification method that takes advantage of the information embedded in the own test set. This method is supported on the idea that similar documents must belong to the same category. Particularly, it classifies the documents by considering not only their own content but also information about the assigned category to other similar documents from the same test set. Experimental results in four data sets of different sizes are encouraging. They indicate that the proposed method is appropriate to be used with small training sets, where it could significantly outperform the results from traditional approaches such as Naive Bayes and Support Vector Machines.

1 Introduction

The tremendous amount of digital documents available on the Web has motivated the development of different automatic mechanisms that facilitate their access, organization and analysis. One example of such mechanisms are text classification methods, which focus on the assignment of documents into a set of predefined classes or topics [1].

Over the years several methods and algorithms for text classification have been proposed. In particular, the leading approach considers the use of machine learning techniques such as bayesian models, support vector machines and prototype-based classifiers to mention some. Under this supervised approach it is necessary to have an adequate training set consisting of manually labeled documents. As expected, the more the labeled documents are, the better the classification model is [2]. Unfortunately, in many real-world applications training sets are extremely small, and, what is more, their generation is very expensive.

Regarding the above problem, current efforts have focused on the generation of high-performance classification models using few labeled training data. On the

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 627–637, 2010.

one hand, some methods take advantage of available unlabeled documents to, iteratively, generate a robust classification model [3,4]. On the other hand, there are some methods that use information about the similarity of the documents from the own test collection in order to improve their classification. Particularly, most of these methods employ clustering techniques to enrich the representation of documents by adding or replacing some attributes [2,5,6].

The approach proposed in this paper belongs to the second group of works; nevertheless, it does not aim to enrich the representation of test documents by including information extracted from other similar documents, instead, it attempts to improve their individual classifications by considering the categories assigned to their nearest neighbors (from to same test set). In other words, the idea behind our proposal may be expressed by the popular proverb "a man is known by the company he keeps".

Given that prototype-based classifiers are very simple and have demonstrated to consistently outperform others algorithms such as Naive Bayes, K-Nearest Neighbors and C4.5 in text classification tasks [1], we decided to implemented the proposed approach using this classification algorithm. In general, our prototype-based method decides about the category of a given document by determining the class which prototype is more similar to it and its nearest neighbors.

Experimental results in four data sets of different sizes are encouraging. They indicate that the proposed approach could significantly outperform the results from a traditional prototype-based method as well as the results achieved by Naive Bayes and Support Vector Machines. On the other hand, these results also demonstrate the appropriateness of the approach for dealing with small training sets.

The remainder of paper is organized as follows. Section 2 explains the prototype-based classification method. Section 3 introduces the proposed approach. Section 4 describes the experimental configuration and shows the results obtained in four document collections. Finally, Section 5 presents our conclusions and exposes some future work ideas.

2 Prototype-Based Classification

This section describes the prototype-based classification method, which is used as base method in the proposed approach.

Prototype-based classification is one of the traditional methods for supervised text classification. This method may be summed up in a few words as follows. In the training phase, it considers the construction of one single representative instance, called prototype, for each class. Then, in a test phase, each given unlabeled document is compared against all prototypes and is assigned to the class having the greatest similarity score [1,7,8,9]. Evidently there are several ways to calculate the prototypes as well as to measure the similarity between documents and prototypes. Next we describe the alternative used in this paper.

The definition of the prototype for each class c_i is based on the normalized sum model, where each class is represented by a vector which is the sum of all document vectors from the class, normalized so that it has a unitary length [9,10]:

$$P_i = \frac{1}{||\sum_{d \in c_i} d||} \sum_{d \in c_i} d \tag{1}$$

In this case, documents are represented by vectors in the term-space, $d = \{w_1, w_2, ..., w_m\}$, where m indicates the number of different terms in the whole training set.

On the other hand, the assignation of the category to a given unlabeled document d is based on the following criterion:

$$class(d) = \arg\max_i(sim(d, P_i)) \tag{2}$$

where,

$$sim(d, P_i) = \frac{d \cdot P_i}{||d|| \times ||P_i||} \tag{3}$$

In Formulas 1 and 3, $||z||$ denotes the 2-norm of z, and $v \cdot z$ denotes the dot product of v and z vectors.

3 The Proposed Method

Figure 1 shows the general schema of the proposed method. It consists of two main phases. The first focuses on the construction of the class prototypes using the traditional techniques. The second involves, on the one hand, the identification of the nearest neighbors for each unlabeled document, and, on the other hand, their classification considering information from their own as well as from their neighbors. Following we present a brief description of each one of these processes.

Prototype Construction. This process carries out the construction of the class prototypes. In particular, given a set of labeled documents (i.e., training set) organized in set of classes, it computes the prototype for each class using Formula 1. This process is performed only once at the training phase.

Nearest Neighbors Identification. This process focuses on the identification of the N nearest neighbors for each document of the test set. In order to do that it firstly computes the similarity between each pair of documents from the test set using the cosine formula (refer to Formula 3), and then, based on the obtained similarity values, selects the N nearest neighbors for each document.

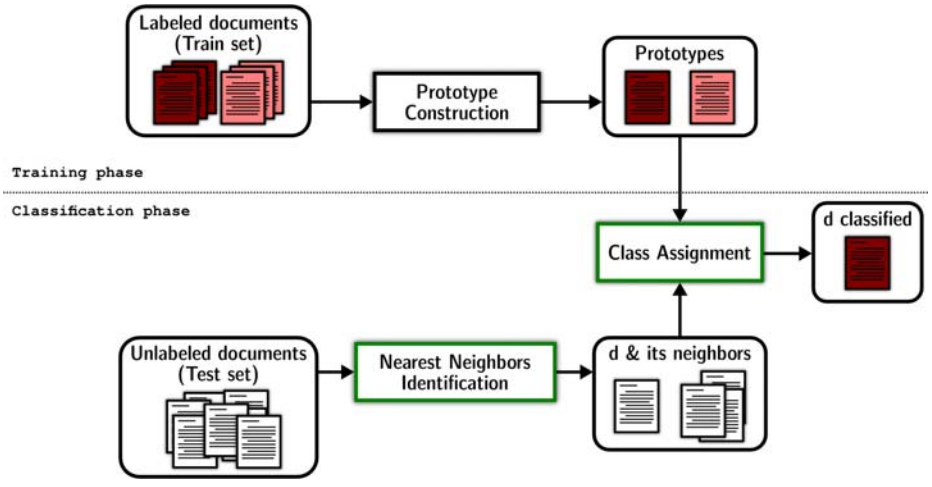


Fig. 1. Genaral scheme of the proposed text classification method

Class Assignment. Given a document d from the test set in conjunction with its N nearest neighbors, this process assigns a class to d using the following formula:

$$class(d) = \arg\max_i \left(sim(d, P_i) + \lambda \frac{1}{N} \sum_{j=1}^{N} [inf(d, v_j) \times sim(v_j, P_i)] \right) \quad (4)$$

where,

- sim is the cosine similarity function defined in Formula 3.
- N is the number of neighbors considered to provide information about document d.
- λ is a constant used to determine the relative importance of both, the information from the own document (d) and the information from its neighbors. The greater the value of λ is, the greater the contribution of the neighbors, and vice versa.
- inf is an influence function used to weight the contribution of each neighbor v_j to the classification of d. The purpose of this function is to give more relevance to the closer neighbors. In particular, we define this influence in direct proportion to the similarity between each neighbor and d calculated using the cosine formula (refer to Formula 3).

In order to give more information about these processes, Figure 2 presents the algorithm of the proposed method.

Let L be the set of labeled documents from the training set, U the set of test documents, C the set of classes in the set L, V^d the set of N neighbors of d, T_L the terms obtained from L, T_U the terms obtained from U.

```
Represent each d ∈ L by a vector d = {t_1, t_2, ..., t_|T_L|}.
For each c_i ∈ C
    Compute the prototype P_i using Formula 1.

Represent each d ∈ U by a vector d = {t_1, t_2, ..., t_|T_U|}.
For each d ∈ U
    V^d ← ∅.
    repeat from 1 to N
        Search v ∈ {U − V^d − d} : sim(d, v) is the greatest, where sim
        is given by Formula 3.
        V^d ← {V^d + v}.

Represent each d ∈ U by a vector d = {t_1, t_2, ..., t_|T_L|}.
For each d ∈ U
   Assign a class using Formula 4.
```

Fig. 2. Algorithm of the proposed method

4 Experimental Evaluation

4.1 Datasets

For the evaluation of the proposed method we considered the R8 collection. This collection was previously used by Cardoso-Cachopo and Oliveira [9], and it is formed by the eight largest classes from the Reuters-21578 collection, which documents belong to only one class. Table 1 shows some data about this collection, such as the number of documents per class in the training and test sets.

Table 1. The R8 collection

Class	Documents in training set	Documents in test set
acq	1596	696
crude	253	121
earn	2840	1083
grain	41	10
interest	190	81
money-fx	206	87
ship	108	36
trade	251	75
Total	5485	2189

Table 2. The four evaluation datasets

Collection	Documents in training set	Vocabulary
R8	5485	3711
R8-reduced-41	328	2887
R8-reduced-20	160	1807
R8-reduced-10	80	1116

With the aim of evaluating the proposed method in situations having small training sets, we generated three smaller collections from the original R8 corpus: R8-reduced-41, R8-reduced-20 and R8-reduced-10, consisting of 41, 20 and 10 labeled documents per class respectively. Table 2 shows some data about these four collections, such as the number of documents in the training set and the number of terms from the vocabulary of each class. The number of documents in the test set were not included since they are the same for all collections (2189) and were previously presented in Table 1.

4.2 Evalaution Measure

The evaluation of the performance of the proposed method was carried out by means of the F-measure. This measure is a linear combination of the precision and recall values from all class $c_i \in C$. It is defined as follows:

$$F - Measure = \frac{1}{|C|} \sum_{i=1}^{|C|} \left[\frac{2 \times Recall(c_i) \times Precision(c_i)}{Recall(c_i) + Precision(c_i)} \right] \tag{5}$$

$$Recall(c_i) = \frac{number\ of\ correct\ predictions\ of\ c_i}{number\ of\ examples\ of\ c_i} \tag{6}$$

$$Precision(c_i) = \frac{number\ of\ correct\ predictions\ of\ c_i}{number\ of\ predictions\ as\ c_i} \tag{7}$$

4.3 Baseline Results

In order to generate the baseline results we considered three of the most used methods for text classification, namely, Naive Bayes (NB) [11], Support Vector Machines (SVM) [12] and the prototype-based method (PBC) described in Section 2. Table 3 shows the results obtained by these methods in the four used datasets. These results confirm the robustness of the prototype-based method for dealing with small training sets. Particularly, it is of special interest to notice that reducing the training set in 94% (R8-reduced-41 in relation to R8) only caused a decrement of 4.7% in the F-measure value.

Table 3. F-measure results from three classification methods

Collection	NB	SVM	PBC
R8	0.828	**0.886**	0.876
R8-reduced-41	0.747	0.812	**0.836**
R8-reduced-20	0.689	0.760	**0.803**
R8-reduced-10	0.634	0.646	**0.767**

4.4 Results

As described in Section 3, the main idea of the proposed method is to classify the documents by considering not only their own content but also information about the assigned category to other similar documents. Based on this idea, Formula 4 attempts to combine both kinds of information, being λ a constant that determines their relative importance.

Considering the above situation, we designed the experiments in such a way that we could evaluate the impact on the classification results caused by the selection of different values of λ. In particular we used $\lambda = 1, 2, 3$ in order to assign equal, double or triple relevance to the neighbors information in relation to the information from the document itself.

In addition, with the purpose of analyzing the impact caused by the inclusion of non-relevant neighbors into the class assignment process, we also considered different number of neighbors; we used $N = 1...30$.

Experiment 1. The objective of this experiment was to analyze the performance of the proposed method in collections having small training sets, which complicate the construction of accurate classification models. Table 4 shows the F-measure values achieved by the proposed method in three collections using different values of λ and N. Results in bold indicate that the method significantly outperformed the baseline result. We evaluated the statistical significance of results using the z-test with a confidence of the 95%.

The obtained results show that the method could improve the classification performance in all collections, but especially in those having smaller training sets. For instance, for the R8-reduced-10 collection the improvement was as high as 9.7%. There is also important to mention that the method demonstrated not to be very sensitive to the values of λ and N, achieving –in general– the best results with $\lambda = 3$ and $N < 10$.

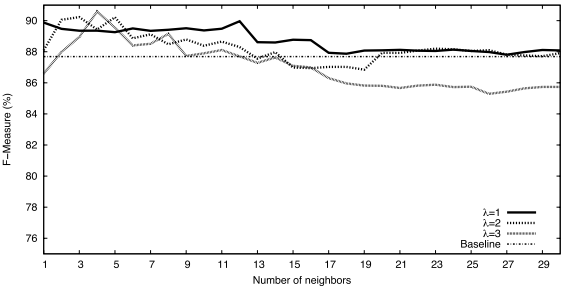
Experiment 2. The objective of this second experiment was to evaluate the performance of the method in a traditional classification scenario, having available enough training examples. In particular, we used the R8 collection which allowed to generate accurate classification models as shown in Section 4.3. Table 5 shows the results from this experiment, indicating in bold numbers the cases where the proposed method significantly outperformed the baseline result. In general, these results indicate that our method could also obtain satisfactory results with a larger training set. However, in this case, most relevant results were achieved using $\lambda = 1$.

Table 4. F-measure results of the proposed method on the three reduced datasets

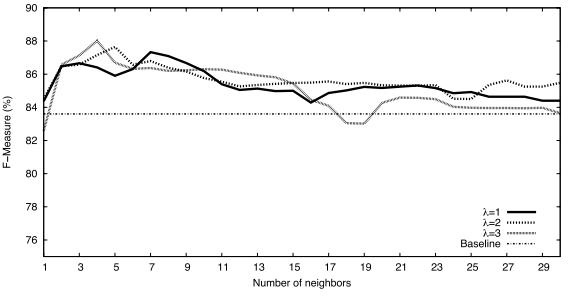
	R8-reduced-41 (baseline=0.836)			**R8-reduced-20** (baseline=0.803)			**R8-reduced-10** (baseline=0.767)		
N	$\lambda = 1$	$\lambda = 2$	$\lambda = 3$	$\lambda = 1$	$\lambda = 2$	$\lambda = 3$	$\lambda = 1$	$\lambda = 2$	$\lambda = 3$
1	0.843	0.845	0.825	0.813	**0.821**	0.804	**0.807**	**0.804**	0.780
2	**0.864**	**0.864**	**0.865**	0.819	**0.824**	**0.831**	**0.813**	**0.816**	**0.821**
3	**0.866**	**0.865**	**0.871**	**0.839**	**0.845**	**0.846**	**0.806**	**0.829**	**0.825**
4	**0.864**	**0.871**	**0.880**	**0.838**	**0.845**	**0.846**	**0.813**	**0.829**	**0.836**
5	**0.859**	**0.876**	**0.866**	**0.832**	**0.846**	**0.844**	**0.819**	**0.829**	**0.836**
6	**0.863**	**0.865**	**0.863**	**0.831**	**0.860**	**0.844**	**0.812**	**0.826**	**0.837**
7	**0.873**	**0.867**	**0.863**	**0.839**	**0.855**	**0.839**	**0.812**	**0.829**	**0.841**
8	**0.870**	**0.863**	**0.861**	**0.845**	**0.858**	**0.841**	**0.812**	**0.829**	**0.838**
9	**0.866**	**0.861**	**0.862**	**0.847**	**0.854**	**0.851**	**0.812**	**0.827**	**0.836**
10	**0.861**	**0.857**	**0.862**	**0.841**	**0.852**	**0.837**	**0.813**	**0.825**	**0.829**
11	**0.853**	**0.855**	**0.862**	**0.842**	**0.851**	**0.825**	**0.811**	**0.827**	**0.829**
12	0.850	**0.852**	**0.860**	**0.840**	**0.833**	**0.827**	**0.811**	**0.829**	**0.836**
13	0.851	**0.853**	**0.859**	**0.840**	**0.837**	**0.831**	**0.810**	**0.822**	**0.813**
14	0.849	**0.854**	**0.858**	**0.838**	**0.836**	**0.832**	**0.808**	**0.817**	**0.818**
15	0.849	**0.854**	**0.854**	**0.823**	**0.832**	**0.830**	**0.804**	**0.823**	**0.806**
16	0.842	**0.854**	0.844	**0.821**	**0.828**	**0.830**	**0.804**	**0.822**	**0.805**
17	0.848	**0.855**	0.840	**0.823**	**0.829**	**0.827**	**0.805**	**0.822**	**0.802**
18	0.850	**0.854**	0.830	**0.822**	**0.831**	0.819	**0.802**	**0.813**	**0.801**
19	**0.852**	**0.854**	0.830	0.817	**0.832**	0.818	**0.801**	**0.811**	**0.798**
20	0.851	**0.853**	0.842	0.816	**0.833**	**0.822**	**0.796**	**0.811**	**0.798**
21	**0.852**	**0.853**	0.845	0.816	**0.831**	**0.824**	**0.795**	**0.804**	**0.807**
22	**0.853**	**0.853**	0.845	0.814	**0.830**	**0.825**	**0.797**	**0.803**	**0.797**
23	0.851	**0.853**	0.844	0.816	**0.827**	**0.822**	**0.796**	**0.798**	**0.798**
24	0.848	0.845	0.840	0.806	0.817	**0.821**	**0.795**	**0.805**	**0.796**
25	0.849	0.845	0.839	0.805	0.817	0.819	**0.795**	**0.800**	**0.794**
26	0.846	**0.853**	0.839	0.807	0.818	**0.820**	**0.795**	**0.800**	**0.803**
27	0.846	**0.856**	0.839	0.807	0.817	**0.820**	**0.795**	**0.800**	**0.803**
28	0.846	**0.852**	0.839	0.811	0.819	**0.823**	**0.794**	**0.799**	**0.800**
29	0.843	**0.852**	0.839	0.810	0.820	**0.822**	**0.795**	**0.801**	**0.789**
30	0.843	**0.854**	0.836	0.810	0.820	**0.822**	**0.795**	**0.801**	**0.788**

Table 5. F-measure results of the proposed method on the R8 collection

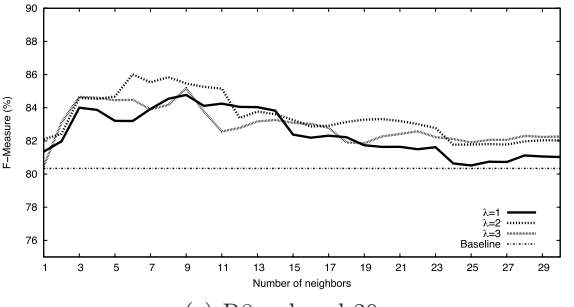
	(baseline=0.876)		
N	$\lambda = 1$	$\lambda = 2$	$\lambda = 3$
1	**0.898**	0.880	0.865
2	**0.894**	**0.900**	0.879
3	**0.893**	**0.902**	0.889
4	**0.893**	**0.894**	**0.905**
5	**0.892**	**0.902**	**0.895**
6	**0.895**	0.888	0.884
7	**0.893**	**0.891**	0.885
8	**0.894**	0.884	**0.891**
9	**0.895**	0.887	0.877
10	**0.893**	0.883	0.878
11	**0.894**	0.886	0.881
12	**0.899**	0.883	0.877
13	0.886	0.875	0.872
14	0.885	0.879	0.876
15	0.887	0.869	0.870
16	0.887	0.869	0.869
17	0.879	0.870	0.863
18	0.878	0.870	0.859
19	0.880	0.868	0.858
20	0.880	0.879	0.858
21	0.881	0.879	0.856
22	0.880	0.880	0.858
23	0.880	0.881	0.858
24	0.881	0.881	0.857
25	0.880	0.880	0.857
26	0.879	0.881	0.852
27	0.878	0.878	0.854
28	0.879	0.877	0.856
29	0.881	0.877	0.857
30	0.880	0.879	0.857



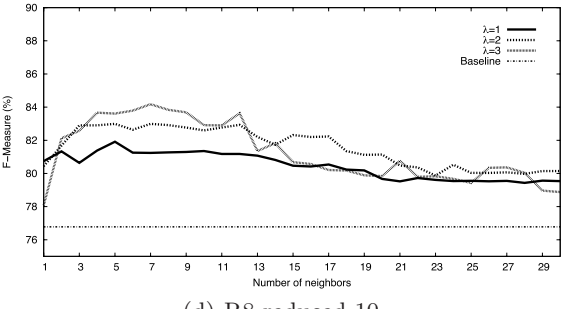
(a) R8



(b) R8-reduced-41



(c) R8-reduced-20



(d) R8-reduced-10

Fig. 3. Comparison of the results obtained in four collections using different values of λ and N

Comparative analysis of results. In order to get a better idea about the behavior of the proposed method, Figure 3 shows its performance in all collections using the different values of λ and N. From this figure it is possible to make the following observations regarding this method:

- It requires a relative small number of neighbors to achieve the highest performance value; in all collections it used less than 10 neighbors. Moreover, as intuitively expected, it is possible to notice that the lesser the number of training examples, the greater the number of neighbors required to achieve the maximum performance value. For instance, for R8 there were needed only four neighbors, whereas, for R8-reduced-10 there were seven.
- The lower the number of documents in the training set, the greater the improvement achieved by the proposed method (in comparison with the baseline). This fact demonstrates that this method is especially appropriate to be used with small training sets, where current approaches tend to generate poor classification models.
- In most cases the best results were achieved using $\lambda > 1$. Somehow this fact indicates that information from neighbors may be useful in practically any classification scenario, including or not sufficient training examples.

5 Conclusions

Inspired by the popular proverb "a man is known by the company he keeps", in this paper we proposed a new text classification method that carries out the classification of documents by considering not only their own content but also the information about the assigned category to their similar documents.

Experimental results in four collections with training sets of different sizes demonstrated the robustness of the proposed method, which could significantly outperformed the results from methods such as Naive Bayes, Support Vector Machines (SVM) and a traditional prototype-based classifier. In relation to this last point, it is important to point out that the proposed method, using only 2% of the labeled instances (i.e., R8-reduced-10), achieved a similar performance than Naive Bayes when it employed the complete training set (i.e., R8).

As described in Section 3, the proposed method has two main parameters: λ and N. Experimental results indicated that the method is not very sensitive to the selection of the value of these parameters. Nevertheless, it was observed that the lesser the number of training examples, the greater the values of λ and N required to achieve the maximum performance value.

As future work we plan to: (i) implement the proposed method using other algorithms as base classifiers such as Naive Bayes and SVM, (ii) evaluate the method in a cross-lingual text classification task as well as in a transfer-learning kind of problem, and (iii) define the influence function based on other similarity measures.

Acknowledgments

This work was done under partial support of CONACyT-Mexico (project grant 83459 and scholarship 239516).

References

1. Han, E.H., Karypis, G.: Centroid-based document classification: Analysis and experimental results. In: Zighed, D.A., Komorowski, J., Żytkow, J.M. (eds.) PKDD 2000. LNCS (LNAI), vol. 1910, pp. 424–431. Springer, Heidelberg (2000)
2. Derivaux, S., Forestier, G., Wemmert, C.: Improving supervised learning with multiple clusterings. In: Workshop on Supervised and Unsupervised Ensemble Methods and Their Applications in conjunction with ECAI, Patras, Greece, pp. 57–60 (2008)
3. Cong, G., Lee, W.S., Wu, H., Liu, B.: Semi-supervised text classification using partitioned em. In: Lee, Y., Li, J., Whang, K.-Y., Lee, D. (eds.) DASFAA 2004. LNCS, vol. 2973, pp. 229–239. Springer, Heidelberg (2004)
4. Nigam, K., Mccallum, A., Thrun, S., Mitchell, T.: Text classification from labeled and unlabeled documents using em. In: Machine Learning, 103–134 (1999)
5. Kyriakopoulou, A., Kalamboukis, T.: Using clustering to enhance text classification. In: SIGIR 2007: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 805–806. ACM, New York (2007)
6. Fang, Y.C., Parthasarathy, S., Schwartz, F.: Using clustering to boost text classification. In: Workshop on Text Mining, TextDM 2001 (2001)
7. Lertnattee, V., Theeramunkong, T.: Term-length normalization for centroid-based text categorization. In: Palade, V., Howlett, R.J., Jain, L. (eds.) KES 2003. LNCS, vol. 2774, pp. 850–856. Springer, Heidelberg (2003)
8. Guan, H., Zhou, J., Guo, M.: A class-feature-centroid classifier for text categorization. In: WWW 2009: Proceedings of the 18th international conference on World wide web, pp. 201–210. ACM, New York (2009)
9. Cardoso-Cachopo, A., Oliveira, A.L.: Semi-supervised single-label text categorization using centroid-based classifiers. In: SAC 2007: Proceedings of the, ACM symposium on Applied computing, pp. 844–851. ACM, New York (2007)
10. Tan, S.: An improved centroid classifier for text categorization. Expert Systems with Applications 35, 279–285 (2008)
11. Lewis, D.D.: Naive (bayes) at forty: The independence assumption in information retrieval, pp. 4–15. Springer, Heidelberg (1998)
12. Joachims, T.: Text categorization with support vector machines: Learning with many relevant features, pp. 137–142. Springer, Heidelberg (1998)

Rank Distance Aggregation as a Fixed Classifier Combining Rule for Text Categorization

Liviu P. Dinu and Andrei Rusu

University of Bucharest, Faculty of Mathematics and Computer Science,
14 Academiei, 010014, Bucharest, Romania
ldinu@funinf.cs.unibuc.ro, andrei.alex.rusu@gmail.com

Abstract. In this paper we show that Rank Distance Aggregation can improve ensemble classifier precision in the classical text categorization task by presenting a series of experiments done on a 20 class newsgroup corpus, with a single correct class per document. We aggregate four established document classification methods (TF-IDF, Probabilistic Indexing, Naive Bayes and KNN) in different training scenarios, and compare these results to widely used fixed combining rules such as Voting, Min, Max, Sum, Product and Median.

1 Introduction

1.1 Motivation

The issue of using multiple classification methods together (ensembles) to form a better classifier is a well researched problem and appears in a wealth of classical Machine Learning scenarios. Many examples come to mind, from combining the same classifier on different feature sets, or different classifiers on the same feature, to algorithms like bagging and boosting that have some combiner as a final decision maker. Also, the more general learning of complex decision boundaries (e.g. not lines, nor circles), by means of multiple classifiers, employs a combination or selection scheme to form a decision boundary shape that the individual classifiers (usually) cannot learn.

1.2 Background

The fields of Pattern Recognition and Machine Learning have reached a point where many approaches are available for all the usual stages of a categorization project. That is, for most of the real world applications, many feature extraction techniques have been proposed, tested and theoretically analyzed, which has lead to methodologies for data analysis being exported to different applications and projects. With diverse feature selection methods, extensive testing of many classifiers was possible. Most of the time though, more than one classifier–feature pair have proven "the best" precision or recall, and the nature of the supervised generalization problem has made choosing a clear cut winner, at least per application, particularly difficult.

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 638–647, 2010.

Since the early days of ML, combination schemes for pattern classifiers were proposed. From asking each classifier to vote for the right class, algebraic and statistically motivated combiners, to supervised learning combiners, these approaches to ensembles of classifiers have shown improved precision and increased reliability (see [9] for a survey). More and more real world problems are starting to benefit from ensembles trained on different features, or different partitions of the training data, in trying to achieve classifier diversity. This has triggered a shift in focus, from trying to create a "perfect" classifier, to combining collections of classification methods so that the information in the feature spaces is exploited to its full potential. This path has been field tested, see [4] for a comprehensive study on handwritten digit recognition, and the general guidelines for choosing between the different combination schemes have been laid out, see for example [4], [5], [10].

1.3 New to Our Approach

Usual ensemble methods put a considerable degree of emphasis on the numerical values outputted by different classifiers for each category. In theoretical settings they are processed and used as probabilities or as confidences. Different normalizing schemes are usually employed, from the straightforward division after summation, to sigmoid functions. After these steps the resulting values decide, in a fixed way, one winning class.

Our method discards these values and transforms the classifier outputs into rankings of class labels. This information is thereafter ignored, to the extent given by the positions in the rankings and the ranking lengths. Then we compute the rankings that are closest to all the base classifiers' outputs in terms of the rank-distance [1]. A set of rankings with this property is called the Rank Distance Aggregation of the original rankings. In the single correct class setting we simply take a vote among the aggregate rankings and output that class. If a tie between 2 or more classes appears, we select one at random, with equal probability. This is the usual voting fixed combiner on top of the aggregation set, instead of the base classifier outputs.

2 Fixed and Trained Combiners

Also known as classifier fusion, the problem of optimally using the available pattern classifiers and feature spaces together is considered to be well understood [9]. This does not mean settled, since many papers show different methods to be superior in certain settings, theoretical or applied, but no fusion rule has been proven universally better than others. One of the prominent debates refers to fixed vs. trained combiners. According to [5] fixed combiners are best just under very strict conditions and, in theory, they are sub-optimal if the underlying classifiers generate unreliable results. The same paper proposes trained combiners to overcome this problem. This view is not shared in [10], where the problem of imbalanced classifier fusion is analyzed. In practice, classifier–feature pairs usually exhibit very different accuracies, and this setting is exactly where trained

rules are claimed to outperform fixed fusion rules. However, the experiments in [10] show the improvements of two trained rules over the fixed combiners to be modest, even in "ideal" settings. This suggests that more careful study is necessary to identify the conditions where trained combiners really outperform fixed rules.

On the other hand, there is a consensus that careful training of the base classifiers and feature diversity are prerequisites for, and somewhat "guarantee" in practice, effective fusion with fixed combiners. By classifier diversity we mean considerably different decision boundaries that are still largely consistent with the training data, but generalize well. This means special care must be taken to avoid over-fitting and ensure reliable results. This is nothing out of the normal requirements of any categorization task with a single classifier, but the importance for the ensemble must be stressed, at least for fixed combiners. In the case of trained combiners the burden of careful training and over-fitting avoidance moves from the base classifiers to the combiner. This usually means further splitting of the training set and the theoretical superiority of trained combiners is hard to achieve in practice, since they are subject to the same weaknesses as the base classifiers, that is, too scarce training data, over-fitting or weak generalization. In real applications, like the handwritten digit recognition task, two layers of trained combiners were necessary to achieve truly superior, reliable, results. This means one layer of trained combiners over 6 feature sets and another "combining combiner" over the trained fusion classifiers. In our opinion this more than makes up for the complexity of just producing reliable results with the base classifiers (in the first place) and simply using just one fixed combining rule. Nevertheless, voting on the rank-distance aggregation set performed remarkably in this task as well, as reported in [3], which encouraged testing this approach in the setting of document classification.

3 Rank Distance Aggregation

The rank-distance metric was introduced by Dinu in [1] and we survey the key points in this chapter. For a more elaborate analysis and implementation suggestions see [2]; for the underlying algorithmic solutions, check [7], [6].

3.1 Rankings

A ranking is an ordered list of labels and can be viewed as the result of applying an ordering criterion to a set of objects. Rankings appear naturally in many selection processes, where votes can be assigned by simply selecting entities in a specific, subjective order of preference. This is widely encountered in real life settings like competitions, or public opinion surveys. The underlying subjective criteria for creating rankings can be very different, and might not even be applicable to all the existing objects. Therefore rankings usually account for a small number of all the possible objects and a longer ranking suggests a more thorough criterion. With this intuition in mind we will give a formal description of rankings and the rank-distance, following [2].

3.2 Rank Distance

Let $\mathcal{U}$ be a finite set of objects, called universe. Let's assume, without loss of generality, that $\mathcal{U} = \{1, 2, ..., \#\mathcal{U}\}$ (where $\#\mathcal{U}$ denotes the cardinality of $\mathcal{U}$). A ranking over $\mathcal{U}$ is an ordered list: $\tau = (x_1 > x_2 > ... > x_d)$, where $x_i \in \mathcal{U}$ for all $1 \leq i \leq d$, $x_i \neq x_j$ for all $1 \leq i \neq j \leq d$, and $>$ a strict ordering relation on the set $\{x_1, x_2, ..., x_d\}$. A ranking defines a partial function on $\mathcal{U}$ such that, for each object $i \in \mathcal{U}$, $\tau(i)$ represents the position of object i in ranking τ. It is worth noticing that highly ranked objects in τ have the lowest positions.

The rankings that contains all the objects of universe $\mathcal{U}$ are called full rankings. All others are partial rankings. If n is the length of a ranking $\sigma = (x_1 > x_2 > ... > x_n)$, then $\forall x \in \mathcal{U} \cap \sigma$ we define the order of object x in ranking σ by $ord(\sigma, x) = |n - \sigma(x)|$. By convention, if $x \in \mathcal{U} \setminus \sigma$, we have $ord(\sigma, x) = 0$.

Definition 1. *Given two partial rankings σ and τ over the same universe $\mathcal{U}$, we define the rank-distance between them as:*

$$\Delta(\sigma, \tau) = \sum_{x \in \sigma \cup \tau} |ord(\sigma, x) - ord(\tau, x)|.$$

Since for all $x \in \mathcal{U} \setminus \sigma$ we have $ord(\sigma, x) = ord(\tau, x) = 0$, the following holds:

$$\begin{aligned}
\Delta(\sigma, \tau) &= \sum_{x \in \sigma \cup \tau} |ord(\sigma, x) - ord(\tau, x)| \\
&= \sum_{x \in \sigma \cup \tau} |ord(\sigma, x) - ord(\tau, x)| \\
&+ \sum_{x \in \mathcal{U} \setminus (\sigma \cup \tau)} |ord(\sigma, x) - ord(\tau, x)| \\
&= \sum_{x \in \mathcal{U}} |ord(\sigma, x) - ord(\tau, x)|.
\end{aligned}$$

Theorem 1. *Δ is a distance function.*

Proof. See [1].

The motivation behind using orders instead of ranking positions is based on the intuition that ranking differences on the highly ranked objects should have a larger inpact on the overall distance than disagreements on the lower ranked objects. Secondly, the length of the raking is not discounted. This complies with the intuition that longer rankings are produces by more thorough criteria, although it puts extra pressure on base rankings; this means longer hierarchy must be justified, with the benefit of gaining extra expressivity.

Computing the rank-distance (RD) of two rankings is straight-forward and linear in the number of objects in the two rankings. This number is small in many practical applications, much lower than the total number of universe objects ($n = \#\mathcal{U}$). When implemented as random access arrays indexed by the universe objects, the rank-distance computation has complexity $\mathcal{O}(n)$ in the worst case.

3.3 Rank Aggregation

In a selection process rankings are issued for a common decision problem, therefore a ranking that "combines" all the original (base) rankings is required. One commonsense solution is finding a ranking that is as close as possible to all the particular rankings. Apart from many paradoxes of different aggregation methods, this problem is NP-hard for most non-trivial distances.

Formally, the result of all the individually considered selection criteria is a finite collection of, not necessarily different, (partial) rankings, that we will call a ranking multiset $\mathcal{T} = \{\tau_1, \tau_2, ..., \tau_k\}$. When aggregating $\mathcal{T}$ into a single ranking we are looking for a σ with a minimal rank distance to all the rankings of the multiset; since Δ takes only positive values, we have to minimize the sum:

$$\Delta(\sigma, \mathcal{T}) = \sum_{\tau \in \mathcal{T}} \Delta(\sigma, \tau).$$

Definition 2. *Let $\mathcal{T} = \{\tau_1, \tau_2, ..., \tau_k\}$ be a multiset of rankings over object universe $\mathcal{U}$. A rank-distance aggregation (RDA) of $\mathcal{T}$ is a ranking σ (over the same universe $\mathcal{U}$) that minimizes $\Delta(\sigma, \mathcal{T})$. We denote the set of RD aggregations by $agr(\mathcal{T})$.*

A partition of the set $agr(\mathcal{T})$ by ranking length will give an effective means of computation.

Definition 3. *Let $1 \leq t \leq \#\mathcal{U}$ and $\mathcal{T}$ a multiset of rankings over $\mathcal{U}$. A partial ranking τ of length t that minimizes $\Delta(\tau, \mathcal{T})$ among all other rankings of length t is said to be a t-aggregation of multiset $\mathcal{T}$.*

Obviously, for any $1 \leq t \leq \#\mathcal{U}$ there exists at least one t-aggregation σ, with an associated minimal distance $d_t = \Delta(\sigma, \mathcal{T})$. To compute $agr(\mathcal{T})$ it is sufficient to compute the minimal distance:

$$d_{min} = \min_{1 \leq t \leq \#\mathcal{U}} \{d_1, d_2, ..., d_{\#\mathcal{U}}\},$$

and the set of indices:

$$D = \{s | d_s = d_{min}, 1 \leq s \leq \#\mathcal{U}\}.$$

Also, for any s-aggregation $\sigma \in agr(\mathcal{T})$, all other s-aggregations are in $agr(\mathcal{T})$, since, for a fixed integer s, all other s-aggregations have (by definition) the same minimal distance to $\mathcal{T}$ as σ. We can now summarize this approach:

Algorithm 1. *Let $\mathcal{T} = \{\tau_1, \tau_2, ..., \tau_k\}$*
1: for $t = 1$ to $\#\mathcal{U}$ then
2: compute a t-aggregation of $\mathcal{T}$, namely π_t;
3: end
4: let $d_{min} = \min_{1 \leq s \leq \#\mathcal{U}} \Delta(\pi_s, \mathcal{T})$;
5: for all t such that $\Delta(\pi_t, \mathcal{T}) = d_{min}$ then
6: compute and output all the t-aggregations of $\mathcal{T}$;
7: end

Computing the closest t-aggregation to $\mathcal{T}$ in line 2 is equivalent to finding one solution for a certain assignment problem / minimal weight bipartite perfect matching, which has complexity $\mathcal{O}(n^3)$, where n is the number of individual objects mentioned in at least one of the rankings. Line 6 is equivalent to enumerating all minimal weight perfect matchings in a certain bipartite graph, see [7], for example. Therefore, the total time needed [6] is $\mathcal{O}((2x+2)n^4)$, where x is the number of existing aggregate rankings (i.e. rankings with the minimal RD to the base set). See [2] for further details.

Example 1. Let $\mathcal{T}$ be the following multiset of rankings over the universe of objects $\mathcal{U} = \{1, 2, 3, 4\}$:

$$\mathcal{T} = \{(1 > 2 > 3), (3 > 4), (1 > 3 > 2 > 4)\}.$$

The RDA of $\mathcal{T}$ is the set:

$$agr(\mathcal{T}) = \{(1 > 2 > 3), (1 > 3 > 4), (1 > 3 > 2 > 4)\}.$$

Notice that all the 3&4-aggregations are present in $agr(\mathcal{T})$. The 1&2-aggregations have larger distances to $\mathcal{T}$, so all are excluded. Also important to note is that $agr(\mathcal{T})$ is not necessarily a subset of $\mathcal{T}$, a desirable rationality condition for an aggregation method, known as "absence of dictator". Other rationality conditions verified by the rank-distance aggregation are Pareto optimality, reasonableness, stability, loyalty, inversion and free order [1].

3.4 Rank Distance Categorization

Now we can formally introduce *Rank Distance Categorization* (RDC) method.

Let d be a pattern , $C = \{c_1, c_2, \ldots, c_m\}$ be a set of all m possible categories of d, and $l_1, l_2, \ldots, l_n$ be n classifiers .

Each classifier gives a ranking of classes; let $\mathcal{L}$=$L_1, L_2, \ldots, L_n$ be the multiset of the individual rankings obtained by applying the previous n classifiers.

Let $agr(\mathcal{L})$=$\{A_1,\ A_2, \ldots, A_k\}$ be the aggregation of the multiset $\mathcal{L}$.

Definition 4. *The class of the object predicted by the RDC method is the one that occupies most frequently the first position in the rankings* $A_1, .., A_k$.

Example 2. Set the following sequence of 5 rankings: $\mathcal{L} = \{(1 > 2 > 3), (1 > 2 > 3), (3 > 1 > 2), (2 > 3 > 1), (2 > 3 > 1)\}$.
We have: $agr(\mathcal{L}) = \{(1 > 2 > 3), (2 > 1 > 3), (2 > 3 > 1)\}$. So, the class predicted by RDC method is the class 2.

In other words, RDC is a voting method on rank distance aggregations.

3.5 Properties of RDC

In the following we skip the proof of propositions; the proofs can be read in [3].

Proposition 1. *If all classifiers predict a class s on the first place for a given object d, then the RDC method will predict class s on first place.*

If we apply our method only for two objects (e.g. in binary classification), then we obtain the same results as the simple majority method. This is a consequence of the fact that RDA satisfies the reasonability condition [1].

We are also interested in the behavior of RDC method when a ranking has majority in the multiset $\mathcal{L}$. The following proposition help us to investigate this problem:

Proposition 2. *If in multiset $\mathcal{T}$ of k rankings there is a ranking $\sigma \in \mathcal{T}$ which appears in a majority of times, then σ is in $agr(\mathcal{T})$.*

This proposition says only that the $\sigma \in agr(\mathcal{T})$, which does not necessary imply that the class predicted by RDC is the object placed in first position in σ, as we can see from the following example:

Example 3. Let $\mathcal{T} = (2,4,5),(2,3,5),(2,3,5),(1,2),(1,2),(1,2),(1,2),(3,2,5,4)$ be a multiset of 8 rankings. $agr(\mathcal{T}) = \{(2),(2,5),(2,1),(1,2),(2,1,5)\}$, so the class predicted by RDC is 2.

Proposition 3. *If in multiset $\mathcal{T}$ of k rankings there is a ranking $\sigma \in \mathcal{T}$ which appears by at least a majority of times plus one, then $agr(\mathcal{T}) = \{\sigma\}$, and, by consequence, the element placed in first position of σ is the class predicted by RDC.*

An immediate consequence of previous proposition is that if we have a multiset with an odd number of rankings and one of this rankings appears by a majority of times, then the element placed in its first place is the class predicted by RDC.

4 Experiments in Text Categorization

We chose to conducted our experiments with fixed combining rules on top of the *libbow* library [8], available on most Unix systems, including Linux, Solaris, SUNOS, Irix and HPUX. We used the categorization tool *rainbow* and the *20_newsgroups* corpus, both provided by the library's development team. The corpus consists of 20.000 newsgroup articles, uniformly distributed across 20 classes. The *rainbow* text classification tool supports Naive Bayes, TFIDF/Rocchio, Probabilistic Indexing and K-nearest neighbor, 4 established text categorization methods, with well known favorable settings and shortcomings, which also provide reliable results under certain conditions.

For our purposes, the universe objects are these 20 assignable classes, and the classifier outputs are transformed into rankings of class labels. There is a "pruning" phase, where values outputted by the classifiers for each document–class pair are rescaled per document and sorted in descending order. The most probable class is put first on the ranking. After that, only values which make up 60% or more of the previous class' probability are added to the resulting ranking.

Classifiers	2pc	5pc	10pc
TFIDF	**79.23**	70.46	**93.10**
PRIND	42.56	56.76	71.30
KNN	71.90	74.86	75.36
NBAYES	75.23	76.26	92.53
Voting	75.50	77.96	91.69
Product	75.50	77.00	**92.73**
Sum	74.90	**81.09**	**92.66**
Max	75.06	**80.79**	92.56
Min	74.13	72.80	85.60
Median	76.96	76.13	**92.76**
Voting on RDA	76.23	77.06	91.86

Fig. 1. Precision(%). **Underlined** is the maximum, **bold** is everything closer than 0.50% to the maximum.

This method is empirically consistent with the requirement that longer rankings are produced only when justified by the underlying criterion. The actual aggregation is done locally, on the 4 rankings available for each test document. The number of involved objects is much smaller than 20, usually less than 5 classes are competing for the first place. This fact makes the aggregation problem computationally trivial by today's resources, such that fast and parallel aggregation for thousands of documents is possible in a very short time.

The number of training documents, as well as how representative they are statistically, are very important parameters for supervised categorization methods in general, with severe performance penalties. This means that, in many real life situations, much less information is available to these methods than required for descent classification precision. To be fair, we have chosen 7 training settings. Each setting consists of N random documents per class for training the classifiers (on the same documents), and 500 documents (per class) for testing, where:

$$N \in \{2, 5, 10, 20, 50, 100, 500\}.$$

These settings are consistent with the Reuters-21578 collection, which is challenging also because of the large number of under-sampled classes, and the more than 90 topics.

As shown in figure 1, if the number of training documents is relatively small, the base classifiers produce unreliable results, some more than others. In this scenario some of the base classifiers overtake the aggregations, as expected. Aggregating in these settings is also referenced in the literature as unbalanced classifier fusion.

On the other hand, if the training set is sufficiently large (or if special care is taken in the training process), the aggregations usually do better than the individual classifiers, as seen in figure 2. In this case Voting on RDA outperforms the other fixed fusion rules in all 4 training scenarios. What is remarkable is that Voting on RDA manages to outperform all the other methods, although

Classifiers	20pc	50pc	100pc	500pc
TFIDF	**92.83**	91.53	**91.63**	**91.76**
PRIND	77.19	82.86	83.86	86.86
KNN	81.83	89.16	89.83	88.96
NBAYES	91.63	91.19	91.03	**92.00**
Voting	92.00	**92.09**	**91.93**	**92.16**
Product	92.26	**92.06**	91.56	91.40
Sum	**92.46**	91.66	91.33	**92.30**
Max	91.36	91.40	91.00	**91.96**
Min	86.36	88.93	90.60	91.70
Median	91.96	91.39	90.96	**92.23**
Voting on RDA	**92.66**	**92.56**	**92.16**	**92.40**

Fig. 2. Precision(%). **Underlined** is the maximum, **bold** is everything closer than 0.50% to the maximum.

there is significant precision fluctuations in the base classifiers over the 4 different scenarios, suggesting increased reliability. Interestingly enough, voting on RDA outperforms plain voting, by as much as 0.66% which means 33 additional documents classified correctly. In an application that requires extreme precision, like person identification in security application, this is significant, since false positives can result in unauthorized access by coincidence or fake credentials, like fake fingerprints.

5 Summary and Conclusions

This article presents a series of experiments with text categorization methods, combined by the common, fixed, classifier fusion rules and by the new voting on the rank-distance aggregation set. The categorization task (on the *20_newsgroup* corpus) features 20 assignable classes and 10.000 documents for testing (500 per class). We use the *rainbow* Unix document classification tool to output the results of 4 different text categorization methods, and we aggregate by 6 established fixed fusion rules. We compare these results with Voting on the Rank Distance Aggregation set, which demonstrates superior precision over all the fixed rules tested.

Acknowledgments

Research supported by CNCSIS, PNII-Idei, project 228.

References

1. Dinu, L.P.: On the classification and aggregation of hierarchies with difererent constitutive elements. Fundamenta Informaticae 55(1), 39–50 (2002)
2. Dinu, L.P., Manea, F.: An eficient approach for the rank aggregation problem. Theoretical Computer Science 359(1), 455–461 (2006)

3. Dinu, L.P., Popescu, M.: A multi-criteria decision method based on rank distance. Fundamenta Informaticae 86(1-2), 79–91 (2008)
4. Duin, R.P.W., Tax, D.M.J.: Experiments with classifier combining rules. In: Kittler, J., Roli, F. (eds.) MCS 2000. LNCS, vol. 1857, pp. 16–29. Springer, Heidelberg (2000)
5. Duin, R.P.W.: The combining classifier: To train or not to train? Pattern Recognition 2 (2002)
6. Fukuda, K., Matsui, T.: Finding all minimum cost perfect matchings in bipartite graphs. Networks 22, 461–468 (1992)
7. Manea, F., Ploscaru, C.: A generalization of the assignment problem, and its application to the rank aggregation problem. Fundamenta Informaticae 81(4), 459–471 (2007)
8. McCallum, A.K.: Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering (1996), http://www.cs.cmu.edu/~mccallum/bow
9. Polikar, R.: Ensemble based systems in decision making. IEEE Circuits and Systems Magazine 6(3), 21–45 (2006)
10. Roli, F., Fumera, G.: Fixed and trained combiners for fusion of imbalanced pattern classifiers. In: Proc. 5th Int. Conf. on Information Fusion, pp. 278–284 (2002)

The Influence of Collocation Segmentation and Top 10 Items to Keyword Assignment Performance

Vidas Daudaravicius

Vytautas Magnus University
Vileikos 8, Kaunas, Lithuania
vidas@donelaitis.vdu.lt

Abstract. Automatic document annotation from a controlled conceptual thesaurus is useful for establishing precise links between similar documents. This study presents a language independent document annotation system based on features derived from a novel collocation segmentation method. Using the multilingual conceptual thesaurus EuroVoc, we evaluate filtered and unfiltered version of the method, comparing it against other language independent methods based on single words and bigrams. Testing our new method against the manually tagged multilingual corpus Acquis Communautaire 3.0 (AC) using all descriptors found there, we attain improvements in keyword assignment precision from 18 to 29 percent and in F-measure from 17.2 to 27.6 for 5 keywords assigned to a document. The further filtering out of the top 10 frequent items improves precision by 4 percent and collocation segmentation improves precision by 9 percent on the average, over 21 languages tested.

Keywords: collocation segmentation, top 10 items, multilinguality, keyword assignment, stop-word list.

1 Introduction

The multilingual documentation in our Information Society has become a common phenomenon in many official institutions, private companies and on World Wide Web. This textual information needs to be categorised and it could be done through a keyword assignment. A keyword assignment is the identification of appropriate keywords from the controlled vocabulary of a reference list (*a thesaurus*). Controlled vocabulary keywords, which are usually referred to as descriptors, are therefore not necessarily present explicitly in the text. Within this particular context, some monolingual and cross-lingual indexing systems were developed [10], [11], [2] and [3] through multilingual thesaurus, as EuroVoc [17].

A keyword assignment statistical system uses a training corpus of manually indexed documents to produce, for each descriptor, a feature vector of words whose presence in a text indicates that the descriptor may be appropriate for this text. Generally, feature vectors are associated with single words [11], bigrams or

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 648–660, 2010.

trigrams [2]. A frequency approach [19:32-34], [10] could be used to create multi-word feature vectors from a corpus. This approach takes word combinations with a frequent occurrence. A straightforward implementation simply examines all combinations of up to the maximum length of n-gram. Clearly, the number of features grows significantly when longer multi-word features are considered. This approach is not effective for low-frequency words [13]. A novel method for an efficient (single-) multi-word unit identification is presented in this paper. This method uses collocation extraction approach [4] to segment a text into small pieces.

Stop-word lists are used to create more consistent feature vectors, and influence keyword assignment performance. The general trend in IR systems has been from standard use of large stop lists (200-300 terms) to very small stop lists (7-12 terms) to no stop lists whatsoever [9:26]. [11] describes a tuned stop-word list for Spanish of 1533 items. This list helped to improve keyword assignment precision from 40.3 to 45.6 percent. In this multilingual study we were not able to gather stop-word lists for twenty one languages. Therefore, we introduce the top 10 frequent words method to alter-nate the stop-word list. The study in this paper shows that the top 10 frequent items can improve keyword assignment in a similar way as a well tuned stop-word list.

The structure of this paper is as follows. Section 2 introduces the Conceptual Thesaurus EuroVoc. Section 3 is devoted to the novel language independent statistical collocation segmentation method. Later, in Section 4, the data sets and a preprocessing of data are presented. The Top 10 frequent items principle is introduced and the process of producing feature vectors and assigning descriptors to a text is described in Section 5. The results of assigning descriptors to a text are presented in Section 6. Finally, some conclusions are stated in Section 7.

2 Conceptual Thesaurus

Conceptual thesaurus (CT) is a list of descriptors that are relatively abstract, conceptual terms. An example for a CT is EuroVoc [17], whose descriptors describe the main concepts of a wide variety of subject fields by using high-level descriptor terms. This thesaurus contains more than 6000 descriptors. EuroVoc covers diverse fields such as politics, law, finance, social questions, science, transport, environment, geography, organisations, etc. The thesaurus is translated into the most of EU official languages. EuroVoc descriptors are defined precisely, using scope notes, so that each descriptor has exactly one translation into each language. Most of EU official documents have been manually classified into subject domain classes using the EuroVoc thesaurus. For instance, EU document with CELEX[1] number 22005X0604(01) contains following manually assigned descriptors: EC agreement, interest, Monaco, savings, tax on income, tax system. This document will be used to present examples for collocation segmentation and automatic descriptor assignment in this paper.

[1] CELEX document number is the identification number of EU legislation/law document which provides direct access to a specific document.

3 Collocation Segmentation

The Dice score is used to measure the association strength of two words. This score is used, for instance, in the collocation compiler XTract [13] and in the lexicon extraction system Champollion [14]. Dice is defined as follows:

$$Dice(x;y) = \frac{2 \cdot f(x;y)}{f(x) + f(y)}$$

$f(x;y)$ being the frequency of co-occurrence of x and y, and $f(x)$ and $f(y)$ are the frequencies of occurrence of x and y anywhere in the text. If x and y tend to occur in conjunction, their Dice score will be high. There are many other association measures such as Mutual Information (MI), T-score, Log-Likelihood and etc. The most widely used measure is MI. MI and Dice scores are almost similar in the sense of distribution of values [4]. The values of MI grow together with the size of a corpus, while the Dice score is not sensitive to the corpus size and score values are always between 0 and 1. A threshold is used often to define strong and weak associativities. MI threshold level depends on the corpus size. Therefore, it is more useful to use Dice score just for the practical reasons: the threshold level moves slowly compare to the corpus size. Thus, in practice we need to define the threshold of Dice score only once for many different corpora. For instance, we found that the threshold is very close and could be seen as the constant for a corpus of the 500 thousand words and for a corpus 100 million words. We propose to use the score to produce a discrete signal of a text. Thus, a text is seen as a changing curve of Dice values (see Figure 1). The collocation segmentation is the process of detecting the boundaries of collocation segments within a text. A collocation segment is a piece of a text between boundaries. The boundaries are set following. First, we set the boundary between two words within a text where the Dice value is lower than a threshold. The threshold value is set manually and is kept as low as possible. We set the threshold at the Dice value of 0.0003355 in our experiment. The logarithm of this value is equal to minus 8. This decision was based on the shape of the curve found in [4] and by our manual evaluation. The threshold is used to set 'strong' boundaries. Second, we introduce an average minimum law (AML). The average minimum law is applied to the three adjacent Dice values (i.e., four words). The law is expressed as follows:

$$\frac{Dice(x_{-2};x_{-1}) + Dice(x_0;x_1)}{2} < Dice(x_{-1};x_0) \Rightarrow setboundary(x_{-1},x_0)$$

The boundary between two words within a text is set where the Dice value is lower than the average of preceding and following Dice values. The example of setting the boundaries for English sentence is presented in Figure 1, and it shows a sentence and Dice values between words.

The proposed segmentation method is new and different from other widely used statistical methods for collocation extraction [18]. For instance, the general method used by Choueka [1] is the following: for each length n, $(1 \leq n \leq 6)$,

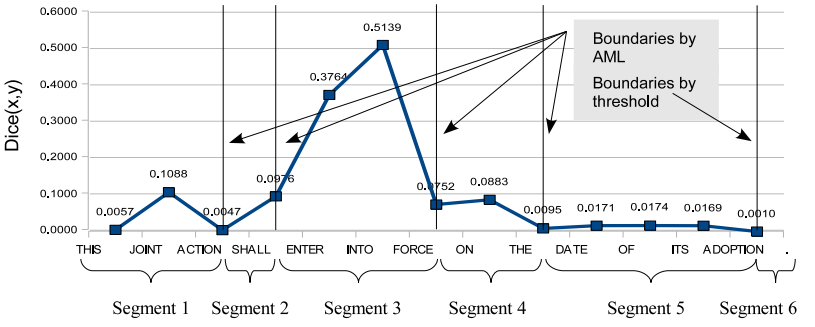


Fig. 1. The segmentation of the English sentence. Threshold is set to 0.0015, AML applied.

information_relating to_the **entry** into_force **of_the** agreement_between **the** european_community **and_the** principality_of **monaco_providing_for** measures **equivalent** to **those** laid_down **in** council_directive **2003_/** 48 **/_ec** on **taxation_of** savings_income **in_the** form_of **interest_payments (_2005_/_c** 137 **/_01_)**
the_agreement_between the **european_community** and_the **principality_of** monaco_providing_for **measures** equivalent **to** those **laid_down** in **council_directive** 2003_/ **48** /_ec **on** taxation_of **savings_income** in_the **form_of** interest_payments **will** enter_into_force **on** 1 **july_2005** ._the **procedures** provided_for **in_article** 16 **of_the** agreement **having** been_completed **on_31_may_2005_.**

Fig. 2. The collocation segmentation of the full text of 22005X0604(01) document

produce all the word sequences of length n and sort them by frequency; impose a threshold frequency 14. Xtract is designed to extract significant bigrams, and then expands 2-Grams to N-Grams [13]. Lin [8] extends the collocation extraction methods with syntactic dependency triples. These collocation extraction methods are performed on a dictionary level. The result of this process is a dictionary of collocations. Our collocation segmentation is performed within a text and the result of this process is the segmented text (see Figure 2). The segmented text could be used later to create a dictionary of collocations. Such dictionary accepts all collocation segments. The main difference from Choueka and Smadja methods is that our proposed method accepts all collocations and no significance tests for collocations are performed. On the other hand, the disadvantage of our method is that the segments do not always conform to the correct grammatical and lexical phrases. The linguistic nature of the segments in most cases is a noun phrase (*all relevant information*), a verb phrase (*are carried out*), a prepositional phrase (*by mutual agreement*), a noun or verb phrase with preposition at the end (*additional information on*, *carried out by*), the combination of a preposition and an article (*in the*). Also, it is important to notice that the collocation segmentation of the same translated text is similar for different languages, even if a word or phrase order is different. The same sentence from AC corpus jrc32005E0143 document in nineteen languages is segmented and the result is shown in Figure 3. This collocation segmentation study will be extended in near feature to evaluate the conformity of the proposed collocation segmentation method to phrase based segmentation by using parsers.

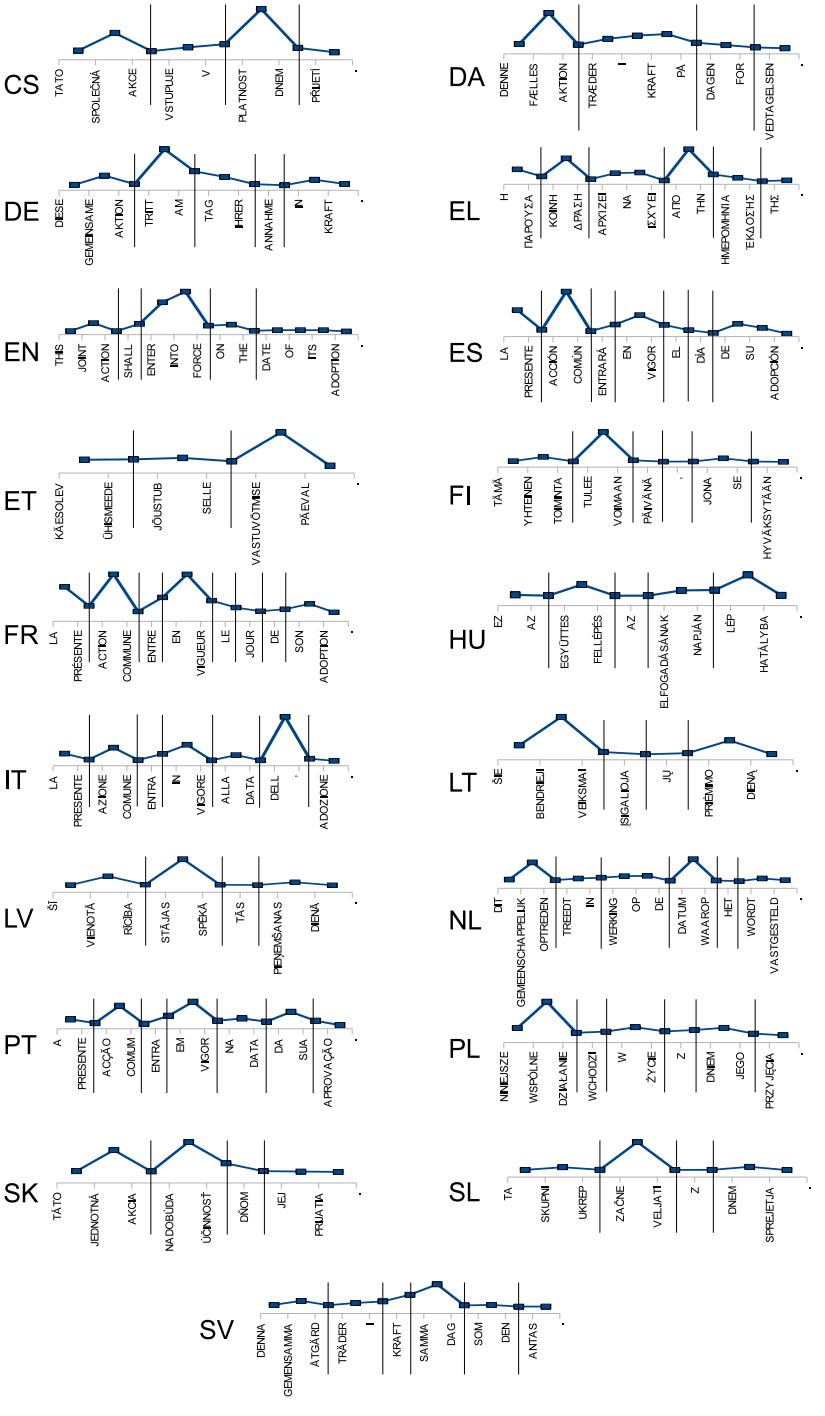


Fig. 3. The segmentation of the same sentence in nineteen languages

4 Data Sets

Experiments were carried out on the Acquis Communautaire (AC) corpus [15]. This large text collection contains documents selected from the European Union (EU) legislation in all the official EU languages. Most of these documents have been manually classified according to the EuroVoc thesaurus [17]. In this study we take only those documents that have assigned EuroVoc descriptors. We did not found any Romanian documents with assigned descriptors. Therefore we were not able to include Romanian language in our study. The AC documents are XML marked. The body parts of the documents were used in our study. Annexes and Signatures were excluded. Also, the minimum of word list size for each document was set to 20 types. This conforms to 200 characters on the average. A few documents were excluded by this requirement. After applying these requirements the average number of documents per language was 18568.

The corpus was split into the development part and test part. For each language 2/3 of documents were randomly selected for the development corpus and the rest of the documents were used for the test corpus. There is no overlap between the development corpus and the test corpus.

No requirements were applied for descriptors to occur in several documents. The average number of descriptors per language in the development corpus was 3481 and in the test corpus it was 2650. This is three times more than in [2] and similar in [11]. The average number of descriptors per document was 5.65.

We tried to keep the linguistic effort minimal in order to be able to apply the same algorithm to all twenty one languages for which we have training material. Before we trained and tested our text classifier, the AC corpus underwent a basic preprocessing consisting of lower-casing and isolation of the punctuation marks. The numbers were left as is. We decided not to apply any language-dependent preprocessing, such as lemmatisation, since our study is intended to work in a multilingual environment. The linguistic preprocessing might have improved the accuracy of classifiers [11]. An average number of tokens per language of the preprocessed corpus is 19.2 million in the development. This number in development corpus is three times smaller than in [11]. Thus, we expect lower base performance for keyword assignment in our study. An average number of types per language is 252 thousand in the development.

The collocation segmentation was performed for each language corpora. An average number of tokens or segment tokens per language decreased from 19.2 million to 11.6 million in the development corpus. The segmentation increased the number of segment types by 5 times. The average size of the dictionary of collocation segments is 1.3 million in the development corpus. The number of the types is at least double for highly inflected languages compare to almost non-inflected languages. For instance, the ratio of the word dictionary size for Finnish and English is 2.56. The segmentation increased the dictionary size of each language considerably. The increase of the dictionary size is smaller for highly inflected languages and bigger for non-inflected languages. Thus, the dictionaries of segments become more comparable than dictionaries of words. For instance, the ratio of the segment dictionary size for Finnish and English is 1.7.

Table 1. The top frequency words, bigrams and segments for a descriptor *Transfer pricing*

rank	type	frequency	segment type	frequency	bigram type	frequency
1	**the**	**75**	**of the**	**16**	**- -**	**49**
2	**-**	**55**	**in the**	**16**	**transfer pricing**	**18**
3	**of**	**48**	**member states**	**14**	**of the**	**17**
4	**,**	**39**	**transfer pricing documentation**	**12**	**in the**	**16**
5	**to**	**31**	**,**	**11**	**member states**	**16**
6	**and**	**30**	**european union**	**8**	**pricing documentation**	**12**
7	**in**	**30**	**and**	**8**	**the european**	**10**
8	**.**	**27**	**associated enterprises**	**7**	**code of**	**9**
9	**documentation**	**24**	**for**	**7**	**of conduct**	**9**
10	**transfer**	**20**	**code of**	**6**	**european union**	**9**
11	member	19	documentation	6	considering that	8
12	a	19	considering that	6	associated enterprises	8
13	pricing	18	partially centralised	5	to the	7
14	for	16	of	5	standardised and	6
15	states	16	internal market	5	enterprises in	6
16	that	15	standardised	5	for associated	6
17	enterprises	13	the	4	. member	6
18	)	11	for the	4	internal market	6
19	code	11	.	4	documentation for	6
20	(	11	to the	4	the member	5
21	european	10	eu tpd	4	eu tpd	5
22	on	10	(	3	and partially	5
23	union	9	code of conduct	3	the council	5
24	conduct	9	% quot %	3	partially centralised	5
25	considering	8	conduct	3	' s	5
26	%	8	joint	3	the commission	4
27	associated	8	contained	3	states should	4
28	eu	8	conduct on	3	centralised transfer	4
29	not	8	having regard	3	the internal	4
30	market	7	, and	3	contained in	4

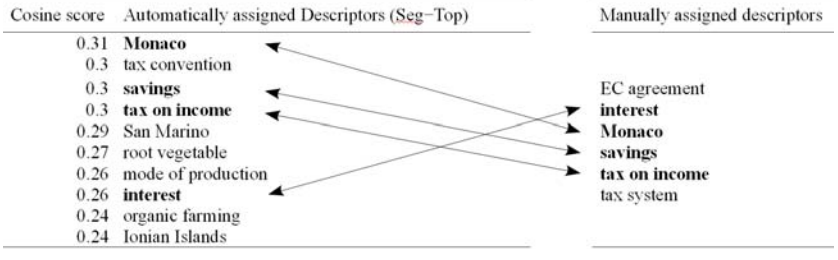


Fig. 4. Manually and automatically assigned descriptors for 22005X0604(01) document

5 Producing Feature Vectors and Assigning Descriptors to a Text

We used minimum requirements and parameters to produce feature vectors. We did not use any restrictions for descriptors and for words or segments to be used in several documents of the corpus. All descriptors and all words or segments were taken into consideration in order to study the influence of the collocation segmentation and the top 10 frequent items to the performance of keyword assignment. Feature vectors are produced on the basis of one large meta-text per descriptor (concatenation of all texts indexed with this descriptor), and a feature value is a raw frequency.

A fundamental tool in text classification is a list of stop-words that are unlikely to assist in classification and hence are deleted. The general strategy for determining a stop list is to sort the terms by frequency (the total number of

times each term appears in a corpus), and then to take the most frequent items, often hand-filtered and domain dependent. A widely used semi-manual approach is described in [6]. In our multilingual study we were not able to gather stop-word lists for twenty one languages. Therefore, we introduce the Top 10 frequent items method to alternate the stop-word lists. In the paper we want to test this method for conformity with stop list, and our results show that the proposed method is effective. The method could be useful in multilingual or multi-domain tasks when ready to use stop lists are not available. We should notice that the Top 10 method does not generate stop-word list. Instead, the method takes the Top 10 frequent items from a feature vector, and this Top 10 list is used for that particular feature vector only. The top 10 features differ from list to list as the rank changes in feature vectors. The examples of the Top 10 items of feature vectors are shown in Table 1. Many of the top 10 features of each feature vector are good candidates for a stop-word list. Thus, we use the Top 10 features as stop-words for the particular feature vector. The automatic extraction of stop list (by the determination of the number of the most frequent items to take) is not reliable because the frequency lists are very different in size from language to language, from domain to domain. We found only a few studies on automatic extraction and evaluation of stop lists [7]. Stop-words are widely used but such lists are usually extracted manually or semi-manually. For interest, we took the Top 10 features from the feature vectors of descriptors, and we counted the total number of the distinct features. The average number for the different languages of such stop-word candidates for the Base experiment is 548, for the Bigram experiment is 3596, and for the Seg experiment is 2260. This numbers shows the amount of candidates for the stop-word lists that could be hand-filtered. Nevertheless, the numbers are similar to the size of stop-word list that are used in other systems like in [11]. In Section 6 we show that the Top 10 and well-tuned stop-word lists allow to achieve the similar improvement. Thus, the proposed method could be used to automatically produce lists that conform to stop lists.

We already noticed that we did not applied any stop-word list in the preprocessing of the corpus and the regular stop words appear in the feature vectors. The feature vectors were created for each descriptor of the development corpus, and the vector consisted of words, bigrams or collocation segments and their frequencies, as shown in Table 3. We set six feature vectors for each descriptor for the different experiments:

- full list of words **(B)**;
- the same as **B** and the top 10 frequent words excluded for each vector **(T)**;
- full list of bigrams **(Bi)**;
- the same as **Bi** and the top 10 frequent bigrams excluded for each vector **(BiT)**;
- full list of collocation segments **(S)**;
- the same as **S** and the top 10 frequent segments excluded for each vector **(ST)**.

Bigram feature in our experiments is referred as two consequent words within a text. For instance, the bigram features of the sentence *the form of interest*

Table 2. Descriptor assignment Precision results

	1 descriptor						2 descriptors						5 descriptors						10 descriptors					
	B	T	S	ST	Bi	BiT	B	T	S	ST	Bi	BiT	B	T	S	ST	Bi	BiT	B	T	S	ST	Bi	BiT
bg	25.5	26.8	33.5	**39.2**	26.9	26.2	22.3	23.9	29.1	**34.6**	22.9	22.3	16.5	17.2	20.4	**25.3**	15.6	15.1	11.8	12.5	14.3	**17.8**	10.4	10.1
cs	34.6	39.5	41.7	**46.1**	28.4	27.7	29.4	34.5	35.2	**40.4**	23.9	22.7	21.7	26.2	26.4	**31.0**	16.7	16.3	14.8	18.1	17.8	**21.2**	12.2	11.9
da	29.5	28.7	39.8	**42.0**	26.7	26.3	25.6	25.1	35.5	**37.8**	22.8	22.0	17.6	17.8	25.4	**27.6**	16.0	16.2	12.1	12.5	17.3	**18.9**	10.6	11.1
de	24.9	31.9	40.6	**46.0**	23.1	23.9	20.8	26.7	20.9	**39.8**	20.4	21.4	14.9	19.9	23.2	**28.6**	14.1	15.0	10.7	14.2	17.0	**19.7**	9.8	10.7
el	26.5	28.6	43.6	**44.2**	25.9	26.2	24.0	24.3	37.9	**39.7**	23.0	21.5	17.7	18.1	27.5	**30.4**	16.6	16.9	12.2	12.7	18.3	**20.8**	11.4	12.1
en	35.6	37.8	**45.5**	43.5	30.2	29.7	29.8	33.5	**40.2**	39.6	25.9	25.8	20.6	24.3	28.5	**29.1**	18.5	18.5	14.4	17.2	19.0	**19.6**	12.3	12.6
es	32.2	26.6	44.7	**45.2**	29.8	28.6	26.1	22.9	38.5	**40.4**	24.0	24.5	19.6	17.0	28.9	**30.4**	17.4	17.7	13.7	11.9	19.3	**20.8**	11.9	12.4
et	23.5	40.6	36.7	**43.2**	22.2	24.1	20.7	35.2	31.4	**39.2**	18.5	19.6	14.6	25.5	22.2	**28.6**	12.3	13.3	10.2	17.8	15.6	**19.7**	8.2	9.4
fi	27.1	35.1	34.1	**41.2**	23.8	23.0	23.3	30.5	30.3	**36.3**	20.8	19.0	16.3	22.3	21.0	**26.5**	14.6	13.4	11.0	16.0	14.3	**18.9**	9.8	9.3
fr	29.2	27.6	39.3	**41.8**	23.3	27.7	23.9	22.8	33.7	**39.3**	21.6	23.8	17.4	16.5	24.1	**29.4**	14.6	17.8	11.8	11.5	16.7	**20.3**	10.0	12.6
hu	30.2	41.8	39.8	**42.2**	23.4	24.0	25.4	37.6	35.0	**37.7**	22.0	21.7	17.6	27.5	24.7	**27.4**	16.2	15.2	12.0	18.8	16.8	**19.0**	10.9	10.4
it	31.5	32.8	43.7	**47.7**	28.5	28.8	27.3	30.0	38.6	**43.3**	24.2	23.4	18.8	21.9	27.7	**31.7**	17.9	16.9	12.9	15.2	18.8	**21.9**	11.9	12.1
lt	32.6	41.8	37.7	**45.3**	25.7	22.8	27.1	37.0	32.8	**40.5**	22.2	20.8	19.8	28.5	23.8	**30.4**	16.2	14.7	13.5	19.9	16.1	**21.1**	11.1	10.2
lv	24.7	38.5	36.4	**42.5**	20.9	25.0	21.4	33.6	31.8	**37.6**	18.0	19.9	15.8	24.7	22.5	**27.4**	12.4	14.0	11.0	17.4	15.3	**18.8**	8.4	9.8
mt	17.5	21.9	21.1	**35.5**	17.4	25.3	15.3	20.2	17.9	**32.8**	14.6	22.8	10.9	14.6	12.2	**25.7**	10.7	16.3	7.8	10.3	8.4	**18.1**	7.4	11.2
nl	30.1	33.7	**42.7**	41.3	26.8	26.0	24.5	28.1	36.7	**37.8**	22.8	21.8	17.7	20.4	26.9	**28.6**	17.2	16.9	12.7	14.6	18.3	**19.9**	11.7	12.0
pl	34.5	40.2	40.9	**45.8**	27.4	24.4	29.0	35.7	40.0	**40.8**	23.7	21.7	20.5	26.0	24.8	**29.5**	17.2	15.4	14.1	18.4	16.6	**20.2**	11.4	10.8
pt	32.0	33.9	41.8	**46.5**	22.5	24.5	27.5	29.7	37.4	**41.6**	20.2	22.7	19.7	21.4	26.8	**30.4**	13.3	15.1	13.6	15.1	17.8	**20.8**	8.5	11.0
sk	35.1	43.3	42.4	**47.2**	29.1	23.1	30.7	37.9	37.4	**42.6**	25.6	21.0	21.5	27.7	26.3	**31.2**	17.8	15.0	14.6	19.4	18.0	**21.3**	12.1	10.4
sl	31.8	34.5	39.6	**45.7**	24.6	25.3	27.4	31.1	33.7	**40.1**	23.3	22.4	20.1	23.6	25.2	**30.0**	15.9	16.1	13.9	16.6	17.0	**20.4**	11.4	11.1
sv	31.9	28.1	40.3	**42.3**	28.1	25.7	26.6	24.8	34.8	**39.1**	25.5	23.9	19.3	18.1	26.4	**30.5**	18.2	16.7	13.6	13.1	17.5	**21.0**	12.6	12.1
Avg	29.5	34.0	39.3	**43.5**	25.5	25.6	25.1	29.8	33.8	**39.1**	22.2	22.1	18.0	21.9	24.5	**29.0**	15.7	15.8	12.5	15.4	16.7	**20.0**	10.7	11.1

Table 3. Descriptor assignment Recall results

	1 descriptor						2 descriptors						5 descriptors						10 descriptors					
	B	T	S	ST	Bi	BiT	B	T	S	ST	Bi	BiT	B	T	S	ST	Bi	BiT	B	T	S	ST	Bi	BiT
bg	9.0	9.5	11.9	**13.9**	9.5	9.3	7.9	8.5	10.3	**12.2**	8.1	7.9	14.6	15.2	18.1	**22.4**	13.8	13.4	20.9	22.1	25.3	**31.5**	18.4	17.9
cs	12.2	14.0	14.8	**16.3**	10.1	9.8	10.4	12.2	12.5	**14.3**	8.5	8.0	19.2	23.2	23.4	**27.4**	14.8	14.4	26.2	32.0	31.5	**37.5**	21.6	21.1
da	10.4	10.2	14.1	**14.9**	9.5	9.3	9.1	8.9	12.6	**13.4**	8.1	7.8	15.6	15.8	22.5	**24.4**	14.2	14.3	21.4	22.1	30.6	**33.5**	18.8	19.6
de	8.8	11.3	14.4	**16.3**	8.2	8.5	7.4	9.5	7.4	**14.1**	7.2	7.6	13.2	17.6	20.5	**25.3**	12.5	13.3	18.9	25.1	30.1	**34.9**	17.3	18.9
el	9.4	10.1	15.4	**15.6**	9.2	9.3	8.5	8.6	13.4	**14.1**	8.1	7.6	15.7	16.0	24.3	**26.9**	14.7	15.0	21.6	22.5	32.4	**36.8**	20.2	21.4
en	12.6	13.4	**16.1**	15.4	10.7	10.5	10.5	11.9	**14.2**	14.0	9.2	9.1	18.2	21.5	25.2	**25.8**	16.4	16.4	25.5	30.4	33.6	**34.7**	21.8	22.3
es	11.4	9.4	15.8	**16.0**	10.5	10.1	9.2	8.1	13.6	**14.3**	8.5	8.7	17.3	15.0	25.6	**26.9**	15.4	15.7	24.2	21.1	34.2	**36.8**	21.1	21.9
et	8.3	14.4	13.0	**15.3**	7.9	8.5	7.3	12.5	11.1	**13.9**	6.5	6.9	12.9	22.6	19.6	**25.3**	10.9	11.8	18.1	31.5	27.6	**34.9**	14.5	16.6
fi	9.6	12.4	12.1	**14.6**	8.4	8.1	8.2	10.8	10.7	**12.8**	7.4	6.7	14.4	19.7	18.6	**23.5**	12.9	11.9	19.5	28.3	25.3	**33.5**	17.3	16.5
fr	10.3	9.8	13.9	**14.8**	8.2	9.8	8.5	8.1	11.9	**13.9**	7.6	8.4	15.4	14.6	21.3	**26.0**	12.9	15.8	20.9	20.4	29.6	**35.9**	17.7	22.3
hu	10.7	14.8	14.1	**14.9**	8.3	8.5	9.0	13.3	12.4	**13.3**	7.8	7.7	15.6	24.3	21.9	**24.2**	14.3	13.5	21.2	33.3	29.7	**33.6**	19.3	18.4
it	11.2	11.6	15.5	**16.9**	10.1	10.2	9.7	10.6	13.7	**15.3**	8.6	8.3	16.6	19.4	24.5	**28.1**	15.8	15.0	22.8	26.9	33.3	**38.8**	21.1	21.4
lt	11.5	14.8	13.3	**16.0**	9.1	8.1	9.6	13.1	11.6	**14.3**	7.9	7.4	17.5	25.2	21.1	**26.9**	14.3	13.0	23.9	35.2	28.5	**37.3**	19.6	18.1
lv	8.7	13.6	12.9	**15.0**	7.4	8.8	7.6	11.9	11.3	**13.3**	6.4	7.0	14.0	21.9	19.9	**24.2**	11.0	12.4	19.5	30.8	27.1	**33.3**	14.9	17.3
mt	6.2	7.8	7.5	**12.6**	6.2	9.0	5.4	7.2	6.3	**11.6**	5.2	8.1	9.6	12.9	10.8	**22.7**	9.5	14.4	13.8	18.2	14.9	**32.0**	13.1	19.8
nl	10.7	11.9	**15.1**	14.6	9.5	9.2	8.7	9.9	13.0	**13.4**	8.1	7.7	15.7	18.1	23.8	**25.3**	15.2	15.0	22.5	25.8	32.4	**35.2**	20.7	21.2
pl	12.2	14.2	14.5	**16.2**	9.7	8.6	10.3	12.6	14.2	**14.4**	8.4	7.7	18.1	23.0	21.9	**26.1**	15.2	13.6	25.0	32.6	29.4	**35.8**	20.2	19.1
pt	11.3	12.0	14.8	**16.5**	8.0	8.7	9.7	10.5	13.2	**14.7**	7.2	8.0	17.4	18.9	23.7	**26.9**	11.8	13.4	24.1	26.7	31.5	**36.8**	15.0	19.5
sk	12.4	15.3	15.0	**16.7**	10.3	8.2	10.9	13.4	13.2	**15.1**	9.1	7.4	19.0	24.5	23.3	**27.6**	15.8	13.3	25.8	34.3	31.9	**37.7**	21.4	18.4
sl	11.3	12.2	14.0	**16.2**	8.7	9.0	9.7	11.0	11.9	**14.2**	8.2	7.9	17.8	20.9	22.3	**26.5**	14.1	14.2	24.6	29.4	30.1	**36.1**	20.2	19.6
sv	11.3	9.9	14.3	**15.0**	9.9	9.1	9.4	8.8	12.3	**13.8**	9.0	8.5	17.1	16.0	23.4	**27.0**	16.1	14.8	24.1	23.2	31.0	**37.2**	22.3	21.4
Avg	10.5	12.0	13.9	**15.4**	9.0	9.1	8.9	10.5	11.9	**13.8**	7.9	7.8	15.9	19.3	21.7	**25.7**	13.9	14.0	22.1	27.2	29.5	**35.4**	18.9	19.7

Table 4. Descriptor assignment F1-measure results

	1 descriptor						2 descriptors						5 descriptors						10 descriptors					
	B	T	S	ST	Bi	BiT	B	T	S	ST	Bi	BiT	B	T	S	ST	Bi	BiT	B	T	S	ST	Bi	BiT
bg	13.3	14.0	17.5	**20.5**	14.1	13.7	11.7	12.5	15.2	**18.1**	12.0	11.7	15.5	16.2	19.2	**23.8**	14.6	14.2	15.1	16.0	18.3	**22.7**	13.3	12.9
cs	18.1	20.7	21.8	**24.1**	14.8	14.5	15.4	18.0	18.4	**21.1**	12.5	11.9	20.4	24.6	24.8	**29.1**	15.7	15.3	18.9	23.1	22.7	**27.1**	15.6	15.2
da	15.4	15.0	20.8	**22.0**	14.0	13.8	13.4	13.1	18.6	**19.8**	11.9	11.5	16.5	16.7	23.8	**25.9**	15.0	15.2	15.5	16.0	22.1	**24.2**	13.5	14.2
de	13.0	16.7	21.2	**24.1**	12.1	12.5	10.9	14.0	10.9	**20.8**	10.7	11.2	14.0	18.7	21.8	**26.9**	13.2	14.1	13.7	18.1	21.7	**25.2**	12.5	13.7
el	13.9	15.0	22.8	**23.1**	13.5	13.7	12.5	12.7	19.8	**20.8**	12.0	11.2	16.6	17.0	25.8	**28.5**	15.6	15.9	15.6	16.2	23.4	**26.6**	14.6	15.5
en	18.6	19.8	**23.8**	22.7	15.8	15.5	15.6	17.5	**21.0**	20.7	13.5	13.5	19.3	22.8	26.8	**27.3**	17.4	17.4	18.4	22.0	24.3	**25.0**	15.7	16.1
es	16.8	13.9	23.4	**23.6**	15.6	15.0	13.6	12.0	20.1	**21.1**	12.5	12.8	18.4	16.0	27.1	**28.5**	16.3	16.6	17.5	15.2	24.7	**26.6**	15.2	15.8
et	12.3	21.2	19.2	**22.6**	11.6	12.6	10.8	18.4	16.4	**20.5**	9.7	10.2	13.7	23.9	20.8	**26.9**	11.5	12.5	13.0	22.7	19.9	**25.2**	10.5	12.0
fi	14.2	18.4	17.8	**21.5**	12.4	12.0	12.2	15.9	15.8	**19.0**	10.9	9.9	15.3	20.9	19.7	**24.9**	13.7	12.6	14.1	20.4	18.3	**24.2**	12.5	11.9
fr	15.3	14.4	20.5	**21.9**	12.2	14.5	12.5	11.9	17.6	**20.5**	11.3	12.4	16.3	15.5	22.6	**27.6**	13.7	16.7	15.1	14.7	21.3	**25.9**	12.8	16.1
hu	15.8	21.9	20.8	**22.1**	12.2	12.5	13.3	19.7	18.3	**19.7**	11.5	11.3	16.5	25.8	23.2	**25.7**	15.2	14.3	15.3	24.0	21.5	**24.3**	13.9	13.3
it	16.5	17.2	22.8	**24.9**	14.9	15.1	14.3	15.7	20.2	**22.6**	12.7	12.2	17.7	20.6	26.0	**29.8**	16.8	15.9	16.5	19.4	24.0	**28.0**	15.2	15.5
lt	17.0	21.9	19.7	**23.7**	13.4	11.9	14.2	19.3	17.2	**21.2**	11.6	10.9	18.6	26.8	22.3	**28.5**	15.2	13.8	17.3	25.4	20.6	**27.0**	14.2	13.0
lv	12.9	20.1	19.0	**22.2**	10.9	13.1	11.2	17.6	16.6	**19.7**	9.4	10.4	14.8	23.2	21.1	**25.7**	11.6	13.1	14.1	22.2	19.6	**24.0**	10.7	12.5
mt	9.2	11.5	11.0	**18.6**	9.1	13.2	8.0	10.6	9.4	**17.2**	7.6	11.9	10.2	13.7	11.5	**24.1**	10.0	15.3	10.0	13.2	10.7	**23.1**	9.5	14.3
nl	15.7	17.6	**22.3**	21.6	14.0	13.6	12.8	14.7	19.2	**19.8**	11.9	11.4	16.6	19.2	25.3	**26.9**	16.2	15.9	16.2	18.7	23.4	**25.4**	15.0	15.3
pl	18.0	21.0	21.4	**23.9**	14.3	12.8	15.2	18.7	20.9	**21.3**	12.4	11.3	19.2	24.4	23.3	**27.7**	16.2	14.5	18.0	23.5	21.2	**25.8**	14.6	13.8
pt	16.7	17.7	21.9	**24.3**	11.8	12.8	14.4	15.5	19.6	**21.8**	10.6	11.9	18.5	20.1	25.2	**28.5**	12.5	14.2	17.4	19.3	22.7	**26.6**	10.9	14.1
sk	18.4	22.6	22.2	**24.7**	15.2	12.1	16.1	19.8	19.6	**22.3**	13.4	11.0	20.2	26.0	24.7	**29.3**	16.7	14.1	18.7	24.8	23.0	**27.2**	15.5	13.3
sl	16.6	18.0	20.7	**23.9**	12.9	13.2	14.3	16.3	17.6	**21.0**	12.2	11.7	18.9	22.2	23.7	**28.2**	14.9	15.1	17.8	21.2	21.7	**26.1**	14.6	14.2
sv	16.7	14.7	21.1	**22.1**	14.7	13.4	13.9	13.0	18.2	**20.4**	13.3	12.5	18.1	17.0	24.8	**28.6**	17.1	15.7	17.4	16.7	22.4	**26.8**	16.1	15.5
Avg	15.4	17.8	20.6	**22.8**	13.3	13.4	13.2	15.6	17.6	**20.4**	11.6	11.6	16.9	20.5	23.0	**27.3**	14.7	14.9	16.0	19.7	21.3	**25.6**	13.6	14.2

Table 5. The Precision for the 5 descriptors assignment for English for the documents of the different feature vector size

Feature vector size of test documents			B	T	Bi	BiT	S	ST
20	...	99	4.1	10.4	5.8	3.3	23.0	**36.1**
100	...	249	36.5	**44.4**	42.6	43.7	33.0	30.0
250	...	499	19.8	24.3	18.9	18.2	26.3	**27.5**
500	...	999	18.4	22.6	17.2	18.0	26.7	**27.5**
1000	...	2499	19.9	22.5	16.1	16.0	24.1	**24.6**
2500	...	4999	20.1	21.6	13.1	13.7	24.3	**25.7**
5000	...	9999	21.7	22.0	14.2	13.2	30.2	**33.5**
10000	...	24999	29.2	29.4	17.8	16.5	34.8	**33.6**
25000	...		33.6	30.0	12.6	11.6	33.3	**40.0**

payments are *the form*, *form of*, *of interest*, *interest payments*. Once feature vectors (see Table 1) exist for all descriptors, the descriptors can be assigned to new texts by calculating the similarity between the feature vector of a text and the feature vector of a descriptor. We used raw frequencies of features and Cosine score [12] for calculating the similarity between the descriptor and the document. To this end, we created six feature vectors for each test document in the same way as for descriptor. After preparation of the feature vectors the calculations of the Cosine similarity between each descriptor from development corpus and each document in the test corpus were performed. At the end, the number of the best descriptors for each test document were selected. The example comparison of manually assigned descriptors and the automatically assigned descriptors is shown in Figure 4.

6 Results

The results in Tables 2, 3 and 4 show that, for all languages, using the combination of the collocation segmentation and the exclusion of the top 10 items does indeed produce the best results. The average increase of the precision is from 18 to 29 percent and of F-measure is from 16.9 to 27.3 points for 5 descriptor assignment. However, the segmentation does not always improve better than the exclusion of the top 10. The collocation segmentation significantly improves the descriptor assignment performance for the less inflected languages. The segmentation improved 5 descriptor assignment precision for English by 38%, and the top 10 improved precision by 18%. Conversely, the segmentation improved precision for Finnish by 29%, and the top 10 improved precision by 37%. The result shows the high importance of collocation segmentation for the less inflected language. Also, the results shows that for many languages the assignment performance does not suffer much if the collocation segmentation and the exclusion of the top 10 is carried out.

The segmentation improves the descriptor assignment precision better than using bigrams. Our results show that on the average we achieve minimal degradation when bigrams are used. This result was a small surprise for us. To understand the problem, we looked at the dependency between the precision and

the size of the feature vector of a test document (Table 5). The bigrams allow to achieve best results compare to Base and Seg for documents that contains from 100 to 250 features. For instance, bigrams allow to improve precision from 40 to 49.6 percent for English in [3]. An improvement was achieved in [16] and [20] also. The improvements were achieved by using multinomial naïve Bayes or SVM classifiers. The use of multinomial classifiers makes difficult to judge on the direct influence value of bigrams itself. For instance, [9:251-258] shows that the best classification performance is achieved when the number of features selected are from 100 to 300. Our results show that the collocation segmentation allow to achieve similar performance for the feature vectors of the different lengths (see Table 5). The results in Table 5 show the dependency between the precision and the feature vector length of a document. The conclusion is that bigram features make the improvement for the documents with relatively small feature vectors and Reuters-21578 database is of this kind. The collocation segmentation allows to achieve very good classification results for a very small documents.

The precision for 5 keyword assignment is much higher than for 10 keyword assignment. However, F-measure is similar for 5 and 10 keyword assignment. Thus, the system is capable of capturing more correct descriptors while the number of assigned descriptors is slightly increased. Therefore, for the descriptor assignment it is useful to assign from 5 to 10 descriptors. Our results show that the exclusion of the top 10 items can improve descriptor assignment performance by 3-6%. A manually set stop word list improves classification results at least by 3-5% [11]. This result opens the possibility to use the proposed top 10 method instead of manual stop-word lists for many languages.

The collocation segmentation increases the size of a dictionary from 4 to 6 times. This increase reduces the frequencies of the dictionary entries. Our results show that the increase of a dictionary size and decrease of frequencies do not reduce the keyword assignment performance. This result indicates the importance of the dictionary quality. The reduction of dictionary size is used often in order to reduce the complexity of the system while classification quality remains similar or increases [5]. Thus, the classification performance can be increased by the collocation segmentation and selection of the best features.

7 Conclusion

In the current work, we have presented the influence of the collocation segmentation and the top 10 frequent items to the descriptor assignment to the text. The assignment performance was assessed on the multilingual AC corpus. Two outstanding conclusions can be stated from the results presented. First, the collocation segmentation increases a dictionary size considerably. The increase of the dictionary size is smaller for highly inflected languages and bigger for non-inflected languages. Thus, the dictionaries of segments for different languages become more comparable than dictionaries of words. The segmentation allows to reduce the differences among languages. Our study shows that the deviation of precision, recall and F-measure is lower language by language when the segmentation is performed. Second, the combination of the segmentation and the

exclusion of the top 10 frequent items does indeed produce the best results. The similar performance for the different languages (including Finnish and English) shows that the collocation segmentation and the top 10 items is language independent and that the methods can be applied to further languages.

Acknowledgement. We would like to express our appreciations to Gregory Grefenstette and Ralf Steinberger for the comments and suggestions for this paper.

References

1. Choueka, Y.: Looking for needles in a haystack, or locating interesting collocational expressions in large textual databases. In: Proceedings of the RIAO Conference on User-Oriented Content-Based Text and Image Handling, Cambridge, MA, March 21-24 (1988)
2. Civera, J., Juan, A.: Bilingual Machine-Aided Indexing. In: Proceedings of the 5th international conference on Language Resources and Evaluation, LREC 2006, Genoa, Italy, May 24-26, pp. 1302–1305 (2006)
3. Civera, J.: Novel statistical approaches to text classification, machine translation and computer-assisted translation. PhD thesis, Universidad Politécnica de Valencia (2008)
4. Daudaravicius, V., Marcinkeviciene, R.: Gravity Counts for the Boundaries of Collocations. International Journal of Corpus Linguistics 9(2), 321–348 (2004)
5. Dhillon, I., Kogan, J., Nicholas, C.: Feature Selection and Document Clustering. Survey of Text Mining: Clustering, Classification, and Retrieval (2004)
6. Fox, C.: A stop list for general text. ACM-SIGIR Forum 24, 19–35 (1990)
7. Hao, L., Hao, L.: Automatic Identification of Stop Words in Chinese Text Classification. In: Proceedings of the 2008 international Conference on Computer Science and Software Engineering, CSSE, December 12 - 14, vol. 01, pp. 718–722. IEEE Computer Society, Washington (2008)
8. Lin, D.: Extracting collocations from text corpora. In: First Workshop on Computational Terminology, Montreal (1998)
9. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, Cambridge (2008)
10. Névéol, A., Mork, J.G., Aronson, A.R., Darmoni, S.J.: Evaluation of French and English mesh indexing systems with a parallel corpus. In: Proceedings of the AMIA Symposium (2005)
11. Pouliquen, B., Steinberger, R., Ignat, C.: Automatic Annotation of Multilingual Text Collections with a Conceptual Thesaurus. In: Proceedings of the Workshop Ontologies and Information Extraction at the Summer School, The Semantic Web and Language Technology - Its Potential and Practicalities, Bucharest, Romania (2003)
12. Salton, G.: Automatic Text Processing: the Transformation, Analysis and Retrieval of Information by Computer. Addison-Wesley, Reading (1989)
13. Smadja, F.: Retrieving Collocations from Text: XTRACT. Computational Linguistics 19(1), 143–177 (1993)
14. Smadja, F., McKeown, K.R., Hatzivassiloglou, V.: Translation Collocations for Bilingual Lexicons: A Statistical Approach. Computational Linguistics 22(1), 1–38 (1996)

15. Steinberger, R., Pouliquen, B., Widiger, A., Ignat, C., Erjavec, T., Tufiş, 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), Genoa, Italy, May 24-26, pp. 2142–2147 (2006)
16. Tesar, R., Strnad, V., Jezek, K., Poesio, M.: Extending the single words-based document model: a comparison of bigrams and 2-itemsets. In: Proceedings of the 2006 ACM Symposium on Document Engineering, Amsterdam, The Netherlands, pp. 138–146 (2006)
17. Thesaurus Eurovoc (2009), http://europa.eu/eurovoc/
18. Tjong Kim Sang, E.F., Buchholz, S.: Introduction to the CoNLL-2000 Shared Task: Chunking. In: Proceedings of CoNLL 2000, Lisbon, Portugal, pp. 127–132 (2000)
19. Weiss, S., Indurkhya, N., Zhang, T., Damerau, F.: Text Mining: Predictive Methods for Analyzing Unstructured Information. Springer, Heidelberg (2005)
20. Van Der Zwaan, J., Tjong Kim Sang, E.F., De Rijke, M.: An Experiment in Automatic Classification of Pathological Reports. In: Bellazzi, R., Abu-Hanna, A., Hunter, J. (eds.) AIME 2007. LNCS (LNAI), vol. 4594, pp. 207–216. Springer, Heidelberg (2007)

A General Bio-inspired Method to Improve the Short-Text Clustering Task

Diego Ingaramo[1], Marcelo Errecalde[1], and Paolo Rosso[2]

[1] LIDIC, Universidad Nacional de San Luis, Argentina
[2] Natural Language Eng. Lab. ELiRF, DSIC,
Universidad Politécnica de Valencia, Spain
{daingara,merreca}@unsl.edu.ar, prosso@dsic.upv.es

Abstract. "Short-text clustering" is a very important research field due to the current tendency for people to use very short documents, e.g. blogs, text-messaging and others. In some recent works, new clustering algorithms have been proposed to deal with this difficult problem and novel bio-inspired methods have reported the best results in this area. In this work, a general bio-inspired method based on the AntTree approach is proposed for this task. It takes as input the results obtained by arbitrary clustering algorithms and refines them in different stages. The proposal shows an interesting improvement in the results obtained with different algorithms on several short-text collections.

1 Introduction

Nowadays, the huge amount of information available in the Web offers an unlimited number of opportunities to use this information in different real life problems. Unfortunately, the automatic analysis tools that are required to make this information useful for the human comprehension, such as clustering, categorization and information extraction systems, have to face many difficulties related to the features of the documents to be processed. For example, most of Web documents like blogs, snippets, chats, FAQs, on-line evaluations of commercial products, e-mails, news, scientific abstracts and others are "*short texts*". This is a central aspect if we consider the well-known problems that short documents usually pose to different natural language processing tasks [1,2].

During the last years, different works have recognized the importance (and complexity) of dealing with short documents, and some interesting results have been reported in *short-document clustering* tasks. These studies include the correlation between internal and external validity measures [3], the estimation of the hardness of short-text corpora [1,2] and the use of bio-inspired methods [4,5].

Recently, the *AntSA-CLU* algorithm [6] reported the best results in experiments with different short-text collections of small size. AntSA-CLU is a hierarchical AntTree-based algorithm which incorporates two main concepts: the *Silhouette Coefficient* [7] and the idea of *attraction* of a cluster. A key component of AntSA-CLU is the initial data partition generated by the CLUDIPSO algorithm [4,5] which is used to generate new and better groupings.

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 661–672, 2010.

Despite the good performance showed by AntSA-CLU in that work, some important aspects of this approach deserve a deeper analysis. These aspects can be summarized by the following questions:

1. can these ideas used in *AntSA-CLU* be successfully applied in other arbitrary algorithms? or, in other words, can they be used in a general improvement method for arbitrary clustering algorithms?
2. is the *AntSA-CLU*'s effectiveness limited to small size collections or it can be an useful algorithm for arbitrary size short-text collections?

In the present work, we will address these questions by using a simplified and more general version of AntSA-CLU, named *Partitional AntSA⋆ (PAntSA⋆)* and considering in the experiments a more representative set of short-text collections. PAntSA⋆ is the *partitional* version of the *herarchical* AntSA-CLU method where, furthermore, it is not assumed as input the results of any particular clustering algorithm. In that way, PAntSA⋆ will take the clusterings generated by arbitrary clustering algorithms and attempt to improve them by using techniques based on the Silhouette Coefficient and the idea of attraction.

The remainder of the paper is organized as follows. Section 2 describes the main ideas of PAntSA⋆, the method proposed as a general improvement technique for short-text clustering algorithms. The experimental setup and the analysis of the results obtained from our empirical study is provided in Section 3. Finally, some general conclusions are drawn and possible future work is discussed.

2 The PAntSA⋆ Algorithm

The *Partitional AntSA⋆ (PAntSA⋆)* algorithm is a bio-inspired method intended to improve the results obtained with arbitrary document clustering algorithms. Document clustering is the unsupervised assignment of documents to unknown categories. This task is more difficult than supervised document categorization because the information about categories and correctly categorized documents is not provided in advance. PAntSA⋆ is the partitional version of the *AntSA* (**Ant**Tree-**S**ilhouette-**A**ttraction) algorithm. *AntSA* is based on the AntTree algorithm [8] but it also incorporates information related to the Silhouette Coefficient and the concept of *attraction* of a cluster in different stages of the clustering process.

In AntSA, each ant represents a single datum from the data set and it moves in the structure according to its similarity to the other ants already connected to the tree under construction. Each node in the tree structure represents a single ant and each ant represents a single datum. Each ant to be connected to the tree represents a data to be classified. Starting from an artificial support called a_0, all the ants will be incrementally connected either to that support or to other already connected ants. This process continues until all ants are connected to the structure, i.e., all data are already clustered.

The whole collection of ants is initially represented by a (possibly sorted) list $\mathcal{L}$ of ants waiting to be connected in further steps. During the tree generation

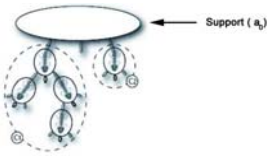


Fig. 1. A tree interpreted as a non hierarchical data partition (adapted from [8])

process each selected ant a_i will be either connected to the support (or another ant) or moving on the tree looking for an adequate place to connect itself. The simulation process continues until all ants have found the more adequate place; either on the support or on another ant.

The resulting tree (see Figure 1) can be interpreted as a data partition (considering each ant connected to a_0 as a different group) as well as a dendrogram where the ants in the inner nodes could move to the leaves following the most similar nodes to them.

An important aspect is the initial arrangement of the ants in the $\mathcal{L}$ list. Since the algorithm iteratively proceeds taking the ants from $\mathcal{L}$, this list determines the order in which ants will be considered to be connected in the support structure (each one representing a different group). For this reason, the features of the first ants in $\mathcal{L}$ will significantly influence the final result.

AntSA differs from AntTree in two main steps: a) the *initial ordering step* that establishes how the ants will be initially ordered in $\mathcal{L}$; b) the *comparison of an arbitrary ant with the ants connected to the support*; this process determines the primary cluster assignments of ants, depending on the selected path. AntSA basically attempts to improve the performance of AntTree by:

1. considering in the *initial step* of AntTree, additional information about the Silhouette Coefficient of previous clusterings;
2. using a more informative criterium (based on the concept of *attraction*) when the ants have to decide which path to follow when they are on the support.

2.1 Using Silhouette Coefficient Information of Previous Clusterings

The initial ordering step defines the order in which ants will be connected to the support (each one representing a different group). Therefore, any little modification in this ordering will significatively impact the clustering results. Our proposal consists in taking as input the clustering obtained with some arbitrary clustering algorithm and using the Silhouette Coefficient (SC) information of this grouping to determine the initial order of ants.

The *Global Silhouette* (GS) coefficient is an *Internal Clustering Validity Measure* (ICVM) which has shown to be a very effective cluster validation tool. However, some recent works have proposed other uses of this measure, specially in the context of short-text clustering problems. In [2] for example, the

evaluation of the GS coefficient and other ICVMs on the "gold standard" of different short-text collections is proposed as a method to estimate the hardness of those corpora. GS has also been used as an explicit *objective function* that the clustering algorithms attempt to optimize. This idea has recently been used in short-texts clustering tasks, using discrete and continuous Particle Swarm Optimization (PSO) algorithms as function optimizers [4,5]. In these works, a discrete PSO algorithm named CLUDIPSO obtained the best results on different short-text corpora when the GS measure was used as objective function.

The GS measure combines two key aspects to determine the quality of a given clustering: *cohesion* and *separation*. Cohesion measures how closely related are objects in a cluster whereas separation quantifies how distinct (well-separated) a cluster from other clusters is. The GS coefficient of a clustering is the average cluster silhouette of all the obtained groups. The cluster silhouette of a cluster C also is an average silhouette coefficient but, in this case, of all objects belonging to C. Therefore, the fundamental component of this measure is the formula used to determine the SC value of any arbitrary object i, that we will refer as $s(i)$ and that is defined as $s(i) = \frac{b(i)-a(i)}{max(a(i),b(i))}$ with $-1 \leq s(i) \leq 1$. The $a(i)$ value denotes the average dissimilarity of the object i to the remaining objects in its own cluster, and $b(i)$ is the average dissimilarity of the object i to all objects in the nearest cluster. From this formula it can be observed that negative values for this measure are undesirable and that we want for this coefficient values as close to 1 as possible.

The SC-based ordering of ants carried out in this stage determines which will be the first ants connected to the support structure. The ants with the highest SC value within each group will be considered more desirable because they are the most representative ants of their groups.

2.2 Using an Attraction-Based Comparison

Another key aspect for an arbitrary ant a_i on the support is the decision about which connected ant a_+ should move toward. In fact, this decision will determine the group in which a_i will be incorporated. AntTree takes into account for this decision, the similarity between a_i and its most similar ant connected to the support (a^+). This is a "local" approach that only considers the ant directly connected to the support structure (a^+) but it does not take into account the ants previously connected to a^+, that will be denoted as $\mathcal{A}_{a^+}$. In the AntSA algorithm a more global approach that also considers some information on $\mathcal{A}_{a^+}$ is used. If $\mathcal{G}_{a^+} = \{a^+\} \cup \mathcal{A}_{a^+}$ is the group formed by a^+ and its descendants, this relationship between the group $\mathcal{G}_{a^+}$ and the ant a_i will be referred as the *attraction of* $\mathcal{G}_{a^+}$ *on* a_i and will be denoted as $att(a_i, \mathcal{G}_{a^+})$.

The idea of having different groups exerting some kind of "attraction" on the objects to be clustered was already posed in [9], where it was used as an efficient tool to obtain "dense" groups. In the present work, we will give a more general sense to the concept of attraction by considering that $att(a_i, \mathcal{G}_{a^+})$ represents *any plausible estimation of the quality of the group that would result if* a_i *were*

incorporated to $\mathcal{G}_{a^+}$ ($\mathcal{G}_{a^+} \cup \{a_i\}$). Thus, the only modification that AntSA introduce to AntTree in this case is the use of $att(a_i, \mathcal{G}_{a^+})$ instead of $Sim(a_i, a^+)$ to determine the ant a^+ with the highest $att(a_i, \mathcal{G}_{a^+})$ value. Then, a_i will be moved toward a^+ and will continue looking for a suitable place to connect itself, either to a^+ or to some a^+'s descendant.

To compute $att(a_i, \mathcal{G}_{a^+})$ we can use any ICVM that allows to estimate the quality of individual clusters, and to apply this ICVM to $\mathcal{G}_{a^+} \cup \{a_i\}$. For instance, any *cohesion*-based ICVM could be used in this case, but other more elaborated approaches (like the density-based ones) would also be valid alternatives. As an example, an effective attraction measure is the average similarity between a_i and all the ants in $\mathcal{G}_{a^+}$ as shown in Equation 1.

$$att(a_i, \mathcal{G}_{a^+}) = \frac{\sum_{a \in \mathcal{G}_{a^+}} Sim(a_i, a)}{|\mathcal{G}_{a^+}|} \tag{1}$$

2.3 PAntSA*, a Partitional Simplified Version of AntSA

When a hierarchical organization of the results is not required, some parameters and initialization steps required by AntSA are not necessary. Removing these aspects, which are specific to the tree generation, results in a partitional version of AntSA, named PAntSA*, which is simpler and more efficient than the original AntSA algorithm. PAntSA* is also based on the use of the GS Coefficient and the idea of attraction-based comparison. However, PAntSA* does not build herarchical structures which have roots (ants) directly connected to the support. In PAntSA*, each ant a_j connected to the support (a_0) and its descendants (the $\mathcal{G}_{a_j}$ group) is considered as a simple set. In that way, when an arbitrary ant a_i has to be incorporated to the group of the ant a^+ that more attraction exerts on a_i, this step is implemented by simply adding a_i to the $\mathcal{G}_{a^+}$ set.

The resulting PAntSA* algorithm is given in Figure 2, where it is possible to observe that it takes an arbitrary clustering as input and carries out the following three steps, in order to obtain the new clustering:

1. Connection to the support.
2. Generation of the $\mathcal{L}$ list.
3. Cluster the ants in $\mathcal{L}$.

In the first step, the most representative ant of each group of the clustering received as input is connected to the support a_0. This task involves to select the ant a_i with the highest SC value of each group C_i, and to connect each one of them to the support by generating a singleton set $\mathcal{G}_{a_i}$.

The second step consists in generating the $\mathcal{L}$ list with the ants not connected in the previous step. This process also considers the SC-based ordering obtained in the previous step, and merges the remaining (ordered) ants of each group by iteratively taking the first ant of each non-empty queue, until all queues are empty.

In the third step, the order in which these ants will be processed is determined by their positions in the $\mathcal{L}$ list. The clustering process of each arbitrary ant a_i

function PAntSA$^\star(\mathcal{C})$ **returns** a clustering $\mathcal{C}^\star$
input: $\mathcal{C} = \{C_1, \ldots, C_k\}$, an initial grouping
1. Connection to the support
1.a. Create a set $\mathcal{Q} = \{q_1, \ldots, q_k\}$ of k data queues (one queue for each group $C_j \in \mathcal{C}$).
1.b. Sort each queue $q_j \in \mathcal{Q}$ in decreasing order according to the Silhouette Coefficient of its elements. Let $\mathcal{Q}' = \{q'_1, \ldots, q'_k\}$ be the resulting set of ordered queues.
1.c. Let $\mathcal{G}_\mathcal{F} = \{a_1, \ldots, a_k\}$ be the set formed by the first ant a_i of each queue $q'_i \in \mathcal{Q}'$. For each ant $a_i \in \mathcal{G}_\mathcal{F}$, remove a_i from q'_i and set $\mathcal{G}_{a_i} = \{a_i\}$ (connect a_i to the support a_0).
2. Generation of the $\mathcal{L}$ list
2.a. Let $\mathcal{Q}'' = \{q''_1, \ldots, q''_k\}$ the set of queues resulting from the previous process of removing the first ant of each queue in $\mathcal{Q}'$.
Generate the $\mathcal{L}$ list by merging the queues in $\mathcal{Q}''$.
3. Clustering process
3.a. Repeat
3.a.1 Select the first ant a_i from the list $\mathcal{L}$.
3.a.2 Let a^+ the ant with the highest $att(a_i, \mathcal{G}_{a^+})$ value.
$\mathcal{G}_{a^+} \leftarrow \mathcal{G}_{a^+} \cup \{a_i\}$
Until $\mathcal{L}$ is empty
return $\mathcal{C}^\star = \{\mathcal{G}_{a_1}, \ldots, \mathcal{G}_{a_k}\}$

Fig. 2. The PAntSA* algorithm

simply determines the connected ant a^+ which exerts more attraction on a_i (according to Equation 1) and then includes a_i in the a^+' group ($\mathcal{G}_{a^+}$). The algorithm finally returns a clustering formed by the groups of the ants connected to the support.

3 Experimental Setting and Analysis of Results

For the experimental work, seven collections with different levels of complexity with respect to the size, length of documents and vocabulary overlapping were selected: CICling-2002, EasyAbstracts, Micro4News, SEPLN-CICLing, R4, R8+ and R8-.

CICling-2002 is a well-known short-text collection that has been recognized in different works [10,11,3,2,4,5] as a very difficult collection since its documents are narrow domain scientific abstracts (short-length documents with a high vocabulary overlapping). Micro4News is a low complexity collection of medium-length documents about well-differentiated topics (wide domain). The EasyAbstracts corpus is composed of short-length documents (scientific abstracts) on well differentiated topics (medium complexity corpus). Finally, SEPLN-CICLing is a corpus that it is supposed to be harder to cluster than the previous corpora since its documents are narrow domain abstracts. SEPLN-CICLing and CICling-2002 have similar characteristics. However, all the SEPLN-CICLing's abstracts guarantee a

minimum quality level with respect to their lengths, an aspect that is not assured by all the CICling-2002's documents.

The four previous corpora are small size collections that allow to carry out a very detailed analysis which would be difficult with standard large size collections. Unfortunately, if only these collections were considered in our study it would not be possible to determine if the conclusions also apply to larger collections. For this reason, other three larger collections were considered in the experiments: R4, R8+ and R8-. These collections are subsets of the well known R8-Test corpus, a subcollection of the Reuters-21578 dataset. The R4 collection has the same number of groups that the previous collections (4 groups) but it is considerably larger. The R8+ and R8- collections have 8 groups like the original R8-Test but they differ in the length of their documents. The length of the R8+'s documents is, on average, ten times the length of the R8-'s documents [1].

The documents were represented with the standard (normalized) tf-idf codification after a *stop-word* removing process. The popular *cosine measure* was used to estimate the similarity between two documents. The parameter settings for CLUDIPSO and the remainder algorithms used in the comparison with PAntSA* correspond to the parameters empirically derived in [5]. The attraction measure ($att(\cdot)$) used in our study corresponds to the formula presented in Equation 1.

3.1 Experimental Results

The results of PAntSA* were compared with the results of other four clustering algorithms: K-means, K-MajorClust [9], CHAMELEON [12] and CLUDIPSO [4,5]. K-means is one of the most popular clustering algorithms whereas K-MajorClust and CHAMELEON are representative of the density-based approach to the clustering problem and have shown interesting results in similar problems [2]. CLUDIPSO is a bio-inspired algorithm, which attempts to optimize the GS coefficient of the clusterings by using a discrete PSO approach. This algorithm has obtained in previous works [4,5] the best results in experiments with the four small size short-text collections presented in Section 3 (CICling-2002, EasyAbstracts, Micro4News and SEPLN-CICLing).

The quality of the results was evaluated by using the classical (external) F-measure on the clusterings that each algorithm generated in 50 independent runs per collection. The reported results correspond to the minimum (F_{min}), maximum (F_{max}) and average (F_{avg}) F-measure values. The values highlighted in bold in the different tables indicate the best obtained results.

Tables 1 and 2 show the F_{min}, F_{max} and F_{avg} values that K-means, K-MajorClust, CHAMELEON and CLUDIPSO obtained with the seven collections. These tables also include the results obtained with PAntSA* taking as input the groupings generated by these algorithms. They will be denoted with a "$\star$" superscript. Thus, for example, the results obtained with PAntSA* taking as input the groupings generated by K-Means, will be denoted as K-Means*.

[1] A more detailed description of these corpora is given in [10,2,4].

[2] The K-MajorClust algorithm is based on the MajorClust algorithm proposed in [9], but it was modified to generate exactly K groups.

Table 1. Best F-measures values per collection

	Micro4News			EasyAbstracts			SEPLN-CICLing			CICling-2002		
Algorithms	F_{avg}	F_{min}	F_{max}	F_{avg}	F_{min}	F_{max}	F_{avg}	F_{min}	F_{max}	F_{avg}	F_{min}	F_{max}
K-Means	0.67	0.41	0.96	0.54	0.31	0.71	0.49	0.36	0.69	0.45	0.35	0.6
K-Means*	0.84	0.67	**1**	0.76	0.46	0.96	0.63	0.44	0.83	0.54	0.41	0.7
K-MajorClust	0.95	0.94	0.96	0.71	0.48	**0.98**	0.63	0.52	0.75	0.39	0.36	0.48
K-MajorClust*	**0.97**	**0.96**	**1**	0.82	0.71	**0.98**	0.68	0.61	0.83	0.48	0.41	0.57
CHAMELEON	0.76	0.46	0.96	0.74	0.39	0.96	0.64	0.4	0.76	0.46	0.38	0.52
CHAMELEON*	0.85	0.71	0.96	0.91	0.62	**0.98**	0.69	0.53	0.77	0.51	0.42	0.62
CLUDIPSO	0.93	0.85	**1**	0.92	0.85	**0.98**	0.72	0.58	**0.85**	0.6	**0.47**	0.73
CLUDIPSO*	0.96	0.88	**1**	**0.96**	**0.92**	**0.98**	**0.75**	**0.63**	**0.85**	**0.61**	**0.47**	**0.75**

Table 2. Best F-measures values per collection

	R4			R8-			R8+		
Algorithms	F_{avg}	F_{min}	F_{max}	F_{avg}	F_{min}	F_{max}	F_{avg}	F_{min}	F_{max}
K-Means	0.73	0.57	0.91	0.64	0.55	0.72	0.60	0.46	0.72
K-Means*	**0.77**	0.58	**0.95**	0.67	0.52	0.78	0.65	0.56	**0.73**
K-MajorClust	0.70	0.45	0.79	0.61	0.49	0.7	0.57	0.45	0.69
K-MajorClust*	0.70	0.46	0.84	0.61	0.5	0.71	0.63	0.55	0.72
CHAMELEON	0.61	0.47	0.83	0.57	0.41	0.75	0.48	0.4	0.6
CHAMELEON*	0.69	**0.6**	0.87	0.67	**0.6**	0.77	0.61	0.55	0.67
CLUDIPSO	0.64	0.48	0.75	0.62	0.49	0.72	0.57	0.45	0.65
CLUDIPSO*	0.71	0.53	0.85	**0.69**	0.54	**0.79**	**0.66**	**0.57**	0.72

These results confirm the good performance that CLUDIPSO has already shown in previous works with the four small size collections. It achieves the highest F_{max} values in Micro4News, EasyAbstracts and SEPLN-CICLing and the highest F_{min} value for CICling-2002. However, these results obtained by CLUDIPSO are clearly improved by PAntSA* which not only obtains the same highest F_{max} values as CLUDIPSO, it also obtains the highest F_{min} and F_{avg} values for EasyAbstracts, SEPLN-CICLing and CICling-2002 and the best F_{max} value reported in this work for CICling-2002. With respect to Micro4News, it is interesting to observe that the best F_{min} and F_{avg} values for this collection are also obtained by PAntSA* but, in this case, with the groupings generated by K-MajorClust.

PAntSA* also exhibited good improvement capabilities with the larger short-text collections as can be appreciated in Table 2. Here, PAntSA* obtained the highest F values for R4 by improving the groupings obtained by K-Means and CHAMELEON, the highest F values for R8- by improving the groupings obtained by CHAMELEON and CLUDIPSO and the the best F values for R8+ by improving the groupings obtained by CLUDIPSO and K-Means.

Up to now, our analysis has been focused on the best obtained values for each collection. However, it is also interesting to make a comparison between the results that the different algorithms obtain on the seven considered collections

Table 3. Results of PAntSA* vs. groupings generated by different algorithms

	Micro4News			EasyAbstracts			SEPLN-CICLing			CICling-2002		
Algorithms	F_{avg}	F_{min}	F_{max}	F_{avg}	F_{min}	F_{max}	F_{avg}	F_{min}	F_{max}	F_{avg}	F_{min}	F_{max}
K-Means	0.67	0.41	0.96	0.54	0.31	0.71	0.49	0.36	0.69	0.45	0.35	0.6
K-Means*	**0.84**	**0.67**	**1**	**0.76**	**0.46**	**0.96**	**0.63**	**0.44**	**0.83**	**0.54**	**0.41**	**0.7**
K-MajorClust	0.95	0.94	0.96	0.71	0.48	0.98	0.63	0.52	0.75	0.39	0.36	0.48
K-MajorClust*	**0.97**	**0.96**	**1**	**0.82**	**0.71**	**0.98**	**0.68**	**0.61**	**0.83**	**0.48**	**0.41**	**0.57**
CHAMELEON	0.76	0.46	**0.96**	0.74	0.39	0.96	0.64	0.4	0.76	0.46	0.38	0.52
CHAMELEON*	**0.85**	**0.71**	**0.96**	**0.91**	**0.62**	**0.98**	**0.69**	**0.53**	**0.77**	**0.51**	**0.42**	**0.62**
CLUDIPSO	0.93	0.85	**1**	0.92	0.85	**0.98**	0.72	0.58	**0.85**	0.6	**0.47**	0.73
CLUDIPSO*	**0.96**	**0.88**	**1**	**0.96**	**0.92**	**0.98**	**0.75**	**0.63**	**0.85**	**0.61**	**0.47**	**0.75**

Table 4. Results of PAntSA* vs. groupings generated by different algorithms

	R4			R8-			R8+		
Algorithms	F_{avg}	F_{min}	F_{max}	F_{avg}	F_{min}	F_{max}	F_{avg}	F_{min}	F_{max}
K-Means	0.73	0.57	0.91	0.64	**0.55**	0.72	0.60	0.46	0.72
K-Means*	**0.77**	**0.58**	**0.95**	**0.67**	0.52	**0.78**	**0.65**	**0.56**	**0.73**
K-MajorClust	**0.70**	0.45	0.79	**0.61**	0.49	0.7	0.57	0.45	0.69
K-MajorClust*	**0.70**	**0.46**	**0.84**	**0.61**	**0.5**	**0.71**	**0.63**	**0.55**	**0.72**
CHAMELEON	0.61	0.47	0.83	0.57	0.41	0.75	0.48	0.4	0.6
CHAMELEON*	**0.69**	**0.6**	**0.87**	**0.67**	**0.6**	**0.77**	**0.61**	**0.55**	**0.67**
CLUDIPSO	0.64	0.48	0.75	0.62	0.49	0.72	0.57	0.45	0.65
CLUDIPSO*	**0.71**	**0.53**	**0.85**	**0.69**	**0.54**	**0.79**	**0.66**	**0.57**	**0.72**

and the results that PAntSA* obtains taking as input the clustering generated by these algorithms. Tables 3 and 4 facilitate this comparison by presenting the results of each algorithm with the corresponding results obtained by PAntSA* in these cases. The highlighted best values show that PAntSA* seems to obtain a considerable improvement level on all the collections and algorithms considered in the experiments. As un example, when PAntSA* takes as input the clusterings generated by K-MajorClust, these results (identified as K-MajorClust*) are consistently better than (or as good as) those obtained by the K-MajorClust algorithm, on the seven considered collections. These improvements obtained with PAntSA* can also be observed for the remaining algorithms.

Despite the excellent results shown by PAntSA* in the previous comparisons, it is important to observe that, at least in the case of the F_{min} value obtained with K-Means* in R8-, it is possible to observe a deterioration with respect to the K-Means' result. This result suggests that, despite the *average* improvements that PAntSA* achieves on all the considered algorithms, and the highest F_{max} values obtained on the seven collections, a deeper analysis is required that also considers the *improvements* (or the *deteriorations*) that PAntSA* carries out on each particular clustering that it receives as input. The graphics shown

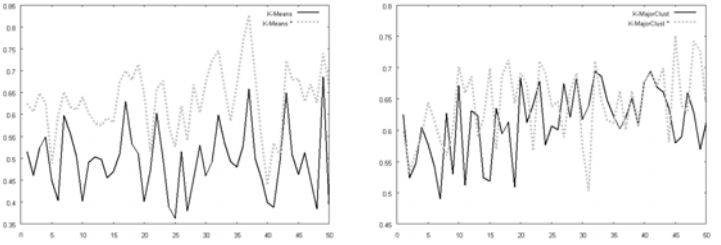


Fig. 3. PAntSA*: significant (left) and minor (right) improvement level

in Figure 3 can help to understand this last aspect. Here, the F-measure values obtained in 50 experiments with two different algorithms (black line) are shown, together with the F-measure values that PAntSA* obtained with these clusterings as input (dashed gray line). The graphic on the left, clearly shows that PAntSA* improves *all* the results of this algorithm. However, in the graphic on the right, a lower effectiveness level of PAntSA* can be observed.

These observations pose some questions about *how often* (and in *what extent*) we can expect to observe an improvement in the quality of the clusterings provided to PAntSA*. Tables 5 and 6 give some insights on this subject, by presenting in Table 5 the *improvement percentage* (IP) and the *improvement magnitude* (IM) obtained with PAntSA*, whereas Table 6 gives the *deterioration percentage* (DP) and the *deterioration magnitude* (DM) that PAntSA* produced on the original clusterings. The *percentage* of cases where PAntSA* produces clusterings with the *same quality* as the clusterings received as input (SQP) can be directly estimated from the two previous percentages. Thus, for example, PAntSA* produced an improvement in the 94% of the cases when received the clusterings generated by K-Means on the Micro4News collection, giving F-measures values which are (on average) a 0.18 higher than the F-measures values obtained with K-Means. In this case, $DP = 4\%$ and $DM = 0.05$ meaning that in 2% of the experiments with this algorithm and this collection, PAntSA* gave results of the same quality ($SQP = 2\%$).

With the exception of the K-Means - R4 combination, where PAntSA* does not obtain significant improvements, the remaining experimental instances are conclusive about the advantages of using PAntSA* as a general improvement method. Thus, for example, in 4 experimental instances (algorithm-collection

Table 5. IP and MP values

	4MNG		Easy		SEPLN-CIC		CIC-2002		R4		R8-		R8+	
Algorithms	IP	MP	IP	MP	IP	MP	IP	MP	IP	MP	IP	MP	IP	MP
K-Means	94	0.18	94	0.24	100	0.14	96	0.09	56	0.1	70	0.07	97	0.07
K-MajorClust	50	0.03	94	0.13	94	0.04	100	0.09	96	0.05	61	0.06	97	0.07
CHAMELEON	87	0.11	100	0.17	100	0.07	75	0.08	91	0.09	85	0.1	100	0.13
CLUDIPSO	74	0.05	86	0.05	84	0.03	92	0.03	76	0.12	84	0.09	89	0.06

Table 6. DP and DM values

	4MNG		Easy		SEPLN-CIC		CIC-2002		R4		R8-		R8+	
Algorithms	DP	DM	DP	DM	DP	DM	DP	DM	DP	DM	DP	DM	DP	DM
K-Means	4	0.05	6	0.04	0	0	4	0.03	44	0.07	30	0.04	2	0.003
K-MajorClust	0	0	0	0	6	0.01	0	0	4	0.03	38	0.04	2	0.001
CHAMELEON	0	0	0	0	0	0	25	0.06	8	0.02	14	0.03	0	0
CLUDIPSO	6	0.04	14	0.02	16	0.02	8	0.009	24	0.05	16	0.04	10	0.03

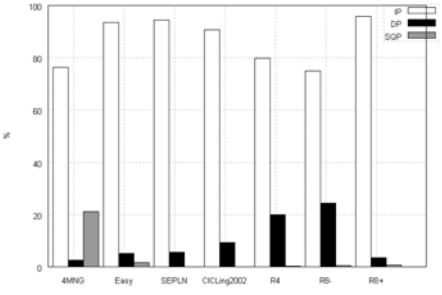


Fig. 4. IP, DP and SQP values per collection

combinations) PAntSA* obtained an improvement in the 100% of the experiments. This excellent performance of PAntSA* can be easily appreciated in Figure 4, where the IP (white bar)), DP (black bar) and SQP (gray bar) values are compared but considering in this case the improvements/deteriorations obtained in each one of the seven collections.

4 Conclusions and Future Work

In this work we presented PAntSA*, a general bio-inspired method to improve the short-text clustering task. PAntSA* achieved the best F_{min}, F_{max} and F_{avg} values on all the considered collections. These results were obtained by improving the clusterings obtained with different clustering algorithms.

PAntSA* does not guarantee an improvement of all the clusterings received as input. However, a decrease in the F-measure values of the results produced by PAntSA* is not a very frequent result. This claim is supported by the following experimental data: on the total of experiments ($1400 = 7$ collections $\times$ 4 algorithms $\times$ 50 runs per algorithm) PAntSA* obtained 1211 improvements, 48 results with the same quality and only 141 lower quality results.

A direct extension to this work, is to provide to PAntSA* with a clustering generated by the own PAntSA* algorithm. This idea gives origin to an iterative version of PAntSA* which has already given some interesting results, even in those cases where PAntSA* is provided with random initial clusterings.

Acknowledgments. We thank the TEXT-ENTERPRISE 2.0 TIN2009-13391-C04-03 research project for funding the work of the second and third authors.

References

1. Pinto, D., Rosso, P.: On the relative hardness of clustering corpora. In: Matoušek, V., Mautner, P. (eds.) TSD 2007. LNCS (LNAI), vol. 4629, pp. 155–161. Springer, Heidelberg (2007)
2. Errecalde, M., Ingaramo, D., Rosso, P.: Proximity estimation and hardness of short-text corpora. In: Proceedings of TIR 2008, pp. 15–19. IEEE CS, Los Alamitos (2008)
3. Ingaramo, D., Pinto, D., Rosso, P., Errecalde, M.: Evaluation of internal validity measures in short-text corpora. In: Gelbukh, A. (ed.) CICLing 2008. LNCS, vol. 4919, pp. 555–567. Springer, Heidelberg (2008)
4. Cagnina, L., Errecalde, M., Ingaramo, D., Rosso, P.: A discrete particle swarm optimizer for clustering short-text corpora. In: BIOMA 2008, pp. 93–103 (2008)
5. Ingaramo, D., Errecalde, M., Cagnina, L., Rosso, P.: Particle Swarm Optimization for clustering short-text corpora. In: Computational Intelligence and Bioengineering, pp. 3–19. IOS Press, Amsterdam (2009)
6. Ingaramo, D., Errecalde, M., Rosso, P.: A new anttree-based algorithm for clustering short-text corpora. JCS&T (2009) (to be published)
7. Rousseeuw, P.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J. Comput. Appl. Math. 20, 53–65 (1987)
8. Azzag, H., Monmarche, N., Slimane, M., Venturini, G., Guinot, C.: AntTree: A new model for clustering with artificial ants. In: Proc. of the CEC 2003, Canberra, pp. 2642–2647. IEEE Press, Los Alamitos (2003)
9. Stein, B., Meyer zu Eißen, S.: Document Categorization with MajorClust. In: Proc. WITS 2002, Technical University of Barcelona, pp. 91–96 (2002)
10. Alexandrov, M., Gelbukh, A., Rosso, P.: An approach to clustering abstracts. In: Montoyo, A., Muńoz, R., Métais, E. (eds.) NLDB 2005. LNCS, vol. 3513, pp. 8–13. Springer, Heidelberg (2005)
11. 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)
12. Karypis, G., Han, E.H., Vipin, K.: Chameleon: Hierarchical clustering using dynamic modeling. Computer 32, 68–75 (1999)

An Empirical Study on the Feature's Type Effect on the Automatic Classification of Arabic Documents

Saeed Raheel and Joseph Dichy

Université Lumière Lyon 2, 86 rue Pasteur, 69635 LYON Cedex 07, France
saeed.raheel@gmail.com, joseph.dichy@univ-lyon2.fr

Abstract. The Arabic language is a highly flexional and morphologically very rich language. It presents serious challenges to the automatic classification of documents, one of which is determining what type of attribute to use in order to get the optimal classification results. Some people use roots or lemmas which, they say, are able to handle problems with the inflections that do not appear in other languages in that fashion. Others prefer to use character-level n-grams since n-grams are simpler to implement, language independent, and produce satisfactory results. So which of these two approaches is better, if any? This paper tries to answer this question by offering a comparative study between four feature types: words in their original form, lemmas, roots, and character level n-grams and shows how each affects the performance of the classifier. We used and compared the performance of Support Vector Machines and Naïve Bayesian Networks algorithms respectively.

Keywords: Arabic document classification, text mining, natural language processing.

1 Introduction

Recently, researchers started to give more importance to the automatic processing of multilingual data (in specific, online data since the biggest information resource nowadays is the Internet). This is due to many reasons: the ever increasing number of online multilingual resources, the further development of the infrastructure of communication and internet, and the always increasing number of internet users whose mother-tongue is not always the English language. Such reasons urged the researches to dig out and find new automatic methods or tailor already existing ones in order to process and organize the continuously increasing immense volume of online data. The manual systems, such as for example, the systems built by experts or knowledge-engineered systems are very expensive in terms of time and human resources and do not offer enough flexibility in terms of generalization and portability to different domains [7]. This is why machine learning algorithms such as Support Vector Machines and Naïve Bayesian Networks emerged. One of the tasks involved in this process is that of the automatic classification of documents which our work extends and applies to the specific and intricate case of *Arabic.*

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 673–686, 2010.

For any statistical document classification task, the morphology of Arabic is a crucial challenge. The problem we have to deal with when using full-form words is the sparsity of data. This problem can be reduced by either applying a morphological analysis or using n-grams.

As far as morphological analysis is concerned, the conversion or reduction of a full-form word into a lemma or root is based on linguistic knowledge and a complex set of rules. This approach is pretty much used by practitioners for the automatic classification of Arabic documents, such is in [23], [26], and [27], and often leads to very good classification accuracy having an F1-measure equal to 0.878 as reported by [27]. By definition, a lemma is the basic dictionary-form for words sharing the same meaning e.g. "الكتابه" (*el kitâba - the writing)*, "كتابه" (*kitâba - writing)*, "الكتابات" (*el kitâbât - the writings)*, and "كتاباتكم" (*kitâbâtoukom - your writings*), etc. have all the same lemma "كتابه" and the root "كتب". The difference between a lemma and a root is that a root is the part of the word that never changes even when morphologically inflected, whilst a lemma is the base form of the verb. In Arabic, things get a lot more complicated and messier as shown in section 3 and, hence, a deep analysis is required. In order to overcome such a challenge, this paper relies on the solutions available through the computerized dictionary DIINAR.1 [16], [17], [18], [19], [20] and its associated modules.

The other alternative approach used in various areas of statistical natural language processing to cope with the sparsity of full-form words, such as in [22], [24], and [25], is the use of character n-grams that lead to very good and highly competitive results such as in [24] who report obtaining a classification accuracy with an F1 measure equal to 0.881 . By definition, the n-grams of a given word (or subsequently, a group of words) are the subsequence of n-characters that form the word. For example, the 3-grams generated from "natural language processing" are: "nat", "atu", "tur", "ral", "al ", " la", "lan", etc.

A general rule of thumb in the domain of document classification is that no two results obtained by any two authors can be directly compared nor conclusions can be directly built or drawn upon. In order to be able to do so both experiments should have been done in strictly the same conditions of input so that the output can be comparable. Therefore, in this paper we used the same set of documents out of which we built five datasets: one for each feature type. We then used the same feature extraction measures to reduce their sizes, the same term weighting mechanism, and same data mining software. As a result, we compared the accuracy of each of the classifiers and tried to find which input leads to the best classification accuracy.

This paper is organized as follows. Section 2, describes the task of automatic document classification. Section 3 describes in particular the difficulties encountered while processing Arabic documents. Support Vector Machines and Naïve Bayesian Networks algorithms are presented in Section 4 and 5, respectively. Section 6 describes the dataset and its conception and preparation phase as well as the preprocessing it undergoes for the purpose of classification. Sections 7 and 8 describe the feature selection and evaluation metrics used in this paper. Section 9 illustrates the different experiments carried out in this paper, and finally Section 10 summarizes the whole work.

2 Automatic Document Classification

Simply put, the automatic classification of documents (henceforth ACD) is the task of assigning a document to a predefined category or set of categories. The fact that the categories have been predefined renders the ACD supervised. In this work, we assign one category to each document. A formal definition of ACD would be the task of building (or learning) an approximation function $\hat{\Phi}$ of the function $\Phi : D \times C \rightarrow \{1,0\}$, which assigns a given document d_i to a category c_j, where *D* is the set of documents and *C* is the set of predefined categories. A value of 1 of the function $\hat{\Phi}$ for the pair (d_i , c_j) means that the document d_i belongs to the category c_j and 0 means otherwise. The built function $\hat{\Phi} : D \times C \rightarrow \{1,0\}$ is called the *classifier.* In order for the classifier to be subjective and generalizable, two conditions must be fulfilled:

- Categories are only nominal labels i.e. their names do not contribute to or affect the decision of the classifier whatsoever.
- Categorization is based solely on the contents of the documents and not on their metadata e.g. name, author, keywords, etc.

In what follows we present a brief description of the main real obstacles encountered while working with Arabic documents and not faced while working documents written in languages using Latin characters and that are overcome by the automatic segmentation of each word using the analyzer and segmenter of the computerized dictionary DIINAR.1. The resulting sub-segments of the words are then used as features in the dataset.

3 The Difficulties Encountered with Arabic

According to the IPR Strategic Business Information Database , Arabic is the mother tongue of 338.4 million people (Article date: April 22, 2009). It is one of the six official languages of the United Nations and contrary to all Latin-based alphabets, its orientation of writing is from right to left. The Arabic alphabet consists of 28 letters and can be extended to ninety by additional shapes, marks and vowels. Each letter can appear in up to four different shapes, depending on whether it occurs at the beginning, in the middle, at the end of a word, or alone. Arabic is a Semitic language whose grammatical system is based on a root-and pattern structure and considered as a root-based. Arabic contains three genders (much like English): masculine, feminine and neuter. It differs from a lot of other languages in that it contains three grammatical numbers instead of the common two numbers (singular and plural). The third one is the dual which refers to precisely two entities. Therefore, when mining documents written in Arabic characters we are faced with a number of problems proper to the language itself. These problems are enumerated in details in [10] and [15]. We briefly state them in what follows:

3.1 Unvowelled Writing

The tradition of 'unvowelled' writing – in usual texts – is, in Arabic, a major impediment for efficient automatic pre-treatment [10]. Automated 'vowelling' is far from being an easy task [17]. Standard script does not include diacritic signs for short vowels[1], consonant doubling, case-endings, etc which often carry crucial morpho-syntactic information. As a result, some words are likely to generate, at word-level, as many as 30 analyses or more [15].

3.2 The Complex Structure of the Arabic Word-Form

According to [18] a substantial subset of Arabic word-forms consist of a stem of the <root + pattern> type, to which a finite set of compatible proclitics and prefixes are agglutinated to the left and a finite set of compatible suffixes and enclitics are agglutinated to the right. This renders the isolation of the stem very difficult.

Many word-forms refer to much more than a noun, verb or adjective. Some can even be equated to a whole sentence in English, e.g.: أستفعلونه*'asataf'alûnahu*, English: 'will you do it?' or the stem قالَ[*qâla*], 'he said', when combined with the clitic interrogative conjunction أ*'a#* (**'#'** is for 'clitic border') produces the written word-form أقال*'a#qâla*, 'did he say...?' In the analysis process, this form is also that of another stem, أقال*'aqâla*, 'he fired'. In this example, both realisations are similar, not only in writing, but also in pronunciation.

Pure surface analysis of words can therefore be expected to be inoperative. The human reading process resorts to complex analyses of contextual signs and indications, while the input of morphological analysis is context-free word-forms.

A surface search for « *قال* » is matched by no less than 146 different forms in our 2 million words corpus. This obviously goes far beyond the number of associated derivations or readings. The request yields such words as: اعتقالهم –الانتقال ... قالب – الأقاليم – وقالباً – اثقالاً –برتقالية – التقاليد – أقالتهم – مقاليد – العقال – استقالة – These words comprise the sequence of letters *qâf-alif-lâm,* but refer to other verbs than قال*qâla* (root /q-w-l/) and may relate to other roots, e.g. قالب*qâlab,* 'mould' (root /q-l-b/). In addition to noises, many forms related to the same lemma cannot be found by mere surface searching, e.g., the imperfective يقول*yaqûlu* of the verb *qâla;* or deverbal forms such as the active participle قائل, *qâ'il*, etc. Considerable silence thus adds to noises.

Users of such requests also encounter problems with partially or entirely vowelled words: قالَ –قالَ –قال –قا are treated as different forms. Answering this type of request means that the analyzer considers, in such words, vowelled and un-vowelled letters alike, which cannot be done if the system only recognises generic characters. In other terms, one needs to erase vowels by hand or automatically before launching the analyzer. In order to meet this issue, we have used for the segmentation of words according to the method described above a morphological analyzer drawing information from the DIINAR.1 language database [10], [16], [18], [19], [20], [21].

[1] Short vowels can be represented by diacritics placed above or below the letter e.g. /a/ as in *had*, /i/ as in *fit*, and /u/ as in *foot*.

4 Support Vector Machines (SVM)

SVMs are a set of supervised binary classifiers proposed to solve two-class problems by finding the optimal separating hyperplane or *margin* between two classes of data as shown in Figure 2. By this we mean that viewing input data as two sets of vectors in an *n*-dimensional space, an SVM will construct a separating hyperplane in that space, one which maximizes the *margin* between the two data sets. To calculate the margin, two parallel hyperplanes are constructed, one on each side of the separating hyperplane, which are "pushed up against" the two data sets.

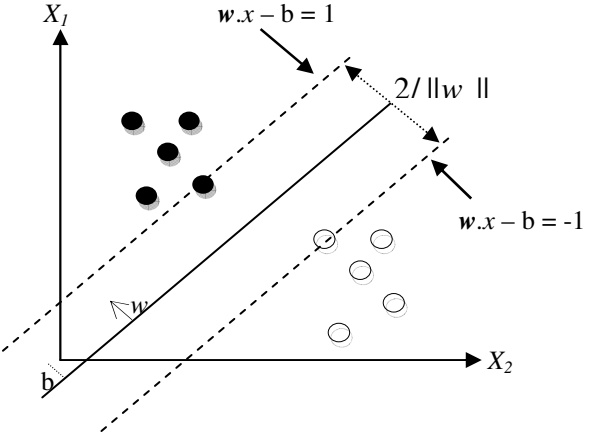


Fig. 1. The optimal separating hyperplane between two classes of data

Let $D = \{(x_i, y_i)\} : 1 \le i \le N\}$ denote the set of instances in the dataset such that $x \in \mathbb{R}^m$ be the set of documents in the dataset, and $y \in \{1, -1\}$ be the set of output classes. In case the data is linearly separable then $\exists$ a vector $\boldsymbol{w} \in \mathbb{R}^n$ and a scalar $b \in \mathbb{R}$ such that:

$$\boldsymbol{w}^T x - b = 0 \tag{1}$$

If we are to maximize the margin and the training data is linearly separable then we have to minimize over ($\boldsymbol{w}$, b) the value of $1/2\|w\|^2$ subject to $y_i(\boldsymbol{w}x_i - b) \ge 1$ where $1 \le i \le N$. In case the data is inseparable then one way to solve this problem is to generalize SVM to the minimization of $1/2\|w\|^2 + C\sum_{i=1}^{n}\xi_i$, where C is a constant to trade off between margin and training error, subject to $y_i(\boldsymbol{w}x_i - b) \ge 1 - \xi_i$ where $1 \le i \le N$.

5 Naïve Bayesian Networks

Naïve Bayes classification algorithms are a probability-driven linear algorithms based on Bayes theorem and on the mere assumption that the terms used in documents are independent. The general Bayes theorem for classification purpose is:

$$\Pr(\text{Class}|\text{Document}) = \frac{\Pr(\text{Class}).\Pr(\text{Document}|\text{Class})}{\Pr(\text{Document})} \quad (2)$$

Since Pr(Document) is a constant divider it is disregarded. Moreover, this equation is further simplified by only considering the words in the document while discarding everything else such as delimiters e.g. whitespaces, punctuation marks, etc., giving:

$$\Pr(\text{Class}|\text{Document}) = \Pr(\text{Class}).\prod_i \Pr(\text{Word}i|\text{Class}) \quad (3)$$

In what follows, we describe the dataset preparation process.

6 The Dataset

There are no Arabic datasets available on the internet for public use. Therefore, we had to build our own from Arabic online websites. We gathered a collection of 7,034 articles of varying lengths partitioned among 7 categories as follows: Politics (1020), Economy (958), Sports (867), Medicine (1190), Science and Technology (945), Law (889), and Religion (1165).

In order to build our dataset we had, in some cases, to manually save local copies of web pages. However, in most cases, we used an in-house application that automatically processes the RSS feeds published by those websites.

The preparation of the Arabic dataset went through the following automatic processes:

- Extract from the RSS feed the url pointing to the webpage containing the complete article and open that page automatically.
- Save the webpage's html content as a separate plain text file within its corresponding category's folder.
- Extract from the text file the body of the article only. The additional parts of the webpage (menus, links to other similar articles, ads, etc.) were ignored automatically.
- Apply the following pre-processing steps:
    - Removal of all diacritics.
    - Removal of all word delimiters i.e. punctuation marks and any non-Arabic characters e.g. numbers, Latin characters, special characters, etc.
    - Removal of all stop words. The list of those words was taken from our computerized dictionary DIINAR.1 [16].
    - Removal of any word that appears less than 3 times in the document.
- Extract/generate the words/lemmas/roots/n-grams for the article and save each in a separate text file.

After all text files are processed and ready, the dataset file is created following to the global vector space model. For the calculation of the terms' weights we used TF-IDF (*Term Frequency-Inverse Document Frequency)* defined as follows:

Let t_i be a random term of the document d_j and tf_{ij} be the number of times the term t_i appears in d_j i.e. the frequency of t_i in d_j. Let df_i be the number of documents containing the term t_i i.e. $df_i = |\{d_j : t_i \in d_j\}|$ and $|D|$ the total number of documents in the dataset. We define the inverse document frequency as:

$$\mathrm{IDF_i} = \log(\frac{|D|}{df_i}) \tag{4}$$

and thus,

$$TFIDF_{ij} = tf_{ij} . IDF_i \tag{5}$$

7 Feature Selection Metrics

The result of the vector space model is a vector space with high dimensionality. Working with such a huge vector space is a cumbersome task for the machine learning algorithm affecting both of its performance and reliability. Therefore, we resorted to what is called a *term space reduction (TSR)* process, and mainly, we applied and compared the results of 2 very well-known feature selection techniques: *Information Gain (IG)*, *Chi Square (χ^2) statistic* defined briefly in what follows:

We define the information gain of an attribute *A* relative to a dataset *D,* denoted by *IG(D, A),* as:

$$IG(D, A) = Entropy(D) - Entropy(A) \tag{6}$$

where,

$$Entropy(D) = \sum_{i=1}^{c} - ex_i \log_2 ex_i \tag{7}$$

where ex_i is the proportion of examples in *D* belonging to class *i* and, the entropy of an attribute *A* with values $\{a_1, \ldots, a_v\}$ as:

$$Entropy(A) = \sum_{i=1}^{v} \frac{|D_v|}{|D|} . Entropy(S_v) \tag{8}$$

where D_v is the subset in *D* for which attribute *A* has value *v* i.e. $D_v = \{ex \in \mathrm{D} \mid A(ex) = v\}$.

The Chi Square (χ^2) statistic measures the lack of independence between a feature and a category. The chi square of term t_i in category c_j is calculated as follows:

$$\chi^2(t_i, c_j) = \frac{|D|.(AE - CB)^2}{(A+C)(B+E)(A+B)(C+E)} \quad (9)$$

where, |D| is the total number of documents in the dataset, A is the number of documents in c_j containing t_i, B is the number of documents containing t_i but not belonging to c_j, C is the number of documents in c_j not containing t_i, and, E is the number of documents neither belonging to c_j nor containing t_i.

8 Evaluation Metrics

Based on Table 2, we evaluated the performance of our system, using the three very well known evaluation measures: *precision, recall,* and the *F-measure* (sometimes referred to as F-1 measure) as defined in Table 3.

Table 1. Contingency table

Category c_i		**Correct Classification**	
		$d \in c_i$	$d \notin c_i$
Classifier's estimate	Document assigned to c_i	TP_i	FP_i
	Document rejected from c_i	FN_i	TN_i

In order to evaluate the performance average across categories we use the conventional method named *macro-averaging.* Macro-averaged performance scores are calculated by first calculating the precision, recall, and F-measure of each category and then dividing each by the total number of categories.

Table 2. Evaluation Measures

Precision	**Recall**	**F-Measure**
$\frac{TP}{TP + FP}$	$\frac{TP}{TP + FN}$	$\frac{2 \times Precision \times Recall}{Precsion + Recall}$

In what follows, we present our experiments and the results obtained.

9 Experimental Results

As we've mentioned before, the dataset used consists of 7,034 documents of different sizes partitioned among 7 categories. We used Weka[2] as a data mining software. We used

[2] http://www.cs.waikato.ac.nz/ml/weka/

stratified 10-fold cross validation for testing purposes since we find that it offers a fair coverage of the whole dataset during the tests. We compared the performance of *SMO* (Support Vector Machines) and *Naïve Bayes Multinomial*, henceforth denoted as NBM.

We started with a small number of features and then increased that number gradually i.e. we started with 400 features, then moved to 650, then 850, then 1000, then 1240, then 1400, then 1750, and finally 2000 features. As we have mentioned earlier, the feature selection process was done using Information Gain and Chi Square. The purpose was to make sure that none of the results obtained is biased by an inappropriate choice of features and to see how the performance of each classifier is affected by the size of the feature space. Figures 2, 4, 6, and 8 show respectively the accuracy of each classifier based on the aforementioned configurations. By *accuracy* we mean the percentage of correctly classified documents. Figures 3, 5, 7, 9 show respectively the F1 measures corresponding to their adjacent tables.

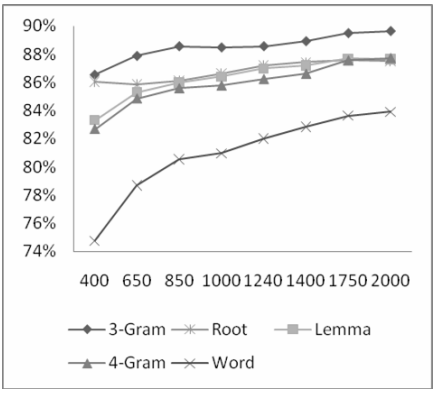


Fig. 2. Accuracy with NBM and IG

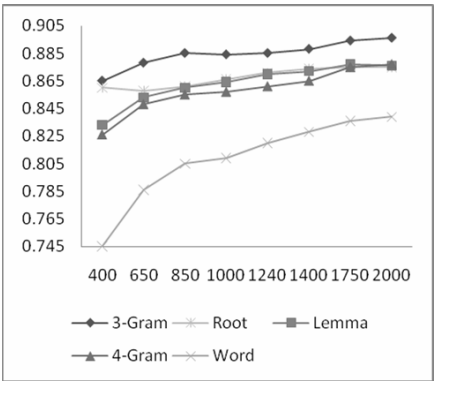


Fig. 3. F-Measure with NBM and IG

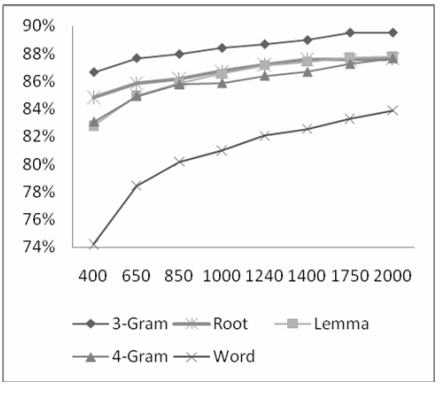


Fig. 4. Accuracy with NBM and χ^2

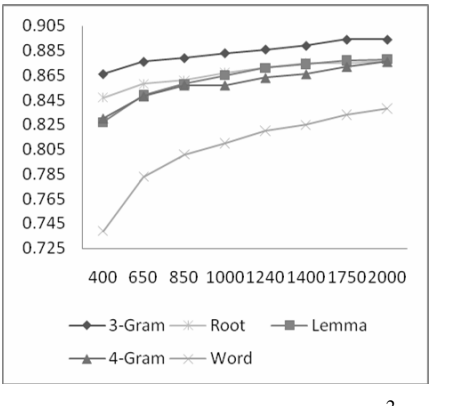


Fig. 5. F-Measure with NBM and χ^2

Table 3. Weighted-averaged experimental results using 2000 features and NBM

Feature Type used	TSR Measure Used	Precision	Recall	F1	Accuracy%
Original	χ^2	0.857	0.839	0.838	83.88
	IG	0.857	0.839	0.839	83.91
Lemma	χ^2	0.884	0.878	0.878	87.79
	IG	0.882	0.876	0.876	87.63
Root	χ^2	0.879	0.876	0.876	87.60
	IG	0.878	0.875	0.875	87.52
3-Gram	χ^2	**0.897**	**0.895**	**0.894**	**89.49**
	IG	**0.899**	**0.896**	**0.896**	**89.62**
4-Gram	χ^2	0.88	0.876	0.876	87.65
	IG	0.882	0.877	0.877	87.73

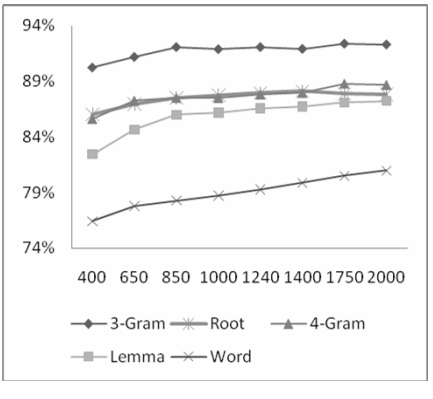


Fig. 6. Accuracy with SMO and IG

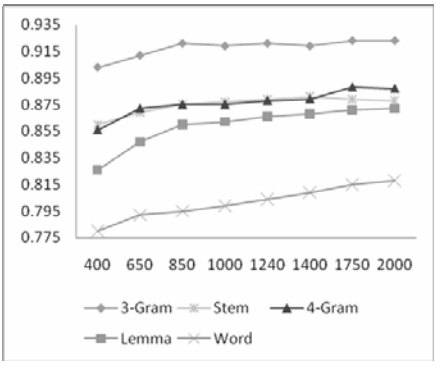


Fig. 7. F-Measure with SMO and IG

Based on the results above, we note the following: by looking at tables 3 and 4, which *only* display the evaluation metrics' values for the best performing scenario i.e. the one using 2000 features, we can see that the Naïve Bayes Multinomial algorithm,

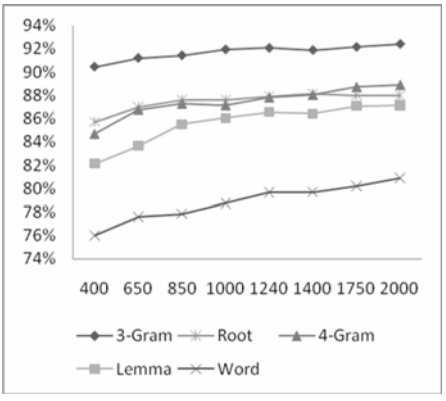


Fig. 8. Accuracy with SMO and χ^2

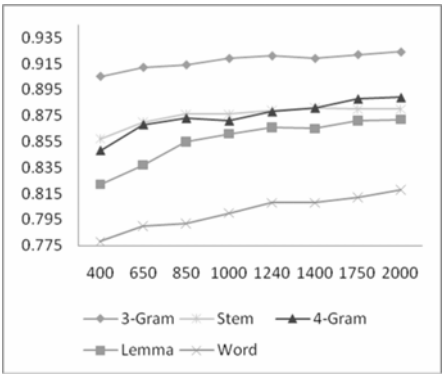


Fig. 9. F-Measure with SMO and χ^2

Table 4. Weighted-averaged experimental results using 2000 features and SMO

Feature Type used	TSR Measure Used	Precision	Recall	F1	Accuracy%
Original	χ^2	0.847	0.809	0.818	80.88
	IG	0.839	0.810	0.818	80.96
Lemma	χ^2	0.880	0.871	0.872	87.13
	IG	0.880	0.872	0.872	87.22
Root	χ^2	0.884	0.88	0.88	87.97
	IG	0.883	0.878	0.878	87.80
3-Gram	χ^2	**0.925**	**0.924**	**0.924**	**92.41**
	IG	**0.923**	**0.923**	**0.923**	**92.28**
4-Gram	χ^2	0.894	0.889	0.889	88.91
	IG	0.892	0.887	0.887	88.66

using 3-grams as a feature and χ^2 as a feature selection measure, gave the best classification with an accuracy of 89.49% and F-1 measure (0.894). Results were slightly better when using Information Gain as a feature selection measure giving an

accuracy of 89.62% and an F-1 measure of 0.896. Classification based on 4-grams came second and the one based on lemmas came third while classification based on stems came fourth and that using full-form words came last. With SMO things were a little bit different for lemmas and stems. SMO based on 3-grams were still in the first place with an accuracy of 92.28% and an F1-measure equal to 0.923. Results this time were better with χ^2 giving an accuracy of 92.41% and an F-1 measure of 0.924. SMO based on 4-grams were always second but SMO using stems led to slightly better results than lemmas and SMO based on full-form word was last.

On the other hand, looking at the whole picture i.e. looking at the graphs that display the accuracy of each classifier at each step and not only at the 2000 terms' point, we can clearly see that classification based on 3-grams still gave the best results with an accuracy better than that of the classification based on stems with a difference fluctuating between 2% and 4%. However, when we tried to consider a bigger n, e.g. n=4, the classification based on stems performed most of the time better than the one using 4-grams. We tried even to increase n to 5 but things got even worse (accuracy dropped to 83%) and classification based on stems stayed in the lead. Apparently, 4-grams and 5-grams did not do well as expected as compared to stems and this might be due to the morphologically rich nature of Arabic which allows for infixes to be widely used. The stem, in the case, will succeed in matching all of its morphological variations whereas the 4-grams and 5-grams will fail leading to a worse accuracy.

10 Conclusion

We conducted in this paper a series of experiments using five datasets generated out of the same set of documents in order to find out if any of those outperforms the others. We compared the accuracy of Support Vector Machines and Naïve Bayes Multinomial with each in order to find out which type of feature (n-gram, stem, lemma, or full-form word) leads to the best classification results. We found out that Support Vector Machines based on 3-grams gave the best classification results with an accuracy exceeding 92% and an F1 measure exceeding 0.92. However, by increasing n, as it was shown earlier, the accuracy of the classifiers degraded drastically.

Therefore, given the high inflective nature of the Arabic language and the complexity of its morphological representation, many practitioners are urged to perform a morphological analysis. We conclude that we cannot stick to a small value for n because many times this approach is considered impractical since a small n generates lots of terms which increases dramatically both the training and the classification time and might raise a serious problem when dealing with large documents and bigger datasets. As a result, we can conclude that neither of n-grams nor stems is *the* best solution. However, another set of experiments involving both is worth being conducted in a future work where a dataset composed of n-grams is generated after pre-processing the documents and then performing a morphological analysis to extract the stems out of the full-form words.

References

1. Hilbe, J.M.: Logistic Regression Models. Chapman & Hall/CRC Press (2009)
2. Forman, G.: An extensive empirical study of feature selection metrics for text classification. J. Mach. Learn. Res. 3, 1289–1305 (2003)
3. Mitchell, T.M.: Machine Learning. McGraw-Hill, New York (1997)
4. MacKay, D.: Information Theory, Inference, and Learning Algorithms (2003)
5. Pilászy, I.: Text Categorization and Support Vector Machines. In: The Proceedings of the 6th International Symposium of Hungarian Researchers on Computational Intelligence (2005)
6. Govindarajan, M.: Text Mining Technique for Data Mining Application. Proceedings of world academy of science, engineering and technology 26 (December 2007)
7. Sebastiani, F.: Machine learning in automated text categorization. ACM Comput. Surv. 34(1), 1–47 (2002)
8. Witten, I.H., Frank, E.: Data mining: practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)
9. Schapire, R.: The Boosting Approach to Machine Learning: An Overview. In: MSRI Workshop on Nonlinear Estimation and Classification (2002)
10. Abbès, R., Dichy, J.: AraConc, an Arabic Concordance Software Based on the DIINAR.1 Language Resource. In: The 6th International Conference on Informatics and Systems (2008)
11. Joachims, T.: Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In: Nédellec, C., Rouveirol, C. (eds.) ECML 1998. LNCS, vol. 1398. Springer, Heidelberg (1998)
12. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, Berlin (1995)
13. Basu, A., Watters, C., Shepherd, M.: Support Vector Machines for Text Categorization. In: Proceedings of the 36th Annual Hawaii international Conference on System Sciences (Hicss 2003) - Track 4, January 06 - 09, vol. 4, p. 103. 3. IEEE Computer Society, Washington (2003)
14. Raheel, S.: Textual Knowledge organization and information retrieval using statistical methods. In: Proceedings of the 7th Conference of the French Chapter of ISKO (2009)
15. Dichy, J.: Arabic lexica in a cross-lingual perspective. In: Proceedings of ARABIC Language Resources and Evaluation: Status and Prospects, A Post Workshop of LREC (2002)
16. Dichy, J., Braham, A., Ghazali, S., Hassoun, M.: La base de connaissances linguistiques DIINAR.1 (DIctionnaire INformatisé de l'Arabe, version 1). Paper presented at the International Symposium on the Processing of Arabic, Tunis (La Manouba), April 18-20 (2002)
17. Ghenima, M.: Analyse morpho-syntaxique en vue de la voyellation assistée par ordinateur des textes écrits en arabe. Thèse de doct., ENSSIB/Université Lyon 2 (1998)
18. Dichy, J.: Pour une lexicomatique de l'arabe: l'unité lexicale simple et l'inventaire fini des spécificateurs du domaine du mot. Meta 42, printemps, Québec, Presses de l'Université de Montréal, pp. 291–306 (1997), http://www.erudit.org/revue/meta/1997/v42/n2/002564ar.pdf
19. Zaafrani, R.: Morphological analysis for an Arabic Computer-aided learning system. In: Proceedings of DIALOGUE 1997, International Conference on computational linguistics and its applications, Yasnaya Polyana, Russia, June 10-15 (1997)

20. Ouersighni, R.: A major offshoot of the DIINAR-MBC project: AraParse, a morpho-syntactic analyzer of unvowelled Arabic texts. In: ACL 39th Annual Meeting. Workshop on Arabic Language Processing: Status and Prospect, Toulouse, pp. 66–72 (2001), http://www.elsnet.org/arabic2001/ouersighni.pd
21. Abbès, R.: Conception et réalisation d'un prototype de concordancier électronique de la langue arabe, Mémoire de DEA en Sciences de l'information et de la Communication, ENSSIB, France (1999)
22. Khreisat, L.: Arabic Text Classification Using N-Gram Frequency Statistics A Comparative Study. In: Proceedings of the 2006 International Conference on Data Mining, Las Vegas, USA, pp. 78–82 (2006)
23. Elkourdi, M., Bensaid, A., Rachidi, T.: Automatic Arabic Document Categorization Based on the Naïve Bayes Algorithm. In: Proceedings of COLING 20th Workshop on Computational Approaches to Arabic Script-based Languages, Geneva, August 23-27, pp. 51–58 (2004)
24. Mesleh, A.M.: CHI Square Feature Extraction Based SVMs Arabic Language Text Categorization System. Journal of Computer Science 3(6), 430–435 (2007)
25. Al-Shalabi, R., Obeidat, R.: Improving KNN Arabic Text Classification with N-Grams Based Document Indexing. In: Proceedings of the Sixth International Conference on Informatics and Systems, Cairo, Egypt, March 27-29 (2008)
26. El-Halees, A.: Arabic Text Classification using K-NN and Naive Bayes. The Islamic University Journal (Series of Natural Studies and Engineering) 15(1), 157–167 (2007), http://www.iugzaza.edu.ps/ara/research/
27. Raheel, S., Dichy, J., Hassoun, M.: The Automatic Categorization of Arabic Documents by Boosting Decision Trees. In: The Proceedings of the 5th International IEEE/ACM Conference on Signal-Image Technology and Internet-Based Systems. IEEE CS Press, Marrakech (2009)

Word Length n-Grams for Text Re-use Detection

Alberto Barrón-Cedeño[1], Chiara Basile[2], Mirko Degli Esposti[2], and Paolo Rosso[1]

[1] NLEL-ELiRF, Department of Information Systems and Computation, Universidad Politécnica de Valencia, Spain
{lbarron,prosso}@dsic.upv.es
http://www.dsic.upv.es/grupos/nle/
[2] Dipartimento di Matematica, Università di Bologna, Italy
{basile,desposti}@dm.unibo.it

Abstract. The automatic detection of shared content in written documents –which includes text reuse and its unacknowledged commitment, plagiarism– has become an important problem in Information Retrieval. This task requires exhaustive comparison of texts in order to determine how similar they are. However, such comparison is impossible in those cases where the amount of documents is too high. Therefore, we have designed a model for the proper pre-selection of closely related documents in order to perform the exhaustive comparison afterwards. We use a similarity measure based on word-level n-grams, which proved to be quite effective in many applications As this approach becomes normally impracticable for real-world large datasets, we propose a method based on a preliminary word-length encoding of texts, substituting a word by its length, providing three important advantages: (i) being the alphabet of the documents reduced to nine symbols, the space needed to store n-gram lists is reduced; (ii) computation times are decreased; and (iii) length n-grams can be represented in a trie, allowing a more flexible and fast comparison. We experimentally show, on the basis of the perplexity measure, that the noise introduced by the length encoding does not decrease importantly the expressiveness of the text. The method is then tested on two large datasets of co-derivatives and simulated plagiarism.

Keywords: word length encoding; text similarity analysis; text reuse analysis; plagiarism detection; information retrieval.

1 Introduction

Similarity between documents is a key factor in diverse Natural Language Processing and Information Retrieval (IR) tasks such as documents clustering and categorization [5]. Problems that require a deeper analysis of similarity between texts are text-reuse analysis [7], co-derivatives analysis [4], information flow tracking [14], and plagiarism detection [13]. In these tasks, we are not only interested in looking up how many keywords a pair of documents have in

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 687–699, 2010.

common, but in how related their contents are. While this could be considered as a semantic problem, different methods based on chunks comparison, a purely syntactic approach, have shown competitive results [13].

The exhaustive comparison of entire documents is a hard task; comparing strings is computationally complex and defining the best chunks to be compared is not straightforward. On the one hand, comparison techniques have been designed on the basis of fingerprint models, such as Winnowing [16]. Fingerprinting is often based on the sub-sampling of text chunks and an information loss must be assumed when opting for these methods. On the other hand, when a comparison of the entire content of the document is required, character and word-level n-grams have shown to be a good option [6].

Detection of text reuse, co-derivatives, and plagiarism can be divided into three steps (cf. [17]): (*i*) heuristic retrieval of potential source documents —given a document, retrieving a proper amount of its potential source documents—; (*ii*) exhaustive comparison of texts —comparing the texts in order to identify those fragments which could be re-used and their potential sources—; and (*iii*) knowledge-based post-processing (only for plagiarism detection) —proper citations are eliminated from the plagiarism candidate fragments—. Nevertheless, research on these tasks often approaches step (*ii*) only, assuming that the rest are solved [12,10,6]. However, this is not true. Note that step (*i*) is a more specific case of clustering and IR: instead of grouping/retrieving a set of related documents, the task is to define a reduced set of potential source documents containing texts with a high probability of being the source of the text fragments in the analysed document.

Hereinafter we propose a method to approach step (*i*). We make it by estimating how close two documents are on the basis of the so named *length encoding*. The method encodes every word in the implied texts by its length in characters and splits the resulting text into n-grams. The comparison between documents can be then performed on the basis of standard measures such as the cosine distance or the Jaccard coefficient [8]. The method is tested on two corpora of simulated plagiarism and text co-derivatives showing promising results.

The remainder of the paper is laid out as follows. Section 2 gives a description of the two corpora we have used in our experiments. Section 3 describes the length encoding method, including an empirical analysis of its validity based on language models and perplexity. Section 4 includes the experiments we have carried on in order to compare how well the model works with respect to a "traditional" word-level n-gram comparison model. Finally, Section 5 draws conclusions and outlines future work.

2 Corpora

In order to perform our experiments we used two datasets: the PAN-PC-09 corpus and the Wikipedia co-derivatives corpus.[1]

[1] Both corpora are available at `http://www.dsic.upv.es/grupos/nle/downloads.html`

2.1 PAN-PC-09 Corpus

The PAN-PC-09 [15] corpus was created in the framework of the 1st International Competition on Plagiarism Detection[2]. This freely available resource for the evaluation of plagiarism detection methods is divided into development and test sections. The former can be used in order to tune up and test models as it includes annotations on the plagiarism cases as well as their sources. This is the section we used for our experiments. It contains 7214 source documents and 7214 suspicious documents. Further descriptions on this corpus are available in [2].

2.2 Co-derivatives Corpus

This corpus was generated for the analysis of co-derivatives, text reuse and (simulated) plagiarism. It is composed of more than 20,000 documents from Wikipedia in four different languages: English, German, Spanish and Hindi. It contains around 5,000 documents for each language, including some of the most frequently accessed articles in Wikipedia. For each article ten revisions were downloaded, composing the set of co-derivatives. The corpus pre-processing includes whitespace normalization, sentence detection, tokenization and case folding. An extensive description of the corpus construction can be found in [2].

3 Method Definition

3.1 Notation

The notation used throughout the rest of the paper is the following. Let D be the set of all reference documents and $\{d_q\}_{q \in Q}$ the set of query documents; these will be either the texts which are suspected of containing plagiarism (PAN-PC-09 corpus) or the most recent revision of each Wikipedia article (co-derivatives corpus). The query documents can be contained in D or not, depending on the experiment. For a document $d \in D$, let $V_n(d)$ be the set of all $n-$grams in d (the *n-gram vocabulary*). Let $D_q \subseteq D$ be the set of the first k neighbours of d_q according to some similarity measure. Let $L_q \subseteq D$ be the set of source documents of the re-used text in d_q: L_q contains, in the first case, all the sources for the plagiarism in d_q, as described in the development section of the PAN-PC-09 corpus, and in the second case it is composed of the 10 revisions of the Wikipedia article, the last of which is precisely d_q.

The goal of the method is to maximize the intersection between L_q and D_q, without increasing too much the number of retrieved texts k.

3.2 Length Encoding

The length encoding model was formerly introduced in [3], where it was used to reduce the search space for the PAN-PC-09 competition dataset. It takes the

[2] `http://www.webis.de/pan-09/competition.php`

idea of word-level n-grams comparison but, instead of comparing word strings, it compares length strings, that in fact become integer numbers. Let w be a word in a given text and let $|w|$ be its length in characters. The steps of the length encoding, including a brief example to illustrate, are the following:

Input:	`This UFO related place is the so-called 'area 51'.`
Pre-processing: substitute any non-letter symbol with a blank space.	`This UFO related place is the so called area`
Encoding: replace each word w with $\min(\|w\|, 9)$	4 3 7 5 2 3 2 6 4

After length encoding the document, n-grams can be obtained to characterise it. For instance, by considering $n = 5$, the resulting n-grams are: {`43752, 37523, 75232, 52326, 23264`}. Such n-grams can be handled as integers instead of strings, causing a saving of memory space (integers, indeed, occupy less space than strings, as discussed afterwards) and accelerating the comparison process. Note that if the "traditional" word n-gram schema is followed, the 5-grams for the example sentence above are: {`this ufo related place is`, `ufo related place is the`, `related place is the so`, `place is the so called`, `is the so called area`}. Clearly, adopting the usual approach also requires a lower-casing process, which is unnecessary with our length encoding.

Nevertheless, information is still redundant in the length n-gram list. In order to reduce redundancy, it is possible to profit from the limited vocabulary these n-grams are composed of. As the vocabulary is $\alpha = \{1, 2, 3, \ldots, 9\}$, indeed, it is straightforward to compose a *trie* (also known as prefix tree) to represent the entire document. This is very difficult when considering the actual document vocabulary. Figure 1 contains the trie characterization of the sample text given before. For instance, in the example the 1-gram `4` (corresponding to both `this` and `area`) appears twice in the text, while the 4-gram `4375`, corresponding to `This UFO related place` (third branch counting from the left and going down to the fourth generation from the root), appears once. The advantages of the proposed method are the following:

1. Computational time is significantly reduced.
2. The space occupied by the encoded documents is reduced with respect to the one needed to encode word n-grams or the list of length $n-$grams.
3. All the n-grams for $n \in \{1, \ldots, N\}$, N being the depth of the trie, are available in the trie itself.

Regarding points 1 and 2, consider that, as beforementioned, instead of strings (be of characters or numbers), integers can be used. Integers can be handled on 32 or 64 bits, whereas strings are composed of chains of 16-bit characters (an average word of 4 characters occupies 64 bits). Additionally, comparing integers is much faster than comparing strings. It is true that other techniques, such

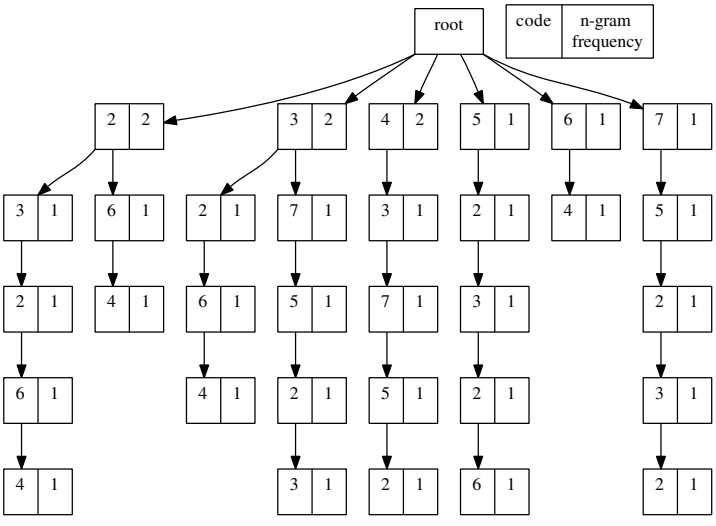


Fig. 1. Length encoding trie for the text "`This UFO related place is the so-called 'area 51'.`" (encoded `4 3 7 5 2 3 2 6 4`). Each node x includes the code of a word (i.e., its length) as well as the frequency of the n-gram which can be read on that branch from the root of the trie down to node x.

as inverted index [1], might decrease even more the calculation time. However, when considering a realistic open retrieval problem (where the set of documents we are searching on has not been previously defined), its creation is not feasible. Also building an inverted index of n-grams, where almost every input appears in one only document, does not make sense. Table 1 includes the size of the files and data structures where the different versions of the text are saved. The data correspond to a sample of 1000 documents from the development section of the PAN-PC-09 corpus. Note that first and last rows include values for the total column only. The encoding based on word n-grams, for $n \in \{1, \ldots, 10\}$, occupies an order of around 60 times the size of the original document d, whereas the length encoding requires only 20 times the size of d. When using the trie data structure, the order decreases to around 15 times the size of d.

Table 1. Average size of the documents (in Kbytes) by considering different encoding strategies and n-gram levels. d = original document; wng = word n-grams; lng = length n-grams; total=total space occupied; rel=total/size(d)

	n-gram level											
encoding	1	2	3	4	5	6	7	8	9	10	total	rel
d											163	1
wng	54	319	605	814	991	1158	1323	1487	1651	1814	10216	62.67
lng	0.27	1	8	45	170	378	549	650	720	781	3302	20.26
$trie$											2557	15.69

With respect to point 3, one single data structure includes the n-grams of all levels (up to a given threshold) in the document. As a result, the comparison can be carried on by considering any value of n without further processing. This makes the comparison strategy much more flexible, which is not possible, for instance, when considering fingerprinting models as Winnowing or SPEX).

The length encoding model certainly adds noise to the texts, since at low levels of n a lot of different text strings are translated into the same code. However, when increasing n, the noise decreases, becoming at last irrelevant. In order to show that, we exploit the concept of perplexity, an entropic measure which estimates the uncertainty of a language model (cf. [9]). Table 2 shows the values of perplexity, divided by the cardinality of the corresponding $n-$gram dictionary, for both word and length $n-$grams and with $n = 1, \ldots, 6$. A convergence of the perplexity of the length $n-$gram model to that of the word $n-$gram model is evident, even if it was not possible to calculate the value for larger levels of n, because perplexity is a sentence-level measure, and sentence-end effects could affect the calculation for larger n.

Table 2. Perplexity for the different level language models, divided by the cardinality of the corresponding $n-$gram dictionary. *wng* = word n-grams; *lng* = length n-grams

encoding	$n = 1$	$n = 2$	$n = 3$	$n = 4$	$n = 5$	$n = 6$
wng	9.2×10^{-3}	1.4×10^{-4}	3.9×10^{-5}	2.7×10^{-5}	2.4×10^{-5}	2.3×10^{-5}
lng	9.6×10^{-1}	1.0×10^{-1}	1.1×10^{-2}	1.2×10^{-3}	1.4×10^{-4}	1.6×10^{-5}

In this context, it is also interesting to observe the distribution of $n-$gram frequencies in a large dataset. In Fig. 2 the data are reported for the distributions in a set composed of 500 documents extracted from the PAN-PC-09 corpus, with both length and word $n-$grams. Note that, as long as n grows, the two distributions tend to coincide, and a large superimposition is reached already for $n = 12$. This observation supports empirically the intuitive idea that, for a large enough n, the length encoding is "almost injective", i.e., very few word $n-$grams are mapped to the same length $n-$gram.

3.3 Similarity Estimation

The similarity measure we opted for is the Jaccard coefficient [8], a very standard indicator based only on a comparison between the $n-$gram vocabularies of the two texts into consideration, totally disregarding $n-$gram statistics. This is a good measure in such cases where the value of n is large enough to make very unlikely that an $n-$gram repeats more than a few times in a text: when considering word $n-$grams, this happens already with $n = 3, 4$ (see for example the case of $n = 5$ in Fig. 2), which are certainly appropriate values in this case where we are considering text re-use cases. The Jaccard coefficient between texts d_q and $d \in D$ is defined as follows ($|\cdot|$ indicating here the cardinality of a set):

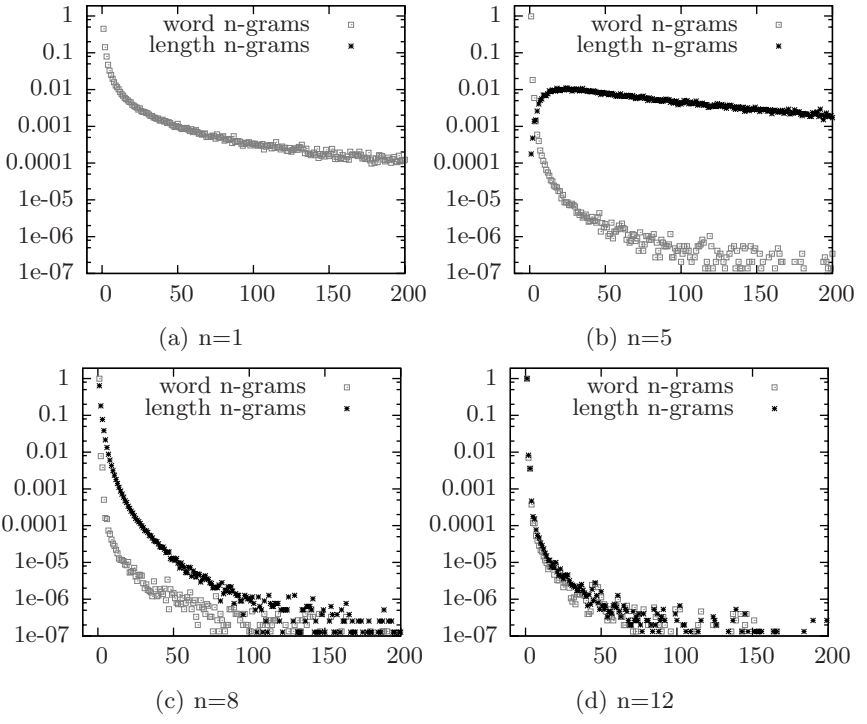


Fig. 2. Frequency distributions for length $n-$grams (*black stars*) and for word $n-$grams (*gray squares*), for some values of n. The number of occurrences lies on the $x-$axis, with the corresponding percentage of n-grams on the $y-$axis. The length $n-$gram distribution converges to the one of word $n-$grams as n grows. No stars appear in the first plot because we show up to 200 occurrences only, which is lower than the frequency of any possible 1-gram of length encoded text in a representative corpus.

$$J_n(d_q, d) := \frac{|V_n(d_q) \cap V_n(d)|}{|V_n(d_q) \cup V_n(d)|}. \tag{1}$$

J_n takes values in the interval $[0, 1]$. It is closer to 1 as long as the superimposition between the vocabularies of d_q and of d is larger.

4 Experiments

We performed experiments in order to compare a common word n-gram encoding to the proposed length n-gram encoding. Evaluation in terms of Recall was carried out by considering two corpora including cases of simulated plagiarism and text co-derivatives (cf. Sections 2.1 and 2.2, respectively). Results are shown in Sections 4.1 and 4.2. The methods were compared in terms of time-performance also, with results shown in Section 4.3.

4.1 Experiments on the PAN-PC-09 Corpus

Experiment Outline. The first experiment on this corpus (**exp1** hereinafter) has the aim of verifying the appropriateness of the length encoding for the recognition of relevant documents for plagiarism cases. In order to identify the appropriate value of n for such task, we first used a repeated sampling technique: for every run, we selected a small random subset $\{d_q\}_{q\in\tilde{Q}}$ of query documents and an appropriate subset $\tilde{D}$ of reference documents, and evaluated the performance. The value of n that performed the best was then used to apply the length encoding method to the whole PAN-PC-09 development corpus.

We have already justified the use of word length encoding in Section 3 however, we also wanted to compare our method with the one based on "traditional" word-level n-grams. Therefore, we performed a second experiment (**exp2** hereinafter) where we selected a subset of the corpus (the same used in [3], composed of 160 query texts and 300 reference documents), and calculated the Jaccard coefficient for both length and word $n-$grams, with $n = 2, 4, \ldots, 20$.

For both experiments we fixed k, the number of retrieved documents, to the value of 10, in agreement with the co-derivatives experiment (Section 4.2).

Measures of Performance. There are various possible definitions of the *recall* for this problem; the choice of the right one depends on what we want to measure precisely. First of all, we have to choose between a single query text average or a global average. To avoid problems of divisions by zero for those query documents not containing plagiarism, we decided to use a global measure, following the approach of [15].

Another choice is whether we want to measure the fraction of recalled *source texts* from which the plagiarism comes or that of the *plagiarised characters* contained in the selected source texts. In the PAN-PC-09 corpus every query document has an associated XML file with detailed annotation about the copied sections, with character-level precision. In order to take advantage from this annotation, we used for both **exp1** and **exp2** the following character-level measure:

$$R_c@k := \frac{\sum_q \sum_{s\in\Delta_q} |s|}{\sum_q \sum_{s\in\Lambda_q} |s|} \ , \tag{2}$$

where Λ_q is the set of all plagiarised sections in d_q, $\Delta_q \subseteq \Lambda_q$ is the set of plagiarised passages in d_q that come from its first k neighbours according to the $n-$gram distance into consideration (i.e., from the selected texts in L_q), and $|s|$ expresses here the length of passage s, measured in characters. This measure gives a larger weight to longer copied passages, in the same spirit of the measure used for the Competition on Plagiarism Detection [15].

Since half of the query documents in the PAN-PC-09 are entirely original, containing no plagiarised passages, we did not calculate any *precision* measure.

Results. The results of **exp1** are given in Fig. 3(a). The recall $R_c@10$ is greater than 0.8 for all values of n larger than 8, and it reaches its maximum for $n = 12$.

An important observation is that identifying the relevant texts in such small samples of the corpus is much simpler, from a purely statistical viewpoint, than the "real" task of detecting few relevant texts for each suspicious document in the whole dataset of 7,214 sources. At this point, thus, having identified 12 as a proper value for n, we calculated the Jaccard coefficient J_{12} on the whole PAN-PC-09 development corpus and obtained a recall $R_c@10 = 0.86$, a value even higher than the one shown in Fig. 3(a) for $n = 12$ with the small samples.

Considering that 13% of the plagiarism cases in the corpus are cross-language (cf. [15]) and the method we propose here has no hope of retrieving such cases, we consider that a recall above 0.85 is a very good result.

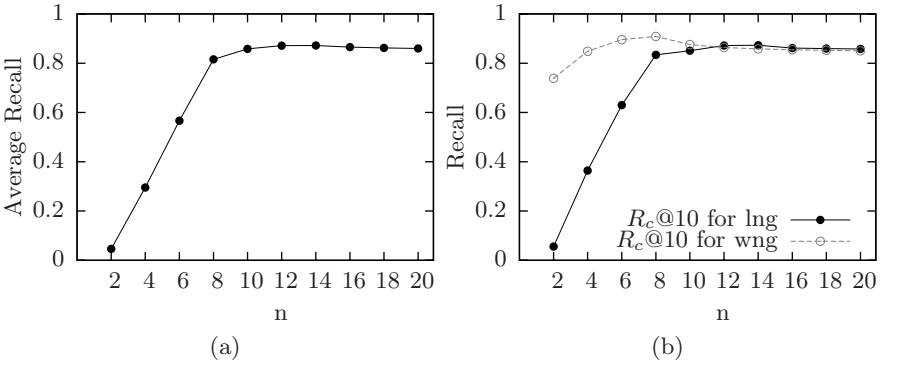


Fig. 3. Recall calculated as in Eq. (2) for the PAN-PC-09 corpus (a) by averaging over 100 samples of 150 random query texts and around 300 reference documents, with length encoding; and (b) compared to word $n-$grams.

Figure 3(b) shows the results of **exp2**. The obtained results confirm what we expected from Section 3: there exists a threshold for n, here around $n = 12$, above which the length encoding and the word $n-$gram methods are perfectly equivalent; to be true, here the encoding method performs always slightly better than word $n-$grams, for $n \geq 12$, but such small differences may not be reliable due to the fact that we are using a small subset of the corpus. This value of n is in concordance with the one used in fingerprinting models such as SPEX [4].

4.2 Experiments on the Co-derivatives Corpus

Experiment Outline. Even if (artificial) plagiarism and Wikipedia collaborative writing are very different phenomena, they can be considered as two sides of the same problem of text re-use identification, and can be stated in the same terms of query document - source documents association. Note, however, that from the viewpoint of the experimental setting there are two main differences between the two corpora. First of all, here the query set is composed of the last

revision of each article, and its 10 revisions, included itself, constitute the reference set: therefore, the set $\{d_q\}_{q \in Q}$ is in this case included in D, in agreement with [2]. Secondly, the set L_q of relevant sources for d_q has in this case a cardinality of 10 for each d_q; therefore, it is even more natural here to choose $k = 10$ as the number of retrieved texts ($|D_q|$).

We followed for this corpus the same outline described in Section 4.1, except for the fundamental differences stated above. In the first experiment (**exp3** hereinafter) we calculated the Jaccard coefficient J_n for various values of n with word length $n-$grams. The second experiment (**exp4** hereinafter) was aimed at comparing the results obtained by considering the length encoding and the traditional word $n-$grams.

Measures of Performance. Since this corpus does not contain any character-level annotation for the revisions of Wikipedia articles, the only measure of recall which applies here is the global text average $R_t@k$, defined as follows:

$$R_t@k := \frac{\sum_q |L_q \cap D_q|}{\sum_q |L_q|} = \frac{\sum_q |L_q \cap D_q|}{k|Q|} \quad , \tag{3}$$

i.e., simply the fraction of relevant documents that the method identifies as such. Since for this corpus the number of retrieved texts k corresponds, for each query text, to the number of relevant documents, the values of precision and recall coincide, i.e. $R_t@10 = P_t@10$.

Results. In Table 3 we report the recall $R_t@10$ for **exp3**, calculated as in Eq. (3), with $|L_q| = |D_q| = 10$ and $n \in \{2, 4, \ldots, 20\}$.

Table 3. Recall $R_t@10$ for the co-derivatives corpus with the Jaccard coefficient on word-length n-grams, varying n

	2	4	6	8	10	12	14	16	18	20
en	0.02396	0.98970	0.99366	0.99465	0.99485	0.99485	0.99465	0.99426	0.99426	0.99406
de	0.22080	0.92911	0.96673	0.97663	0.97703	0.97604	0.97426	0.97208	0.97109	0.96812
es	0.10218	0.90159	0.96198	0.96812	0.96713	0.96495	0.96277	0.96040	0.95723	0.95485
hi	0.45545	0.74495	0.81683	0.84792	0.85010	0.84337	0.83525	0.82950	0.82257	0.81426

In concordance with [2], the results are much better for the English subcorpus than for the Hindi one; the other two languages are located in between, with quite good results. This could also be an effect of the difference in average length of the articles in the four different languages.

Table 4 shows the results of **exp4** with n ranging from 2 to 10 and for the Spanish corpus, which was chosen as the dataset here because the article length is proper and the similarity distribution is adequate for experiments. Again, the results of the two techniques are perfectly equivalent, with the length encoding performing slightly better than word $n-$grams for all values of n larger than 10.

Table 4. Recall R_t@10 for the Spanish section of the co-derivatives corpus, with word $n-$grams (wng) and length $n-$grams (lng), varying n

encoding	2	4	6	8	10	12	14	16	18	20
wng	0.9703	0.9762	0.9737	0.9697	0.9657	0.9622	0.9598	0.9566	0.9541	0.9497
lng	0.1022	0.9016	0.9620	0.9681	0.9671	0.9649	0.9628	0.9604	0.9572	0.9548

The very low recall obtained in all experiments with length bigrams has a very simple statistical explanation. Since the possible bigrams in the alphabet $\{1, \ldots, 9\}$ are just $9^2 = 81$, and since we are considering only a combinatorial measure, disregarding any information about the frequency (this is the essence of the Jaccard coefficient), with high probability all the bigrams appear in each text of the corpus, giving a value 1 for J_2 in any case. Therefore, the selection of the first k neighbours corresponds to a random extraction of k source documents.

4.3 Experiments on Process Speed

We showed experimentally that the length n-gram model performs comparably to the word n-gram model for a proper value of n. Now, we compare the models in terms of processing speed.

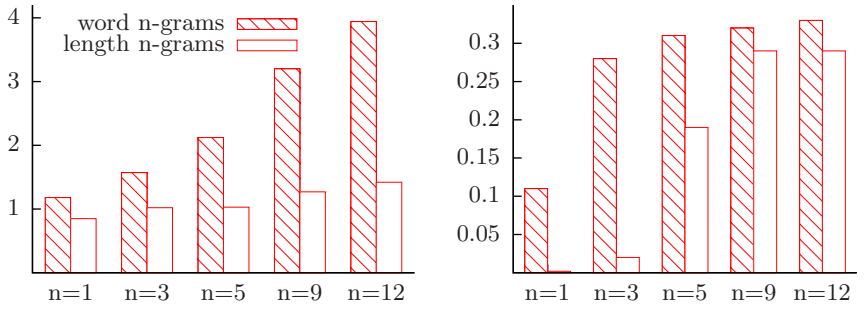
In the first experiment (**exp5** hereinafter), we compared the time needed to encode a text document into either a set of word n-grams or a trie of length n-grams. For this estimation $1,000$ random documents from the PAN-PC-09 corpus were considered.

In the second experiment (**exp6** hereinafter), we compared the time required to compare the document representation by calculating the Jaccard coefficient (*comparison*). In order to perform this experiment 50 suspicious and source documents from the PAN-PC-09 corpus were considered, resulting in $2,500$ comparisons for each value of n.

The obtained results for both experiments are shown in Figure 4. In both cases different values of n were considered: $\{1, 3, 5, 9, 12\}$. From **exp5**, it is clear that the length encoding is much faster than the word encoding. On average, the length encoding takes a half of the time needed to perform the word encoding. This is due to two main reasons. First, in order to compare word n-grams the text must be converted to lowercase, an operation which is unnecessary for the length encoding. Additionally, as less memory is used to save the trie than the set of word n-grams, the resources are used in a more efficient way in the first case.

Experiment **exp6** clearly shows that also the time required to compare length n-grams is shorter than the time needed to compare word n-grams.

Disregarding the precise numerical results, which depend on the specific hardware used, the difference in performance is evident in both experiments and confirms this further advantage of length encoding.



(a) encoding (b) comparison

Fig. 4. Time needed for the encoding and comparison steps, by using word and length n-grams. The values are expressed in seconds and are averages of 2,500 processes.

5 Conclusions

In this paper we approached the problem of the preliminary selection of closely related texts, the first step for text-reuse, co-derivatives analysis, and automatic plagiarism detection. The method we proposed to solve this task encodes the documents on the basis of their word lengths. Whereas some efficient methods, such as fingerprinting, imply a loss of information between the actual document and its fingerprint, our method reduces such loss, as we empirically showed.

Retrieval experiments were performed on two corpora: the first one of simulated plagiarism and the second one of text co-derivatives. The obtained results show that representing the documents by the length encoding: (*i*) does not affect the performance of the retrieval process; (*ii*) favours a flexible comparison of documents as n-grams of any level are available in the text representation; and (*iii*) the entire encoding and comparison process is speeded up, an important factor when the amount of comparisons to perform is significant.

As future work we plan to combine this method with a selection of representative chunks on the basis of entropic methods. Moreover we will compare the similarity measure to a different one such as the Kullback-Leibler distance [11].

Acknowledgements. This work was partially funded by the CONACYT-Mexico 192021 grant, the Text-Enterprise 2.0 TIN2009-13391-C04-03 project, and the INdAM-GNFM Project for Young Researchers "Sequenze, sorgenti e fonti: sistemi dinamici per le misure di similarità".

References

1. Baeza-Yates, R., Ribeiro-Neto, B.: Modern information retrieval, p. 192. Addison-Wesley Longman, Amsterdam (1999)
2. Barrón-Cedeño, A., Eiselt, A., Rosso, P.: Monolingual Text Similarity Measures: A Comparison of Models over Wikipedia Articles Revisions. In: Proceedings of the ICON 2009: 7th International Conference on Natural Language Processing, pp. 29–38. Macmillan Publishers, Basingstoke (2009)

3. Basile, C., Benedetto, D., Caglioti, E., Cristadoro, G., Degli Esposti, M.: A plagiarism detection procedure in three steps: selection, matches and "squares". In: Stein, B., Rosso, P., Stamatatos, E., Koppel, M., Agirre, E. (eds.) SEPLN 2009 Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse (PAN 2009), pp. 1–9. CEUR-WS.org (2009)
4. Bernstein, Y., Zobel, J.: A Scalable System for Identifying Co-Derivative Documents. In: Apostolico, A., Melucci, M. (eds.) SPIRE 2004. LNCS, vol. 3246, pp. 55–67. Springer, Heidelberg (2004)
5. Bigi, B.: Using Kullback-Leibler distance for text categorization. In: Sebastiani, F. (ed.) ECIR 2003. LNCS, vol. 2633, pp. 305–319. Springer, Heidelberg (2003)
6. Broder, A.Z.: On the Resemblance and Containment of Documents. In: Compression and Complexity of Sequences (SEQUENCES 1997), pp. 21–29. IEEE Computer Society, Los Alamitos (1997)
7. Clough, P., Gaizauskas, R., Piao, S., Wilks, Y.: Measuring Text Reuse. In: Proceedings of Association for Computational Linguistics (ACL 2002), Philadelphia, PA, pp. 152–159 (2002)
8. Jaccard, P.: Étude comparative de la distribution florale dans une portion des Alpes et des Jura. Bulletin del la Société Vaudoise des Sciences Naturelles 37, 547–579 (1901)
9. Jurafsky, D., Martin, J.H.: Speech and Language Processing: An introduction to natural language processing, computational linguistics, and speech recognition, 2nd edn. Prentice-Hall, Englewood Cliffs (2009)
10. Kang, N., Gelbukh, A., Han, S.-Y.: PPChecker: Plagiarism pattern checker in document copy detection. In: Sojka, P., Kopeček, I., Pala, K. (eds.) TSD 2006. LNCS (LNAI), vol. 4188, pp. 661–667. Springer, Heidelberg (2006)
11. Kullback, S., Leibler, R.: On information and sufficiency. Annals of Mathematical Statistics 22(1), 79–86 (1951)
12. Lyon, C., Malcolm, J., Dickerson, B.: Detecting Short Passages of Similar Text in Large Document Collections. In: Conference on Empirical Methods in Natural Language Processing, Pennsylvania, pp. 118–125 (2001)
13. Maurer, H., Kappe, F., Zaka, B.: Plagiarism - A Survey. Journal of Universal Computer Science 12(8), 1050–1084 (2006)
14. Metzler, D., Bernstein, Y., Croft, B.W., Moffat, A., Zobel, J.: Similarity Measures for Tracking Information Flow. In: Conference on Information and Knowledge Management, pp. 517–524. ACM Press, New York (2005)
15. Potthast, M., Stein, B., Eiselt, A., Barrón-Cedeño, A., Rosso, P.: Overview of the 1st International Competition on Plagiarism Detection. In: Stein, B., Rosso, P., Stamatatos, E., Koppel, M., Agirre, E. (eds.) SEPLN 2009 Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse, PAN 2009, pp. 1–9. CEUR-WS.org (2009)
16. Schleimer, S., Wilkerson, D.S., Aiken, A.: Winnowing: Local Algorithms for Document Fingerprinting. In: 2003 ACM SIGMOD International Conference on Management of Data. ACM, New York (2003)
17. Stein, B., Meyer zu Eissen, S., Potthast, M.: Strategies for Retrieving Plagiarized Documents. In: Clarke, C., Fuhr, N., Kando, N., Kraaij, W., de Vries, A. (eds.) 30th Annual International ACM SIGIR Conference, pp. 825–826. ACM, New York (2007)

Who's the Thief? Automatic Detection of the Direction of Plagiarism

Cristian Grozea[1,⋆] and Marius Popescu[2,⋆⋆]

[1] Fraunhofer Institute FIRST,
Kekulestrasse 7, 12489 Berlin, Germany
cristian.grozea@first.fraunhofer.de
[2] University of Bucharest,
Faculty of Mathematics and Computer Science,
Academiei 14, Sect. 1,
Bucharest, Romania
popescunmarius@gmail.com

Abstract. Determining the direction of plagiarism (who plagiarized whom in a given pair of documents) is one of the most interesting problems in the field of automatic plagiarism detection. We present here an approach using an extension of the method Encoplot, which won the 1st international competition on plagiarism detection in 2009. We have tested it on a large-scale corpus of artificial plagiarism, with good results.

Keywords: plagiarism detection, plagiarism direction, linguistic forensics.

1 Introduction

Plagiarism is a phenomenon of increasing importance, as it is nowadays facilitated by the multitude of sources accessible through internet. It has hit also the academic world, see *dejavu* [6] for a surprisingly extensive database of plagiarized articles in the medical research, exhibiting among others cross-language plagiarism. In education, there are efforts to fight it using commercial services like Turnitin [2] and in-university developed systems [7] and [8]. A recent competition evaluated many methods of plagiarism detection [10]. The methods participating achieved fairly good results in detecting plagiarism. But detecting plagiarism is only half of the problem. The very next question is who copied after whom, who is the thief and who is the victim – although in some cases, the source is none of the two but some third party. Time stamps of some sort can easily prove the priority, but they are not always available, or are too easy to forge (file dates, timestamps on webpages) to be trusted. When two students (or two researchers) present in the same time work that is too similar, how could one know which is the original and which is the copy? There is even a case involving people as famous as Einstein and Hilbert on a subject as important as

⋆ Corresponding author.
⋆⋆ Research supported by CNCSIS, PNII-Idei, project 228.

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 700–710, 2010.

the theory of relativity, that was even 80 years later still a matter of debate – for details see [5], [14]. Wouldn't it be nice if the proof of originality could be found in the work itself, not in some – maybe unavailable, maybe untrusted – priority timestamp?

1.1 Related Work

Conceptually, the problem of detecting the plagiarism direction is very related to the problem of detecting stylistic changes and inconsistencies like in the intrinsic plagiarism detection and authorship attribution. If a good measure of stylistic similarity is available, this measure can be used for detecting plagiarism direction. Suppose that one is given two texts and an alleged plagiarized text fragment that belongs to both texts. Then, the stylistic similarity between the alleged plagiarized text fragment and others fragments from the two texts can be measured, and the text that is most similar (stylistically consistent) with the alleged plagiarized text fragment will be considered to be the "original", while the less similar text will be considered to be plagiarizing one.

One related research area where the problem is also the identification of which text is the copy and which text is the source is computational stemmatology "Given a collection of imperfect copies of a textual document, the aim of stemmatology is to reconstruct the history of the text, indicating for each variant the source text from it was copied." [11]

The methods used there are phylogenetic methods borrowed from evolutionary biology. Maybe it is not by chance that the only works that address the problem of plagiarism direction [13,12] are also based on phylogenetic methods.

We are aware of no plagiarism detection methods able to identify the true source. In general, the first come is treated by the system as being the source and the second one as copying the source. Many measures developed for plagiarism detections are distances, and as such, symmetric. They consider the "effort" of going from the first text to the second text identical with the one needed to get back. Only by breaking this symmetry could one hope to obtain the information of the direction of plagiarism. Even in [12] where the developed methods target explicitly the creation of a phylogenetic tree of evolution of internet news, a complex time-space asymmetric measure is created for this, which is asymmetric, but simple timestamps (article creation time) are used in the computation that eventually decides for each direction the probability of filiation (and implicitly on which is the source and which is the copy).

2 Methods

2.1 Dataset

To approach this problem we have used the newly published plagiarism corpus [15], that has been created in order to allow for a common base of evaluation of the plagiarism detection methods in the aforementioned competition. It is a

multi-language, large-scale, public corpus of plagiarism, containing only artificial plagiarism instances. The random plagiarizing tried to mimic the attempts a human would make to hide the copying, by obfuscating to a certain degree (through reordering of the phrases, replacing words with synonyms or antonyms, deletions, insertions and changes of the words used). Also, some of the instances involve also a translation of the copied passage in the process of going from the source to the destination text, done by automatic means. The external plagiarism section of the corpus contains 14429 source documents (obtained from the Project Gutenberg [1] archive), 14428 "suspicious" documents, and 73522 plagiarized passages. The suspicious documents are also from the Project Gutenberg archive, in which random passages from the sources have been transfered with the transformations mentioned before. The documents are up to book length.

2.2 Finding the Asymmetry

Two of the methods used in the competition are employing dotplot-like analysis [4] to detect and examine the plagiarism: [3] and [9].

In the figures in both of these papers one could observe the parasitic unwanted dots that appear, in addition to the ones useful for recovering the plagiarized passages.

We have used here the second method, our own "encoplot", for which we have published the source code in [9] and which outperformed all others in the challenge. Back then we were already hinting that this method could be of use to identifying the direction of plagiarism, as we noted an asymmetry there: "it is 10% better to rank all suspicious documents for any fixed source instead of ranking all possible sources for a suspicious document (...) This asymmetry deserves more investigation, being one of the few hints of hope so far to tackling what could be the biggest open problem in automatic plagiarism detection, that is determining the direction of plagiarism in a pair of documents".

We have now found another asymmetry that is more useful than that, as it only concerns the two texts involved into a pairwise comparison. Figure 1 shows one example of "encoplot" for the source document #2400 ("Poems" by William Cullen Bryant) and the suspicious document #2 (based on "Our Churches and Chapels" by "Atticus" A. Hewitson, with changes introduced through randomly plagiarizing from two sources) in this corpus.

In Figure 2 the same pair of documents is processed, just that with twice shorter character-based n-grams (n=8 bytes). One can easier observe the parasitic clouds of dots which tend to elongate like a trace pointing to the copying document axis, parallel with that of the source text. We have used 8-grams throughout the experiment.

The apparition of these clouds is a consequence of the way encoplot works: it pairs the first instance of an n-gram in a text with the first instance of the same n-gram in the other, the second instance of it with the second one in the other text, the third with the third, and so on [9]. When the passage is copied without obfuscation and the n-grams of the passage have a single instance in the source and the copy documents, a perfect diagonal appear, without any clouds.

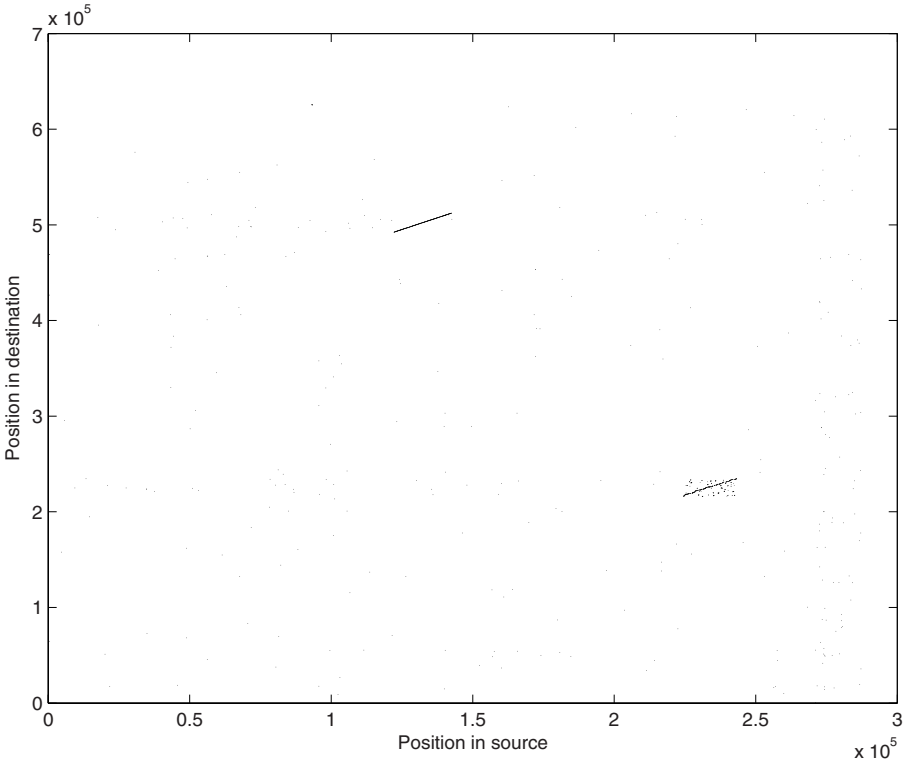


Fig. 1. Clean encoplot example, as used for plagiarism detection. Here the source #2400 and the destination #2 from the corpus, each dot is a 16-bytes n-gram that is shared by the two texts. Two copied passages can be observed as more or less clean local diagonal formations of dots.

For short n-grams, the probabilities for the n-grams to appear multiple times in each document increase. For medium-size n-grams (not very short, but not very long either – n=8 bytes in the herein reported experiment) there is more probably to have multiple instances of the n-grams in the passage in the remaining text of the source document than in the remaining text of the destination document – which is the same with saying that the n-grams distribution in the copied passage matches more the one of the source text than the one of the destination one. What happens when one n-gram from the source document appears not only in the copied passage but also before and after it in the source document? Assuming for simplicity that it only appears once (in the copied passage) in the plagiarizing document, then only one match will be in the encoplot, between the first instance of that n-gram in the source (appearing before the plagiated passage) and the single instance in the destination document, in the copied passage. Effectively this means that the dot which corresponds to that match is moved forward towards the beginning of the source document. For every

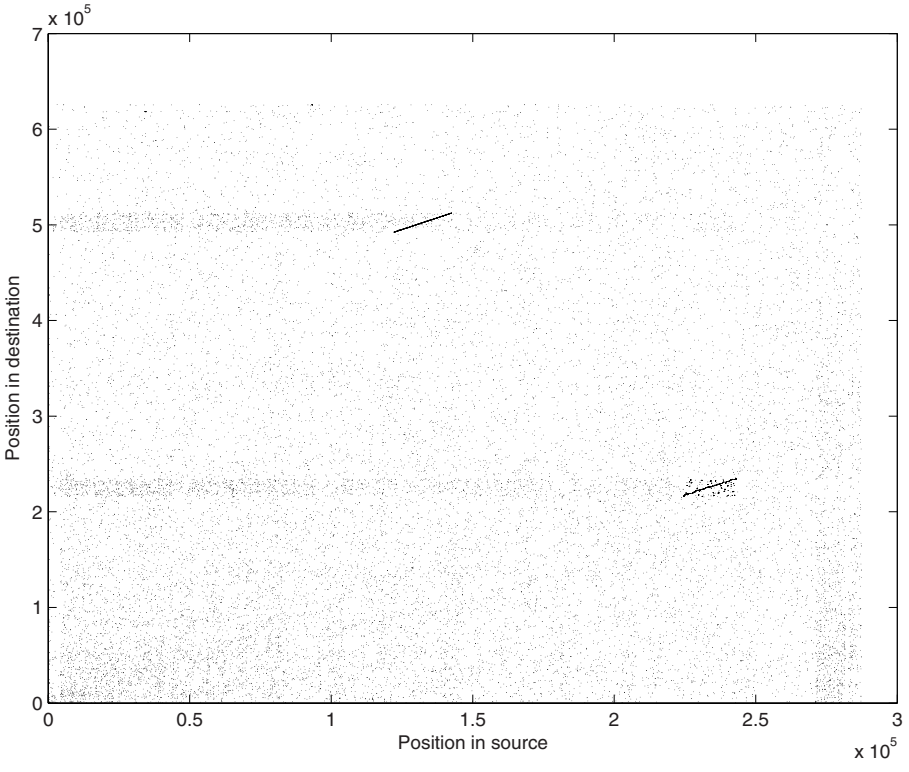


Fig. 2. Asymmetry in Encoplot. The same pair of documents (source #2400 and destination #2) from the corpus, each dot is an 8-bytes n-gram that is shared by the two texts. The shorter n-grams lead to more coincidences, "clouds" of dots that are unwanted for plagiarism detection, but useful for determining the direction of the plagiarism.

n-gram this offset can be different and the result is a cloud of dots moved from the diagonal towards the beginning of the source document. Of course for some n-grams the opposite could be true, to be unique in the source and have instead multiple instances in the destination, including one before the passage, which will have the effect to displace the corresponding dots from the diagonal and move those towards the beginning of the destination document. It is just that we expect this to happen less often that the former case.

We set to test how accurate a method based on this observation would be. Finding the passages in correspondence is the problem of external plagiarism detection, and as it is not our concern now, we assume we have the full details about what passages in what text correspond to what passages in what other text and the only information missing is which is the direction. To model this, we randomly permute the source and the "destination" and try to detect the correct direction using solely the asymmetry of the encoplot.

2.3 Measuring and Using the Asymmetry

Spotting the asymmetry in the encoplot graph is easy for humans. In order to do it automatically, one needs to solve a computer vision problem. The difficulty lays in the size of the encoplot data, that can be as long as one of the texts. This is still much better than the maximum length of the general dotplot sets, as those can extend up to the product of the lengths of the two documents.

Figure 3 shows the regions used for our scoring.

We have modeled the visual contrast between the horizontal trace and it's neighbor regions as the mean of the contrast to the upper band and the contrast to the lower band, each of those of the same width as the trace. The width is sometimes limited, when the trace is too close to the beginning or to the end of the vertical axis.

$$hcontrast = \frac{contrast\_up + contrast\_down}{2}. \tag{1}$$

The contrast between a trace and a neighbor region is measured through the percentage of the points contained in their union that are contained in the trace,

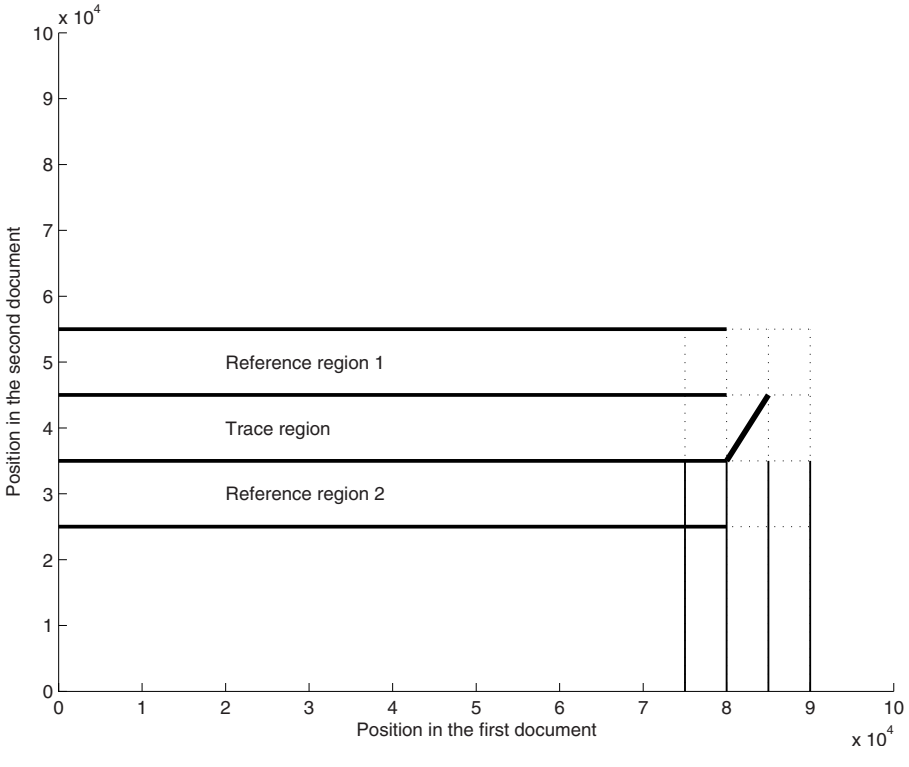


Fig. 3. Regions used to define the scoring. The density of the dots in the considered trace region (either horizontal or vertical) is compared to the density of the dots in the neighboring reference regions.

compensated for the truncation of the neighbor region (needed when the width of the neighbor region is limited by the beginning or the end of the corresponding document).

$$contrast\_up = \frac{\|Traceregion\|}{\|Traceregion\| + \|Region1\| * width_{Trace}/width_{Region1}}. \quad (2)$$

The contrast *vcontrast* between the vertical trace and the neighbor region is defined similarly, but uses the left and right neighbor regions instead of above and below ones.

We then classify each pair according to this heuristic: the higher contrast trace points to the copy and is parallel to the source.

$$encoplot\_block\_asymmetry\_indicator = hcontrast - vcontrast. \quad (3)$$

When the asymmetry indicator is positive, we predict that the source is the document on the horizontal axis, otherwise that it is the document on the vertical axis. We count the cases when the asymmetry indicator is zero as prediction errors (even though half of them could randomly match the true answer).

3 Results and Analysis

The results are summarized in Table 1.

Table 1. Results

Population layer (and proportion)	Prediction accuracy	p-Value for 1 dof χ^2
Whole population (100%)	75.417%	–
Translated (8.68%)	74.361%	0.0502
Not obfuscated (45.21%)	77.852%	$< 10^{-24}$
High obfuscation (18.58%)	69.776%	$< 10^{-52}$
Short passages (26.18%)	68.111%	$< 10^{-121}$
Long passages (73.82%)	78.008%	$< 10^{-43}$
Close to the source start (14.63%)	69.606%	$< 10^{-43}$

The global accuracy (75.417%) is surprisingly good.

It is interesting to see in what cases the method fails and why. The influence of the factors is given in Table 1, together with their statistical significance, computed using a single degree of freedom χ^2 test with the null hypothesis that the factor has no influence on the decision accuracy.

A visual inspection of those cases where the method fails show that they can be classified into one of these classes: too short passages (therefore too few n-grams expected in the asymmetric parasitic clouds of dots); passages too close to the beginning of one of the texts (therefore again too few dots in one of

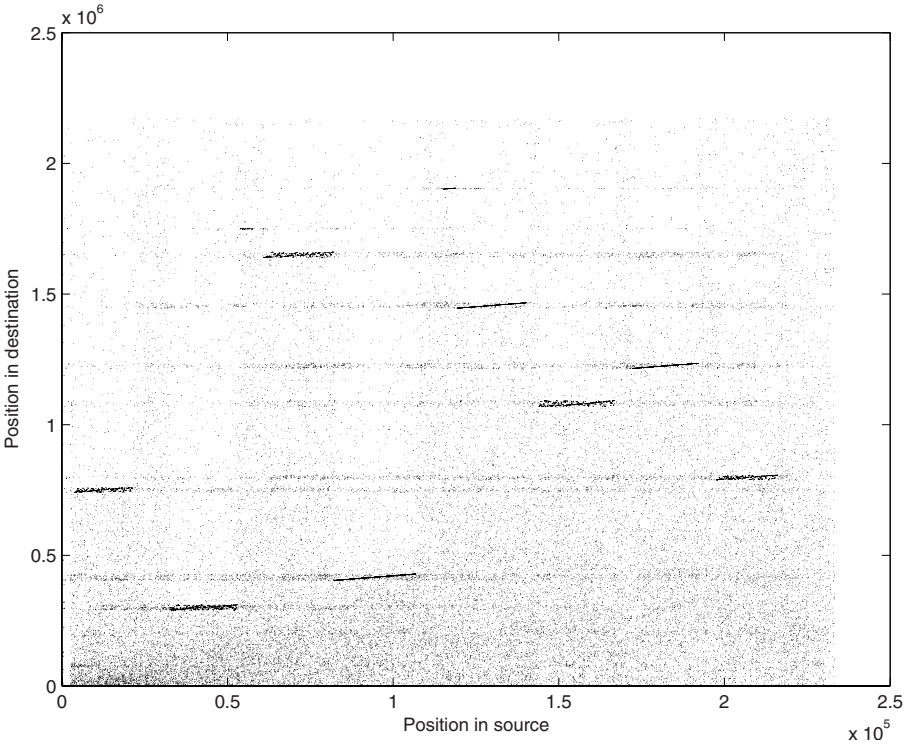


Fig. 4. Crowded encoplot, through many passages plagiarized from the same source; the horizontal traces selectively affect the density of the tested vertical bands. Here displayed source document #2225 versus suspicious document #5.

the clouds); too crowded encoplots (as in Figure 4), with many closely situated passages in correspondence (decreasing thus the contrast of the trace/cloud of interest); too short texts (and again too small dot sets and too high variances of their size).

4 Discussion and Conclusion

We have to state as clearly as possible that we don't claim that we are ready to close any priority/plagiarism dispute simply by presenting the texts to our method. We are very much aware that plagiarism as blatant as many of the instances in the used corpus is maybe never to be seen in practice. This could also be said about so much lack of blending of the copied passages into the destination. All these aspects made a problem – that could be impossible to solve in real cases – solvable in 75% of the instances in this artificial plagiarism corpus.

Why does it work and what else could work? To understand that, we have to consider the meaning of those dots that help us to get back the plagiarism

direction information eventually. They are shared n-grams between the two texts. Their preferential spread / higher density on a direction parallel with the axis of the source document corresponds to a better blending of that passage into the source document than into the copying document. One could say that our asymmetry indicator turned encoplot into a method for intrinsic plagiarism detection, to some extent. One could expect that other methods of intrinsic plagiarism detection can be turned into methods to determine automatically the direction of the plagiarism.

Admittedly we didn't spend too much time with tuning the asymmetry indicator or any of the other parameters. We have tuned the indicator on the first 100 plagiarism cases in a visual data exploration fashion, then we validated it by running on the remaining 73422 cases. It worked as good as it did from the first run, in the first day. Our point was mostly to have a proof of concept that this open problem of the automatic plagiarism detection, the detection of the direction of plagiarism, is solvable at least in many instances of the simulated/artificial plagiarism.

Can the performance on the population layers where the method fails more often be improved? For some of them probably yes, and here is how this could be done: for the texts too close to the beginning of the source, the encoplot can be computed on the mirrored texts. Please note that it is not enough to look for the clouds towards the end of the documents, as the encoplot procedure produces different clouds when computed on the mirrrored texts, and, as explained before, due to the way encoplot matches the texts one should only be interested in the dots displaced towards the beginnings of the texts given as input to encoplot. For the too crowded encoplots (like in Figure 4), the encoplot could be computed repeatedly for each plagiarized passage in turn, overwriting all other passages in correspondence with random text.

We found surprising that the encoplot asymmetry indicator worked so accurate on the translated passages. The encoplot for a pair of documents where the plagiarizing involved translation from Spanish to English is shown in Figure 5. In this case, the automatic translation used left untranslated all person names – as expected – and some spanish words. This was enough for the text to blend better into the original spanish context than into the english context, despite being almost in English after translation.

Following the best practice in science, our results are fully reproducible, as both encoplot and the data used are publicly available. The corpus is available as a web resource [15] and the code for computing encoplot of two files is available in the encoplot paper [9].

To conclude, we have shown that on the largest plagiarism corpus available to date (albeit artificial) the problem of detecting the direction of the plagiarism is solvable with a fairly high accuracy (about 75%). Future work will show how well this method works on natural plagiarism. We are not aware of any publicly available corpus (even of much smaller size) that would have allowed us to test this. We are looking forward to seeing more papers on this subject, more results on the same public corpus, maybe leveraging the intrinsic plagiarism detection methods.

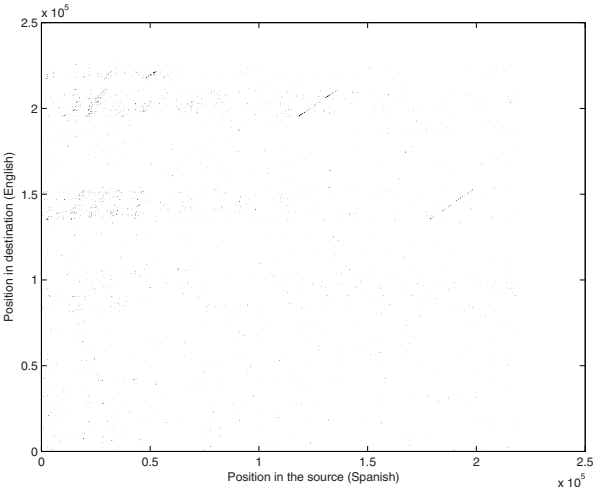


Fig. 5. Plagiarization with translation – the direction is still detectable, although the dot clouds are not very clearly delimited and not very dense. Here displayed source document #2923 (Spanish) versus suspicious document #90 (English).

Acknowledgments. The authors thank Dr. Andreas Ziehe and the anonymous reviewers for the thorough review and their useful suggestions. The contribution of Filip Grozea to the development of the heuristics is also acknowledged.

References

1. Project Gutenberg (1971), http://www.gutenberg.org
2. Baker, R.K., Thornton, B., Adams, M.: An Evaluation of The Effectiveness of Turnitin. Com As A Tool For Reducing Plagiaris in Graduate Student Term Papers. College Teaching Methods & Styles Journal 4(9) (2008)
3. Basile, C., Cristadoro, G., Benedetto, D., Caglioti, E., Degli Esposti, M.: A plagiarism detection procedure in three steps: selection, matches and squares. In: 3rd Pan Workshop. Uncovering Plagiarism, Authorship and Social Software Misuse, p. 19
4. Clough, P.: Old and new challenges in automatic plagiarism detection. National Plagiarism Advisory Service (2003)
5. Corry, L., Renn, J., Stachel, J.: Belated decision in the Hilbert-Einstein priority dispute. Science 278(5341), 1270 (1997)
6. Errami, M., Hicks, J.M., Fisher, W., Trusty, D., Wren, J.D., Long, T.C., Garner, H.R.: Deja vu A study of duplicate citations in Medline. Bioinformatics 24(2), 243 (2008)
7. Freire, M., Cebrian, M.: Design of the AC Academic Plagiarism Detection System. Technical report, Tech. rep., Escuela Politecnica Superior, Universidad Autonoma de Madrid, Madrid, Spain (November 2008)
8. Grozea, C.: Plagiarism detection with state of the art compression programs. Report CDMTCS-247, Centre for Discrete Mathematics and Theoretical Computer Science, University of Auckland, Auckland, New Zealand (August 2004)

9. Grozea, C., Gehl, C., Popescu, M.: ENCOPLOT: Pairwise Sequence Matching in Linear Time Applied to Plagiarism Detection. In: 3rd Pan Workshop. Uncovering Plagiarism, Authorship and Social Software Misuse, p. 10
10. Potthast, M., Stein, B., Eiselt, A., Barrón-Cedeño, A., Rosso, P.: Overview of the 1st International Competition on Plagiarism Detection. In: 3rd Pan Workshop. Uncovering Plagiarism, Authorship and Social Software Misuse, p. 1
11. Roos, T., Heikkila, T.: Evaluating methods for computer-assisted stemmatology using artificial benchmark data sets. Literary and Linguistic Computing 24(4) (2009)
12. Ryu, C.K., Kim, H.J., Cho, H.G.: A detecting and tracing algorithm for unauthorized internet-news plagiarism using spatio-temporal document evolution model. In: Proceedings of the 2009 ACM symposium on Applied Computing, pp. 863–868. ACM, New York (2009)
13. Ryu, C.K., Kim, H.J., Ji, S.H., Woo, G., Cho, H.G.: Detecting and tracing plagiarized documents by reconstruction plagiarism-evolution tree. In: CIT, p. 119 (2008)
14. Sauer, T.: Einstein Equations and Hilbert Action: What is missing on page 8 of the proofs for Hilbert's First Communication on the Foundations of Physics?. Archive for history of exact sciences 59(6), 577–590 (2005)
15. Webis at Bauhaus-Universität Weimar and NLEL at Universidad Politécnica de Valencia. In: Potthast, M., Eiselt, A., Stein, B., Cedeño, A.B., Rosso, P. (eds.) PAN Plagiarism Corpus PAN-PC 2009 (2009), http://www.webis.de/research/corpora

Integer Linear Programming for Dutch Sentence Compression

Jan De Belder and Marie-Francine Moens

Katholieke Universiteit Leuven,
Department of Computer Science,
Celestijnenlaan 200A, B-3001 Heverlee, Belgium
jan.debelder@cs.kuleuven.be, sien.moens@cs.kuleuven.be

Abstract. Sentence compression is a valuable task in the framework of text summarization. In this paper we compress sentences from news articles from Dutch and Flemish newspapers written in Dutch using an integer linear programming approach. We rely on the Alpino parser available for Dutch and on the Latent Words Language Model. We demonstrate that the integer linear programming approach yields good results for compressing Dutch sentences, despite the large freedom in word order.

1 Introduction

Since the end of the 20th century, the compression of texts has been an active research topic in natural language processing (see for example the Document Understanding Conferences [1], and the more recent Text Analysis Conferences [2]). As this is a very difficult problem, it has often been reduced to the summarization of individual sentences, commonly referred to as sentence reduction [3] or sentence compression. This summarization task is the easiest in a word deletion setting, where we remove words from the original sentence, while maintaining a grammatical and coherent sentence that conveys the most important information [4]. For the Dutch language, the research in this area is limited. There has been some work on the summarization of documents in [5]. The compression of individual sentences has only been approached from a subtitle generation viewpoint [6] [7] [8] and a headline generation viewpoint [9]. In this paper, we investigate a generic method for sentence reduction, based on integer linear programming [10]. Required for this method are a language model, a parser, and a integer linear programming (ILP) solver.

The ILP approach operates by viewing sentence compression explicitly as an optimization problem. With a binary decision variable for each word in the original sentence, indicating whether or not it should be in the compressed sentence, the ILP solver finds an assignment for these variables that maximizes the probability of the sentence in the language model. In order to create well-formed summary sentences, the compression model might incorporate additional constraints that use grammatical rules of the language. As the most interesting information is most likely not very prominent in the language model, there is

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 711–723, 2010.

also need for a way of incorporating this information in the compressions. This is the function of the significance model.

In the next section we give an overview of relevant background work. Section 3 shortly introduces the tools we used for Dutch. Section 4 describes the main ideas of the integer linear programming approach. Our experimental setup can be found in section 5, and section 6 reports on the results. Finally, we give our conclusions and indications for future work in section 7.

2 Background

Summarization or compression of text is a useful, but non-trivial application of natural language processing. Currently, there are several settings being researched that include the summarization of single documents [11], the summarization of multiple documents [12], and the summarization of single sentences. In this paper, we address the last setting.

Nearly all approaches of sentence compression rely on word deletion in such a way that the result is still a grammatical sentence, and conveys the most important information of the original sentence. A common application is headline generation based on the content of a larger text. By looking for headlines that are a subsequence of words in the first sentence of a news article, a sentence compression corpus can automatically be constructed. The offset for this approach was given in [4]. These authors used a parallel corpus of compressed and original sentences based on the Ziff-Davis corpus of news articles in the computer technology domain. The authors evaluated two compression methods. A noisy channel model considers an original sentence as the compressed sentence to which noise has been added. It assigns the most likely compression to the full sentence using Bayes rule, where the probability of a noisy component given a summary sentence is learned from the training data. The decision based model learns the discriminative reductions of the parse tree with a decision-tree learner based on the training data. The noisy-channel model is, however, not directly applicable for Dutch, due to lack of a Probabilistic Context Free Grammar. The decision based model has the disadvantage that the desired amount of compression cannot be given as a parameter.

In [9], headline generation was studied for the Dutch language. The method takes inspiration from the linguistically motivated Hegde trimmer algorithm [13], which employs rules to reduce the parse tree of a sentence, but learns the rules automatically using Transformation Based Learning, an error-driven approach for learning an ordered set of rules. The corpus that was used originates from Dutch news articles with matched headlines, taken from the Twente News Corpus.

Another setting in the compression of single sentences is the generation of subtitles for broadcasts. This is the case that has been mostly studied for Dutch [6] [7] [8]. These methods are based on shallow parsing and most of them require a parallel corpus for training. However, recent work [14] has shown that a word deletion approach is not very suited for subtitle generation.

There are also a few unsupervised approaches for sentence compression. [15] summarize the transcription of a spoken sentence, given a fixed compression rate.

They use dynamic programming to find an optimal scoring solution, that takes a language model and the confidence of the speech recognizer into account. [16] define a semi-supervised and unsupervised version of the noisy channel model of [4]. [10] use an integer linear programming approach, which is applicable for any language, given the availability of a parser. This is the method that we will discuss, use, and modify in the remainder of this paper.

3 Language Tools

In this section we describe the tools we used for constructing our Dutch sentence compression system.

3.1 Parsing

For parsing the Dutch sentences, we use the Alpino parser [17]. The Alpino system is a linguistically motivated, wide-coverage grammar and parser for Dutch in the tradition of HPSG. It consists of about 800 grammar rules and a large lexicon of over 300,000 lexemes and various rules to recognize special constructs such as named entities, temporal expressions, etc. The aim of Alpino is to provide computational analysis of Dutch with coverage and accuracy comparable to state-of-the-art parsers for English. It is freely available for download.[1]

3.2 Latent Words Language Model

The Latent Words Language Model (LWLM) models the contextual meaning of words in natural language as latent variables in a Bayesian network [18]. In a training phase the model learns for every word a probabilistic set of synonyms and related words (i.e. the latent words) from a large, unlabeled training corpus. During the inference phase the model is applied to a previously unseen text and estimates for every word the synonyms for this word that are relevant in this particular context. The latent words help to solve the sparsity problem encountered with traditional n-gram models, leading to a higher quality language model, in terms of perplexity reduction on previously unseen texts [19]. In this article the model is trained on a 25m token corpus, consisting of Dutch newspaper articles.

4 Integer Linear Programming Approach to Sentence Compression

In this section we will lay out the sentence compression method based on integer linear programming, following the line of work in [10]. We will start by shortly explaining what integer programming is, and how the basic method works by maximizing a language model probability. There are also extra constraints needed to make sure that a meaningful and grammatical sentence is obtained. In section 4.4 we discuss the significance model, that ensures that the generated compressions also contain topics of interest.

[1] http://www.let.rug.nl/vannoord/alp/Alpino/

4.1 Integer Linear Programming

Integer linear programming is a restricted case of linear programming, where the values of the variables are limited to be only integers, instead of any real number. Linear programming tries to maximize (or minimize) an objective function, by searching for optimal values for the variables that constitute the objective function. This objective function is a linear combination of these variables, hence the name. The finding of an optimal combination of values is usually constrained. These constraints ensure that the variables cannot be infinitely large, and that the value of one variable can influence the other variables.

Integer programming has been used often in Natural Language Processing, for many different tasks. In many situations, NLP constitutes searching in very large hypothesis spaces, like packed forests of parse trees [20]. Other applications include a.o. coreference resolution [21] and semantic role labeling [22]. Integer linear programming, a technique that has often been used in optimalisation theory for many decades, is very well suited for these kind of problems, as it enables us to efficiently search for the optimal solution, and at the same time incorporate constraints on a global scale.

4.2 Integer Programming for Sentence Compression

Given a sentence $W = w_1 \dots w_n$, our goal is to obtain a sentence W^*, with a reduced number of words. For a sentence $W = w_1 \dots w_n$, we first need decision variables to indicate whether or not w_i should be in the compressed sentence. We notate these variables with y_i, with a value of 1 if word w_i is in the compressed sentence, and 0 if it is not. For clarity, suppose we want the ILP solver to find a sentence that maximizes a unigram model, then the objective function would look like this:

$$max\ z = \sum_{i=1}^{n} y_i P(w_i),$$

with $P(w_i)$ being the unigram probabilities. This overly simple model is not adequate; a trigram model would have much better performance. This comes down to adding three additional types of variables. In short, we need n extra variables to indicate whether or not a word starts the sentence (p_i), and $\frac{n \cdot (n-1)}{2}$ decision variables that indicate whether two words end the sentence (q_{ij}). Finally, there are $\frac{n \cdot (n-1) \cdot (n-2)}{6}$ variables needed to indicate whether a specific trigram $w_i w_j w_k$ is in the sentence (x_{ijk}). These three types of variables are needed for constraints on the language model. For example, only one word can start the sentence, which translates to a constraint in the ILP model. Without these constraints, the ILP would set all variables to 1, and say that all words start the sentence. The complete list of constraints can be found in [10], but will not be repeated due to spatial constraints, and the fact that they are not required to understand to operations behind the method. The objective function of the integer linear programming problem is given in the following equation[2]:

[2] From here on we assume all probabilities are log-transformed.

$$max\ z = \sum_{i=1}^{n} p_i P(w_i|\text{start}) + \sum_{i=1}^{n-2} \sum_{j=i+1}^{n-1} \sum_{k=j+1}^{n} x_{ijk} P(w_k|w_i w_j) + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} q_{ij} P(\text{end}|w_i w_j) \quad (1)$$

4.3 Linguistic Constraints

The ILP model given above is language independent. However, the fact that a sentence has a high probability in a language model, does not make it a grammatical and fluent sentence. That is why there is the need to incorporate language specific grammatical information in the method.

The constraints described below are motivated from a linguistic and intuitive point of view and are often dependent on the language used. These constraints are based on a parse tree and the grammatical relations of a sentence, and can be used in combination with any parser. In [10], the Robust Accurate Statistical Parsing toolkit was used [23]. As described in section 3.1, for Dutch we are limited to the use of Alpino.

Modifier Constraints. It is often the case that determiners can be left out of the compression (especially in the case of headline generation). This still yields a grammatical sentence. The other way around, i.e. keeping the determiner but removing its head word, is not acceptable. This leads to the following constraint:

$$y_i - y_j \geq 0 \quad (2)$$
$$\forall i, j : w_j \in w_i\text{'s determiners}$$

If a determiner w_j is in the compression, which corresponds to y_j having the value 1, the constraints force y_i to take the value 1 as well, causing the head word w_i to be in the compression.

Some determiners cannot be left out, especially when they change the meaning of their head word, and thus probably the meaning of the entire sentence. The most trivial one is the word 'not'. We also included the word 'none'. An important modifier for Dutch is the word *er*, which translates roughly as 'there'[3]. This constraint can be removed, but it generates more fluent sentences, rather than headline-style sentences. Possessive modifiers are also added to the list.

$$y_i - y_j = 0 \quad (3)$$
$$\forall i, j : w_j \in w_i\text{'s determiners} \wedge$$
$$w_j \in (\text{not, none, possessives, '}er\text{'})$$

[3] For example in the sentence '*Er is melk in de koelkast*', which translates to 'There is milk in the fridge'. Sometimes this is not as clear. The sentence 'Something has to be done' translates to '*Er moet (has) iets (something) gedaan (done) worden (to be)*'.

Note that the difference between constraints 2 and 3 is in the sign of the equation: constraint 2 uses a $\geq$ sign to indicate that w_i can be in the compression by itself, but w_j can not. Constraint 3 uses an $=$ sign, which mean that either both w_i and w_j have to be in the compression or either none of them can be in the compression.

Argument Structure Constraints. The next constraints are needed for the overall sentence structure. Constraint 4 makes sure that if there is a verb in the compressed sentence, then so must be its arguments. The reverse also has to be true: if there is a subject from the original sentence taken for the compressed sentence, so must be the corresponding verb.

$$y_i - y_j = 0 \tag{4}$$
$$\forall i, j : w_j \in \text{subject/object of verb } w_i$$
$$\sum_{i:w_i \in verbs} y_i \geq 1 \tag{5}$$

Constraint 5 requires that, if there is a verb in the original sentence, there should also be at least one in the compressed sentence.

One of the peculiarities of Dutch[4] are separable verbs that fall apart into their original parts, when you conjugate them. For example, *toepassen* (to apply), becomes in the first person singular *ik pas toe* (I apply). If a compressed sentence contains the stem of the separable verb, it should also include the separated part, and vice versa. The parser detects these separable verbs, so we can define the following constraint:

$$y_i - y_j = 0 \tag{6}$$
$$\forall i, j : w_j = \text{separated part of separable verb } w_i$$

Furthermore we also require the predicative adjectives to be included together with their head, and the same for reflexive objects such as 'themselves'.

There are two other constraints needed for prepositional phrases and subordinate clauses in order to ensure that the introducing term is included, if any word from the phrase or clause are included (defined in equation 7). Subordinate clauses are those that begin with a wh-word, or with subordinating conjunctions such as 'after' or 'because'. The reverse should also hold (see equation 8).

$$y_i - y_j \geq 0 \tag{7}$$
$$\forall i, j : w_j \in \text{PP/SUB} \wedge w_i \quad \text{starts PP/SUB}$$
$$\sum_{i:w_i \in \text{PP/SUB}} y_i - y_j \geq 0 \tag{8}$$
$$\forall j : w_j \text{ starts PP/SUB}$$

[4] This is also common in German and Hungarian.

General Constraints. Alpino is able to detect multi word units (MWUs). These can be names of persons, such as *Minister Van Der Donck*, but also parts of expressions, such as *op wacht staan* (to stand guard). For simplicity we define a constraint that either all words of the MWU should be included, or none of them.

Related to the compression length, it is possible to define an upper and lower bound on the generated compression. Enforcing a length of at least l tokens is done with the following constraint:

$$\sum_{i=1}^{n} y_i \geq l \tag{9}$$

Defining an upper bound can easily be done by replacing the $\geq$ sign with $\leq$.

4.4 Significance Model

A probable side effect of relying on a language model to generate compressions, is that the model will prefer known words. This has as a consequence that the most important words in the sentence, for example names of persons, will not be likely to appear in the compression. The solution for this problem lies in a significance model. This model assigns a weight to every topic word in the sentence, with a topic word being a noun or a verb. The weights are based on several statistics, and calculated with the following equation:

$$I(w_i) = \frac{l}{N} f_i log \frac{F_a}{F_i} \tag{10}$$

where f_i and F_i are the frequencies of word w_i in the document and a large corpus respectively, F_a the sum of all topic words in the corpus. l is based on the level of embedding of w_i: it is the number of clause constituents above w_i, with N being the deepest level in the sentence. To incorporate these weights in the objective function given by equation 1, the sum of equation 10 over the topic words can be simply added, resulting in the following equation:

$$\begin{aligned} max\ z = \lambda \sum_{i=1}^{n} y_i I(w_i) + \sum_{i=1}^{n} p_i P(w_i|\text{start}) \\ + \sum_{i=1}^{n-2} \sum_{j=i+1}^{n-1} \sum_{k=j+1}^{n} x_{ijk} P(w_k|w_i w_j) \\ + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} q_{ij} P(\text{end}|w_i w_j) \end{aligned} \tag{11}$$

The parameter λ weighs the importance of the language model versus the significance model, and can be estimated on a small set of training data.

5 Evaluation

5.1 Data

The data consists of news articles written in Dutch, coming from major Belgian and Dutch newspapers and crawled from the Web pages of the news providers. We selected the websites and articles at random, to have a diverse set of texts. The articles date back to the beginning of 2008. We used a set of articles from 31/1/2008 and 1/2/2008 for development and training, and articles from 6/2/2008 and 8/2/2008 for the evaluation.[5] We manually segmented the articles into sentences, to ensure a clean dataset. The training and development data consisted of 40 articles, the evaluation data of 30.

Since the evaluation is done manually, as will be described in section 5.3, the amount of sentences that we can evaluate is limited. Here we took the first sentence of each article in the evaluation set, and limited these further to sentences that contain at least 15 tokens. This resulted in a set of 21 sentences, with an average length of 20.8 tokens, ranging over a diverse set of topics.

We used a different data set to train the Latent Words Language model. We took a 25 million token subset of the Twente News Corpus [24], from four different newspapers in the year 2005. The dictionary size was limited to 65.000 words. We also used this data to estimate the corpus frequency of the topic words, as described in equation 10. If a topic word was not present in the corpus, we estimated its weight as the average of the other topic words in the sentence.

5.2 Systems

For the evaluation we tested the system in four different settings, all based on the integer linear programming approach. The first system relies solely on the language model, and does not use any grammatical information. The second system does use the grammatical constraints. The third and fourth system both add the significance model, but with different values for the parameter λ. As described in section 4.4, this parameter weighs the importance of the significance model against the language model. During initial testing it became clear that it is very difficult to estimate this parameter. Values that work for some sentences yield lesser results on other sentences. It also has a significant influence on the length of the compression, where higher values for λ tend to generate longer sentences. Higher values cause the system to only include the topic words, while still being limited by the constraints, which results in using all the topic words without everything that is dictated by the constraints. For these reasons, we did the evaluation with two different values for λ: 0.75 and 1.5, that both had good empirical results on the development data.

Finally, we constrained the systems to generate compressions of at least 40% of the original length, by using the constraint in equation 9.

[5] This difference in time was needed to ensure that no articles in the evaluation data overlapped with those in the development data.

5.3 Evaluation

As is good practice in the testing of summarization systems, we opted for manual evaluation. We did two different experiments. In the first experiment, we presented the participants with a list of generated compressions, each from a different original sentence. We asked the participants to give a score for the grammaticality of each sentence, on a five point scale. In the second experiment the participants were given the original sentences together with the corresponding compressions, and they were asked to rate the compressions based on the retention of the most important information, again on a five point scale. The sets of sentences were generated at random: each set contained compressions from the different systems. Together with the four systems defined above, we added a manually constructed set of compressions made by one of the authors. The participants were told that all the sentences were machine generated. This allows us to compare the machine generated compressions with one made by a human, and define an upper bound on the performance that is achievable in a word-deletion setting. In total we had 15 participants, each grading 21 sentences based on grammaticality, and another 21 sentences on content.

Using the manually constructed set of compressions, we also calculated the ROUGE scores [25], as often applied in the DUC competitions. We used the ROUGE-2, ROUGE-L, and ROUGE-SU4 metrics, that assign scores based on bigram co-occurrences, the longest common subsequence, and skip-bigrams in combination with unigrams respectively.

6 Results

6.1 Human Evaluation

Grammaticality. The results on the manual evaluation can be found in table 1. From the column that reports on the grammaticality of compressions, it is clear that the grammatical constraints are necessary. The system that uses only the language model to generate compressions, did not came up with many meaningful sentences. This is very likely due to the limited size of the language model used. The systems that do use the grammatical constraints usually come up with a grammatical compression. The median of grammaticality scores is 4, for each of the three systems that used the grammatical constraints. Annotators often punished the compressions due to not incorporating the determiners, which generates more headline-like compressions. The leaving out of commas was also a cause for lower ratings. In one case none of the systems was able to include the main verb and subject, which did not happen when using a longer minimum compression length. The biggest problem is the needed inversion of a verb and a subject when a prepositional phrase is removed from the beginning of the sentence. Switching the verb and the subject in a sentence would require substantial modifications to the ILP method. The grammatical information from the parse tree would not just lead to the adding of more constraints, but to the addition of more decision variables and a modification of the objective function, which we leave for further research.

Table 1. Manual evaluation results of the four systems and the handcrafted summaries, on grammaticality and information retention of the compressions

System	Avg. Comp. Rate[6]	Grammar	Information
Human	66.9%	4.71 ± 0.40	4.43 ± 0.53
LWLM	43.0%	1.29 ± 0.54	1.26 ± 0.30
LWLM+Gram	43.3%	3.45 ± 1.47	3.14 ± 1.31
LWLM+Gram+Sig ($\lambda = .75$)	49.0%	3.81 ± 1.38	3.19 ± 1.67
LWLM+Gram+Sig ($\lambda = 1.5$)	57.5%	3.98 ± 1.12	3.41 ± 1.19

Significance Model. Looking further we can see that the significance model has an impact on the information retention, although this is rather limited. Despite the fact that the last system ($\lambda = 1.5$) generates on average sentences that are 14% longer, this has little influence on the scores given by the participants of the experiment. The reason for this is that the most important information usually takes the role of subject or object, and is thus already required to be in the compression. The difference in score between the best system and the human made compressions is larger than for the grammaticality, but it should be noted that the human made compressions are on average almost 10% longer.

6.2 Automatic Evaluation

From the results in table 2 we can conclude that the automatic evaluation measures all follow the human judgment. The version of the system with the significance model ($\lambda = 1.5$) scores the best, which indicates that this model generates compressed sentences that are the closest to the handcrafted summaries.

6.3 Discussion

In general, the method performs rather well. When compared to the human made summaries, the best model only scores ±1 point lower, both on grammaticality and content. We also tested whether the human made summaries were possible to create by the ILP method, using the grammatical constraints imposed. In 12 out of the 21 cases, this was not possible. Often the cause was a small error in the parsing process, especially in the case of PP-attachments.

Another related problem can be found in the compression of names. Often these are accompanied by a description of their function, for example 'The French president Sarkozy'. Without loss of information, this can easily be reduced to 'Sarkozy'. But when talking about the Serbian president Boris Tadić, the participants of the experiments preferred the descriptive compression 'the Serbian president' over the actual name 'Boris Tadić'. This problem is not only present in Dutch, but in summarization in general.

In these experiments we defined specific values for λ and a specific lower bound on the sentence length, in order to obtain just one compression from every

[6] We define the average compressed rate as the average percentage of words retained in the compression.

Table 2. Automatic evaluation results with the ROUGE toolkit, using the handcrafted summaries as a gold standard

System	ROUGE-2	ROUGE-L	ROUGE-SU4
LWLM	0.240	0.569	0.341
LWLM+Gram	0.431	0.650	0.469
LWLM+Gram+Sig ($\lambda = .75$)	0.472	0.697	0.505
LWLM+Gram+Sig ($\lambda = 1.5$)	0.508	0.712	0.530

system. However, the systems can easily generate an entire set of compressions by varying the parameters, more often than not generating better compressions than given here. As the solving of the ILP problem is several orders of magnitude faster than parsing the sentence with the Alpino parser, it is our opinion that the determination of the best compression, given a set of possible compressions, can better be handled in a later stage.

7 Conclusion

In this paper we have presented a sentence compression method for Dutch, a free word order language. We used an integer linear programming approach that finds a compression by maximizing the language model probability, while constrained to be grammatical. For this we used Alpino, a parser for Dutch, and the Latent Words Language Model. We needed extra language-specific constraints on the generated compressions to maintain the meaning, which we accomplished by using the output of the parser. We also identified some shortcomings, by checking whether the handcrafted compressions can be generated under the grammatical constraints, which was not always the case.

The next step is to extend the integer linear programming approach to allow for words to swap places, allowing the model to generate more grammatical compressions. We also believe that the meaningful compression of person names with their description could be learned from training data, in addition to this otherwise unsupervised method.

Acknowledgments

This research is funded by the Dutch-Flemish NTU/STEVIN project *DAISY*[7] (ST 07 015) and the EU project *PuppyIR*[8] (EU FP7 231507).

References

1. Nenkova, A.: Automatic text summarization of newswire: Lessons learned from the document understanding conference. In: Proceedings of the National Conference on Artificial Intelligence, vol. 20, p. 1436. MIT Press, Cambridge (2005)

[7] http://www.cs.kuleuven.be/ liir/projects/daisy/
[8] http://www.puppyir.eu

2. Text Analysis Conference (TAC), http://www.nist.gov/tac/
3. Jing, H.: Sentence reduction for automatic text summarization. In: Proceedings of the 6th Applied Natural Language Processing Conference, pp. 310–315 (2000)
4. Knight, K., Marcu, D.: Statistics-based summarization-step one: Sentence compression. In: Proceedings of the National Conference on Artificial Intelligence, pp. 703–710. MIT Press, Cambridge (2000)
5. Angheluta, R., De Busser, R., Moens, M.F.: The use of topic segmentation for automatic summarization. In: Proceedings of the ACL 2002 Workshop on Automatic Summarization, Citeseer (2002)
6. Vandeghinste, V., Pan, Y.: Sentence compression for automated subtitling: A hybrid approach. In: Proceedings of the ACL Workshop on Text Summarization, pp. 89–95 (2004)
7. Vandeghinste, V., Sang, E.: Using a parallel transcript/subtitle corpus for sentence compression. In: Proceedings of LREC 2004, Citeseer (2004)
8. Daelemans, W., Hothker, A., Sang, E.: Automatic sentence simplification for subtitling in Dutch and English. In: Proceedings of the 4th International Conference on Language Resources and Evaluation, Citeseer, pp. 1045–1048 (2004)
9. de Kok, D.: Headline generation for Dutch newspaper articles through transformation-based learning. Master's thesis
10. Clarke, J., Lapata, M.: Global inference for sentence compression: An integer linear programming approach. Journal of Artificial Intelligence Research 31(1), 399–429 (2008)
11. Teng, Z., Liu, Y., Ren, F., Tsuchiya, S.: Single document summarization based on local topic identification and word frequency. In: Gelbukh, A., Morales, E.F. (eds.) MICAI 2008. LNCS (LNAI), vol. 5317, pp. 37–41. Springer, Heidelberg (2008)
12. 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, Suntec, Singapore, August 2009, pp. 297–300. Association for Computational Linguistics (2009)
13. Dorr, B., Zajic, D., Schwartz, R.: Hedge Trimmer: a parse-and-trim approach to headline generation. In: Proceedings of the HLT-NAACL 2003 on Text summarization workshop, vol. 5, pp. 1–8. Association for Computational Linguistics, Morristown (2003)
14. Marsi, E., Krahmer, E., Hendrickx, I., Daelemans, W.: Is sentence compression an NLG task? In: Proceedings of the 12th European Workshop on Natural Language Generation, pp. 25–32. Association for Computational Linguistics (2009)
15. Hori, C., Furui, S.: Speech summarization: an approach through word extraction and a method for evaluation. IEICE Transactions on Information and Systems 87, 15–25 (2004)
16. Turner, J., Charniak, E.: Supervised and unsupervised learning for sentence compression. Ann Arbor 100 (2005)
17. Bouma, G., Van Noord, G., Malouf, R.: Alpino: Wide-coverage computational analysis of Dutch. In: Computational Linguistics in the Netherlands 2000. Selected Papers from the 11th CLIN Meeting (2001)
18. Deschacht, K., Moens, M.F.: Semi-supervised semantic role labeling using the latent words language model. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, EMNLP 2009 (2009)
19. Deschacht, K., Moens, M.F.: The Latent Words Language Model. In: Proceedings of the 18th Annual Belgian-Dutch Conference on Machine Learning (2009)

20. Martins, A., Smith, N., Xing, E.: Concise integer linear programming formulations for dependency parsing. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL 2009), Singapore (2009)
21. Denis, P., Baldridge, J.: Joint determination of anaphoricity and coreference resolution using integer programming. In: Proceedings of NAACL HLT, pp. 236–243 (2007)
22. Roth, D., Yih, W.: Integer linear programming inference for conditional random fields. In: Proceedings of the 22nd international conference on Machine learning, p. 743. ACM, New York (2005)
23. Briscoe, T., Carroll, J., Watson, R.: The second release of the RASP system. In: Proceedings of the COLING/ACL, vol. 6 (2006)
24. Ordelman, R., de Jong, F., van Hessen, A., Hondorp, H.: Twnc: a multifaceted Dutch news corpus. ELRA Newsletter 12(3/4), 4–7 (2007)
25. 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)

GEMS: Generative Modeling for Evaluation of Summaries

Rahul Katragadda

Language Technologies Research Center,
IIIT Hyderabad
rahul_k@research.iiit.ac.in

Abstract. Automated evaluation is crucial in the context of automated text summaries, as is the case with evaluation of any of the language technologies. In this paper we present a Generative Modeling framework for evaluation of *content* of summaries. We used two simple alternatives to identifying *signature-terms* from the reference summaries based on *model consistency* and *Parts-Of-Speech (POS) features*. By using a Generative Modeling approach we capture the sentence level presence of these *signature-terms* in peer summaries. We show that parts-of-speech such as noun and verb, give simple and robust method to *signature-term* identification for the Generative Modeling approach. We also show that having a large set of 'significant *signature-terms*' is better than a small set of 'strong *signature-terms*' for our approach. Our results show that the generative modeling approach is indeed promising — providing high correlations with manual evaluations — and further investigation of *signature-term* identification methods would obtain further better results. The efficacy of the approach can be seen from its ability to capture '*overall responsiveness*' much better than the state-of-the-art in distinguishing a human from a system.

1 Introduction

Automated Text Summarization has rapidly become an important tool for information access in today's ever expanding size of the web. Text summarization addresses the problem of finding relevant information from a set of multiple relevant documents and presenting that information in a readable format. Automated text summarization deals with both the problem of finding relevant and salient information, and the presentation of this information, i.e, content and form respectively.

Evaluation is a crucial component in the area of automatic summarization; it is useful both to rank multiple participant systems in a shared tasks, such as the summarization track at TAC 2009, 2008 and its DUC predecessors, and to developers whose goal is to improve the summarization systems. Summarization evaluation, as has been the case with other information access technologies, can foster the creation of reusable resources and infrastructure; it creates an environment for comparison and replication of results; and it introduces an element of

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 724–735, 2010.

competition to produce better results [1]. However, manual evaluation of a large number of documents necessary for a relatively unbiased view is often unfeasible, especially since multiple evaluations are needed in future to track incremental improvement in systems. Therefore, there is an urgent need for reliable automatic metrics that can perform evaluation in a fast and consistent manner.

Summarization Evaluation, like Machine Translation (MT) evaluation, can be broadly classified into two categories [2]. The first, an *intrinsic* evaluation, tests the summarization system in itself. The second, an *extrinsic* evaluation, tests the summarization system based on how it affects the completion of some other task. In the past *intrinsic* evaluations have assessed mainly informativeness and coherence of the summaries. Meanwhile, *extrinsic* evaluations have been used to test the impact of summarization on tasks like reading comprehension, relevance assessment, etc.

Intrinsic Evaluations. Intrinsic evaluations are where the quality of the created automated summary is measured directly. Intrinsic evaluations requires some standard or model against which to judge the summarization quality and this standard is usually operationalized by utilizing an existing abstract/text dataset or by having humans create model summaries [3]. Intrinsic evaluations have taken two major forms: *manual*, in which one or more people evaluate the system produced summary and *automatic*, in which the summary is evaluated without the human in the loop. But *both* types involve human judgments of some sort and with them their inherent variability.

Extrinsic Evaluation. Extrinsic evaluations are where one measures indirectly how well a summary performs by measuring performance in a task putatively dependent on the quality of summary. Extrinsic evaluations require the selection of an appropriate task that could use summarization and measure the effect of using automatic summaries instead of original text. Critical issues here are the selection of a sensible real task and the metrics that will be sensitive to differences in quality of summaries.

Assessment of Evaluations. Overall, from the literature on text summarization, we can see, along with some definite progress in summarization technology, that automated summary evaluation is more complex than it originally appeared to be. A simple dichotomy between intrinsic and extrinsic evaluations is too crude, and by comparison with other Natural Language Information Processing (NLIP) tasks, evaluation at the intrinsic end of the range of possibilities is of limited value. The forms of gold-standard quasi-evaluation that have been thoroughly useful for other tasks like speech transcription, or machine translation and to some, though lesser, extent for information extraction or question answering, are less indicative of the potential value for summaries than in these cases. At the same time, it is difficult even at such apparently fine-grained forms of summarization evaluations as nugget comparisons, when given the often complex systems involved, to attribute particular performance effects to certain particular

system features or to discriminate among the systems. All this makes the potential task in context extremely problematic. Such a Catch-22 situation is displayed appropriately in [4,5]: they attribute poor system performance (for extractive summarizing) to human gold standard disagreement, so humans ought to agree more. But attempting to specify summarizing requirements so as to achieve this may be as much misconceived as impossible. Similar issues arise with Marcu's development of test corpora from existing source summary data [6].

Despite all the complexity in the process of evaluation of summaries, more accurate automated evaluations of content of a summary is relatively possible. In the past simple approaches such as N-gram matching with reference summaries [7] have shown to produce high correlations with manual evaluations. In this paper we address the problem of *automated intrinsic evaluations* of summary content by modeling the capability of a system in generating *signature-terms* for a summary.

2 Current Summarization Evaluations

In the Text Analysis Conference (TAC) series and the predecessor, the Document Understanding Conferences (DUC) series, the evaluation of summarization quality was conducted using both manual and automated metrics. Manual assessment, performed by human judges centers around two main aspects of summarization quality: *informativeness/content* and *readability/fluency*. Since manual evaluation is still the undisputed gold standard, both at TAC and DUC there was a phenomenal effort to evaluate manually as much data as possible.

2.1 Content Evaluations

The content or informativeness of a summary has been evaluated based on various manual metrics. Earlier, NIST assessors used to rate each summary on a 5-point scale based on whether a summary is "very poor" to "very good". Since 2006, NIST uses the Pyramid framework to measure content responsiveness. In the pyramid method as explained in [8], assessors first extract all possible "information nuggets", called Summary Content Units *(SCUs)*, from human-produced model summaries on a given topic. Each SCU has a weight associated with it based on the number of model summaries in which this information appears. The final score of a peer summary is based on the recall of nuggets in the peer.

All forms of manual assessment is time-consuming, expensive and not repeatable. Such assessment doesn't help system developers – who would ideally like to have fast, reliable and most importantly *automated* evaluation metric that can be used to keep track of incremental improvements in their systems. So despite the strong manual evaluation criterion for informativeness, time tested

automated methods like ROUGE, Basic Elements (BE) have been regularly employed, which positively correlate with manual evaluation metrics like '*modified pyramid score*', '*content responsiveness*' and '*overall responsiveness*' of a summary. The creation and testing of automatic evaluation metrics is therefore an important research avenue where the goal is to create automated evaluation metrics that correlate very highly with these manual metrics.

ROUGE stands for Recall-Oriented Understudy for Gisting Evaluation. It includes measures to automatically determine the quality of a summary by comparing it to other (ideal) summaries created by humans. The measures count the number of overlapping units such as n-gram, word sequences, and word pairs between the computer-generated summaries to be evaluated and the human summaries. Basic Elements (BE), also known as ROUGE-BE [9] is a metric based on the recall of *basic elements*. The basic elements include unigrams, bigrams and dependency relations that appear in a summary, and the evaluation involves the amount of recall of these elements from reference summaries. Recently, [10] extended Basic Elements Recall based evaluations by following a set of transformations on the basic elements. Some effort [11] has been devoted for the automation of the pyramid method, while some research on automated summarization evaluation without human models [12] has been pursued and is of severe interest to the community. Recent tasks at the focused evaluations at TAC have targetted the aspect of "Automatically Evaluating Summaries Of Peers"[1] and the work reported in this paper is in context with the efforts at TAC.

3 Automated Content Evaluations

Based on the arguments set above, automated evaluation of both the content and form are necessary for tracking the developers incremental improvements, and a focused task on creation of automated metrics for content and form would help in the process. This was precisely the point being addressed at the TAC Automatically Evaluating Summaries of Peers (AESOP) task. In TAC 2009, AESOP task involves only "Automated Evaluation of Content and Responsiveness", and this paper addresses the same. The output of the automated metrics are compared against two manual metrics: *(modified) pyramid score*, which measures summary content and *overall responsiveness*, which measures a combination of content and linguistic quality.

Experimental Configuration. In TAC 2009 dataset, for each topic we have 4 reference summaries and 55 peer summaries. The task output is to generate, for each peer summary, a score representing (in the semantics of the metric) the goodness of the summary content, measured against or without the use of model summaries. A snapshot of the output obtained from a metric is shown in Figure 1.

[1] http://nist.gov/tac/2009/Summarization/index.html

Setting	Summary	Score
AllPeers	D0911-A.M.100.C.15	0.16185179
AllPeers	D0936-A.M.100.G.50	0.15664380
AllPeers	D0911-A.M.100.C.16	0.07936582
AllPeers	D0936-A.M.100.G.51	0.15486772
AllPeers	D0911-A.M.100.C.13	0.09932766
⋮	⋮	⋮
NoModels	D0940-B.M.100.H.48	0.12425819
NoModels	D0922-B.M.100.D.41	0.06903358
NoModels	D0940-B.M.100.H.49	0.11649885
NoModels	D0922-B.M.100.D.40	0.11413575
NoModels	D0922-B.M.100.D.43	0.09901282
⋮	⋮	⋮

Fig. 1. Sample output generated by evaluation metric

4 Approach

We followed a generative modeling based approach to summarization evaluation where we modeled the amount of signature-terms being captured by peer summaries at sentence level based on how they are distributed in the source documents.

4.1 Generative Modeling of Reference Summaries

In [13] we described two models based on the '*generative modeling framework*': a binomial model and a multinomial model, which we used to show that automated systems are being *query-biased* to be able to perform better on ROUGE like surface metrics. Our approach uses the same generative models to evaluate summaries. In the following sections, we describe how various *signature-terms* extracted from reference summaries can be used in modeling how strongly a peer summary is able to imitate reference summaries.

We use generative modeling to model the distribution of *signature-terms* in the source and obtain the "likelihood of a summary being biased towards these *signature-terms*". In the following sections we describe the two models of generative modeling, Binomial and Multinomial models.

Binomial Model. Let us consider there are 'k' words that we consider *signature-terms*, as identified by any of the methods described in Section 4.2. The sentences in the input document collection are represented as a binomial distribution over the type of sentences. Let $C_i \in \{C_0, C_1\}$ denote classes of sentences without and with those '*signature terms*' respectively. For each sentence $s \in C_i$ in the input collection, we associate a probability $p(C_i)$ for it to be emitted into a summary based on the class C_i to which it belongs.

Then the likelihood of a summary that it would generate a signature-term is:

$$L[summary; p(C_i)] = \frac{N!}{n_0! n_1!} p(C_0)^{n_0} p(C_1)^{n_1} \quad (1)$$

Where N is the number of sentences in the summary, and $n_0 + n_1 = \mathrm{N}$; n_0 and n_1 are the cardinalities of C_0 and C_1 in the summary.

Multinomial Model. Previously, we described the binomial model where we classified each sentence into two classes, as being biased towards a *signature term* or not. However, if we were to quantify the amount of *signature-term bias* in a sentence, we associate each sentence to one among k possible classes leading to a multinomial distribution. Let $C_i \in \{C_0, C_1, C_2, \ldots, C_k\}$ denote the k levels of *signature-term bias* where C_i is the set of sentences having i signature terms.

The number of sentences participating in each class varies highly, with C_0 bagging a high percentage of sentences and the rest $\{C_1, C_2, \ldots, C_k\}$ distributing among themselves the rest sentences. Since the distribution is highly-skewed to the left, distinguishing systems based on log-likelihood scores using this model is easier and perhaps more accurate.

The likelihood of a summary that it would generate sentences that contain signature-terms is:

$$L[summary; p(C_i)] = \frac{N!}{n_0! n_1! \cdots n_k!} p(C_0)^{n_0} p(C_1)^{n_1} \cdots p(C_k)^{n_k} \quad (2)$$

Where N is the number of sentences in the 'peer summary', and $n_0 + n_1 + \cdots + n_k = \mathrm{N}$; n_0, n_1,$\cdots$,n_k are respectively the cardinalities of C_0, C_1, $\cdots$,C_k, in the summary.

4.2 Signature Terms

The likelihood of certain characteristics based on the *binomial* or *multinomial* model shows how well those characteristics of the input have been captured in a summary. In applying our approach, we need keywords from the reference summaries that are considered to be very important for the topic/query combination. We choose multiple alternative methods for the identification of such signature-terms. Here we list these methods:

1. Query terms
2. Model consistency
3. Part-Of-Speech (POS)

Query Terms. *Query-bias* in sentences could be seen as a trivial way of trying to find sentence relevance to the query [14] and if we consider *query terms* as the characteristics that discriminate important sentences from unimportant ones, we obtain the likelihood of a summary emitting a *query-biased* sentence. Earlier, [13] have shown that such a likelihood has very high system-level correlation with ROUGE scores. Since ROUGE correlates very highly with manual evaluations

('*pyramid evaluation*' or '*overall responsiveness*'), a naïve assumption is that likelihood modeling of *query-bias* would correlate well with manual evaluations. This assumption led us to use this method as a baseline for our experiments. Our baselines for this work have been explained further in Section 5.

Model Consistency. Human reference summaries are written by experts in summary writing tasks, and it can be safely assumed that the reference summaries contain least amount of redundant information (or even un-important information) since humans would not want to waste the scarce real-estate available for the summary. Hence the hypothesis behind this method is that a term is important if it is part of a reference summary. In this method we obtain all the terms that are commonly agreed upon by reference summaries. Our idea is that the more the reference summaries agree the more important they are; that is if 3 reference summaries agree on word X and 2 reference summaries agree on the word Y then X is more important than Y, in similar vein to how SCUs were treated in [8]. This is based on the assumption that word level importance sums up towards sentence inclusion. There are 4 reference summaries available for each topic, and we can use the reference agreement in two ways:

- **Total Agreement.** In the case of *total agreement*, only the words that occur in all reference summaries are considered to be important. This case leads to only a single run which we would call '*total-agreement*'.
- **Partial Agreement.** In the case of *partial agreement*, words that occur in at least 'k' reference summaries are considered to be important. Since there are 4 reference summaries per topic, a term would be considered a '*signature term*' if it occurs in 'k' of those 4 reference summaries. There were a total of 3 runs in this case: '*partial-agreement-1*', '*partial-agreement-2*' and '*partial-agreement-3*'.

POS Features. We hypothesized that a certain type of words (or parts-of-speech) could be more informative than the other words, and that in modeling their occurrence in peer summaries we are defining informativeness of the peers with respect to models.

Part-of-Speech Tagger. Traditional grammar classifies words based on eight *parts-of-speech*: the verb, the noun, the adjective, the pronoun, the adverb, the preposition, the conjunction and the interjection. Each part of speech explains not what the word is, but how the word is used. Infact the same word can be a noun in one sentence and a verb or adjective in another. We have used the Penn Treebank Tag-set [15] for our purposes. For automated tagging we have used the Stanford POS tagger [16,17] in these experiments.

Tag Subset Selection – Feature Selection. Based on an analysis of how each 'POS tag' performs at the task we selectively combine the set of features. We used the following 'POS tag' features: *NN*, *NNP*, *NNPS*, *VB*, *VBN*, *VBD*, *CD*, *SYMB*, and their combinations. We zeroed on to a final list of combinations that form the runs described in this paper based on thorough experimentation with various combinations of these features.

The final list of runs comprises of some of the individual 'POS tag' features and some combinations, they are:

- NN
- NNP
- NNPS
- NOUN – A combination of NN, NNP and NNPS.
- VB
- VBN
- VBD
- VERB – A combination of VB, VBN and VBD.
- CD
- SYMB
- MISC – A combination of CD and SYMB.
- ALL – A combination of NOUN, VERB and MISC.

5 Experiments and Evaluations

Our experimental setup was primarily defined based on how signature terms have been identified. We have detailed few methods of identification of signature-terms in Section 4.2. For each method of identifying *signature terms* we have one or more runs as described earlier.

5.1 Baselines

Apart from the set of runs described in Section 4.2, we propose to use the following two baselines.

- Binomial modeling for *query terms*. This approach uses the Binomial model described in Section 4.1 to obtain the likelihood that a system would generate summaries that comprises of sentences containing query-terms.
- Multinomial modeling for *query terms*. This baseline approach uses the Multinomial model described in Section 4.1 to obtain the multinomial likelihood that a system would generate summaries that comprises of sentences containing query-terms. This model distinguishes sentences that contain a single query-term to sentences containing two query-terms and so on.

5.2 Datasets

The experiments shown here were performed on TAC 2009 update summarization datasets which have 44 topics and 55 system summaries for each topic apart from 4 human reference summaries. And since in our methods there is no clear way to distinguish evaluation of cluster A's or cluster B's summary – we don't evaluate the update of a summary – we effectively have 88 topics to evaluate on. Despite that we report the results of both the clusters A and B separately in Tables 1 and 2.

Table 1. Correlation scores for Cluster A

RUN	Pyramid		Responsiveness	
	AllPeers	NoModels	AllPeers	NoModels
High Baselines				
ROUGE-SU4	0.734	0.921	0.617	0.767
Basic Elements (BE)	0.586	0.857	0.456	0.692
Baselines				
Binom(query)	0.217	0.528	0.163	0.509
Multinom(query)	0.117	0.523	0.626	0.514
Experimental Runs				
POS based				
NN	0.909	0.867	0.853	0.766
NNP	0.666	0.504	0.661	0.463
NOUN	0.923	0.882	0.870	0.779
VB	0.913	0.820	0.877	0.705
VBN	0.931	0.817	0.929	0.683
VBD	0.944	0.859	0.927	0.698
VERB	0.972	0.902	**0.952**	0.733
CD	0.762	0.601	0.757	0.561
MISC	0.762	0.601	0.757	0.561
ALL	**0.969**	**0.913**	0.934	**0.802**
Model Consistency/Agreement				
total-agreement	0.727	0.768	0.659	0.682
partial-agreement-3	0.867	0.856	0.813	0.757
partial-agreement-2	0.936	0.893	0.886	**0.791**
partial-agreement-1	**0.966**	**0.895**	**0.930**	0.768

5.3 Evaluations

This paper dictates automated approaches that imitate manual evaluation metric for content responsiveness. Each evaluation metric produces a score for each summary, and an evaluation of these new *summarization evaluation metrics* is done based on how well these new metrics correlate with manual evaluations at system level averages. This task, despite the complexity involved, boils down to a simpler problem, that of information ordering. We have a reference ordering and have various metrics that provide their own ordering for these systems. Comparing an ordering of information with another is a fairly well understood task and we would use correlations between these manual metrics and the metrics we proposed in this work to show how well our metrics are able to imitate human evaluations in being able to generate similar ordering of systems. We used Pearson's Correlation (r) — a non-parametric measure of correlation that assesses how an arbitrary monotonic function might be able to describe the relationship between two variables, making an assumption that the nature of the relationship between the variables is linear — of system level averages based on our metrics and by the manual methods.

Apart from distinguishing the *content responsiveness* of system summaries against each other, distinguishing a reference summary from a system summary is also an equally important task, which may be easy for automated approaches. To comprehend the capability of a metric in distinguishing a reference summary

Table 2. Correlation scores for Cluster B

RUN	Pyramid		Responsiveness	
	AllPeers	NoModels	AllPeers	NoModels
High Baselines				
ROUGE-SU4	0.726	0.940	0.564	0.729
Basic Elements (BE)	0.629	0.924	0.447	0.694
Baselines				
Binom(query)	0.210	0.364	0.178	0.372
Multinom(query)	-0.004	0.361	-0.020	0.446
Experimental Runs				
POS based				
NN	0.908	0.845	0.877	0.788
NNP	0.646	0.453	0.631	0.380
NOUN	0.909	0.848	0.878	0.783
VB	0.872	0.871	0.875	0.742
VBN	0.934	0.873	**0.944**	0.720
VBD	0.922	0.909	0.914	0.718
VERB	0.949	**0.951**	0.942	0.784
CD	0.807	0.599	0.800	0.497
MISC	0.807	0.599	0.800	0.497
ALL	**0.957**	0.921	0.931	**0.793**
Model Consistency/Agreement				
total-agreement	0.811	0.738	0.808	0.762
partial-agreement-3	0.901	0.839	0.882	0.806
partial-agreement-2	0.949	0.898	0.924	**0.817**
partial-agreement-1	**0.960**	**0.903**	**0.936**	0.763

from a system summary we use two cases, AllPeers and NoModels. If a metric is able to obtain high correlations in AllPeers case when compared to NoModels case then it means the metric is able to distinguish reference summaries from system summaries.

6 Results

Our vision for these focused experiments were to create alternatives to the manual content evaluation metrics. In achieving this we must create metrics that are neither 'too expensive' nor 'non-replicable' nor both, and yet be able to capture the responsiveness of a summary towards a query on a topic. It is unlikely that any single automated evaluation measure would be able to correctly reflect both readability and content responsiveness, since they represent form and content which are separate qualities of a summary and would need different measures. We chose to imitate content since having better content in a summary is more important than having 'just' a readable summary.

6.1 Discussion

We have used two separate settings for displaying results: an *AllPeers* case and a *NoModels* case. AllPeers case consists of the scores returned by the metric for

all the summarizers (automated and human), while in the case of NoModels case only automated summarizers are scored using the evaluation metrics. This setup helps distinguish methods that are able to differentiate two things:

- Metrics that are able to differentiate humans from automated summarizers.
- Metrics that are able to rank automated summarizers in the desired order.

Results[2] have shown that no single metric is good at distinguishing everything, however they also show that certain type of keywords have been instrumental in providing the key distinguishing power to the metric. For example, *VERB and NOUN* features have been key the contributors to *ALL* run. Also as an interesting side note we observe that having high number of 'significant' signature-terms seems to be better than a low number of *'strong' signature-terms*, as seen from the experiments on *total-agreement* and *partial-agreement*. The most important result of our approach has been that our method was very highly correlated with "overall responsiveness", which again is a very good sign for an evaluation metric.

7 Conclusion

In the context of *Automatically Evaluating Summares Of Peers* task, we model the problem as an information ordering problem; our approach (and indeed others) is now to rank systems (and possibly human summarizers) in the same order as human evaluation would have produced. We show how a well known generative model could be used to create automated evaluation systems comparable to the state-of-the-art. Our method is based on a multinomial model distribution of *signature terms* in document collections, and how they are captured in peers.

We have used two types of signature-terms to model the evaluation metrics. The first is based on POS tags of important terms in a model summary and the second is based on how much information the reference summaries shared among themselves. Our results show that verbs and nouns are key contributors to our best run which was dependent on various individual features. Another important observation was that all the metrics were consistent in that they produced similar results for both cluster A and cluster B (update summaries). The most startling result is that in comparison with the automated evaluation metrics currently in use (ROUGE, Basic Elements) our approach has been very good at capturing "overall responsiveness" apart from pyramid based manual scores.

References

1. Hirschman, L., Mani, I.: Evaluation (2001)
2. Jones, K.S., Galliers, J.R.: Evaluating Natural Language Processing Systems: An Analysis and Review. Springer, New York (1996)

[2] We have excluded *NNPS* and *SYMB* from the analysis since they didn't have enough samples in the testset, so as to obtain consistent results.

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. Lin, C.Y., Hovy, E.: Automatic evaluation of summaries using n-gram co-occurrence statistics. In: NAACL 2003: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, Morristown, NJ, USA, pp. 71–78. Association for Computational Linguistics (2003)
5. Lin, C.-Y., Hovy, E.: The potential and limitations of automatic sentence extraction for summarization. In: Proceedings of the HLT-NAACL 2003 on Text summarization workshop, Morristown, NJ, USA, pp. 73–80. Association for Computational Linguistics (2003)
6. Marcu, D.: The automatic construction of large-scale corpora for summarization research, University of California, Berkely, pp. 137–144 (1999)
7. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: The proceedings of ACL Workshop on Text Summarization Branches Out. ACL (2004)
8. Nenkova, A., Passonneau, R., McKeown, K.: The pyramid method: Incorporating human content selection variation in summarization evaluation. ACM Trans. Speech Lang. Process. 4 (2007)
9. Hovy, E., Lin, C.-y., Zhou, L., Fukumoto, J.: Automated summarization evaluation with basic elements. In: Proceedings of the Fifth Conference on Language Resources and Evaluation, LREC (2006)
10. Tratz, S., Hovy, E.: Summarization evaluation using transformed basic elements. In: Proceedings of Text Analysis Conference (2008)
11. Ani, A.H., Nenkova, A., Passonneau, R., Rambow, O.: Automation of summary evaluation by the pyramid method. In: Proceedings of the Conference of Recent Advances in Natural Language Processing (RANLP), p. 226 (2005)
12. Louis, A., Nenkova, A.: Automatically evaluating content selection in summarization without human models. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, Singapore, pp. 306–314. Association for Computational Linguistics (2009)
13. Katragadda, R., Varma, V.: Query-focused summaries or query-biased summaries ? In: Proceedings of the joint conference of the 47th Annual meeting of the Association of Computational Linguistics and the 4th meeting of International Joint Conference on Natural Language Processing, ACL-IJCNLP 2009. Association of Computational Linguistics (2009)
14. Gupta, S., Nenkova, A., Jurafsky, D.: Measuring importance and query relevance in topic-focused multi-document summarization. ACL companion volume, 2007 (2007)
15. Marcus, M.P., Marcinkiewicz, M.A., Santorini, B.: Building a large annotated corpus of english: the penn treebank. Comput. Linguist. 19, 313–330 (1993)
16. Toutanova, K., Manning, C.D.: Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In: Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora, Morristown, NJ, USA, pp. 63–70. Association for Computational Linguistics (2000)
17. Toutanova, K., Klein, D., Manning, C.D., Singer, Y.: Feature-rich part-of-speech tagging with a cyclic dependency network. In: NAACL 2003: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, Morristown, NJ, USA, pp. 173–180. Association for Computational Linguistics (2003)

Quantitative Evaluation of Grammaticality of Summaries

Ravikiran Vadlapudi and Rahul Katragadda

Language Technologies Research Center,
IIIT Hyderabad
{ravikiranv,rahul_k}@research.iiit.ac.in

Abstract. Automated evaluation is crucial in the context of automated text summaries, as is the case with evaluation of any of the language technologies. While the quality of a summary is determined by both content and form of a summary, throughout the literature there has been extensive study on the automatic and semi-automatic evaluation of content of summaries and most such applications have been largely successful. What lacks is a careful investigation of automated evaluation of readability aspects of a summary. In this work we dissect readability into five parameters and try to automate the evaluation of grammaticality of text summaries. We use surface level methods like Ngrams and LCS sequence on POS-tag sequences and chunk-tag sequences to capture acceptable grammatical constructions, and these approaches have produced impressive results. Our results show that it is possible to use relatively shallow features to quantify degree of acceptance of grammaticality.

1 Introduction

Automated Text Summarization has rapidly become an important tool for information access in today's ever expanding size of the web. Text summarization addresses the problem of finding relevant information from a set of multiple relevant documents and presenting that information in a readable format. Automated text summarization deals with both the problem of finding relevant information and the presentation of the information, i.e, content and form respectively.

Evaluation is a crucial component in the area of automatic summarization; it is used both to rank multiple participant systems in shared tasks, such as the summarization tracks at TAC 2009, 2008 and its DUC predecessors, and to developers whose goal is to improve the summarization systems. Summarization evaluation help in the creation of reusable resources and infrastructure; it sets up the stage for comparison and replication of results by introducing an element of competition to produce better results [1].

Manual Readability Evaluations. Readability or Fluency of a summary is evaluated based on certain set of linguistic quality questions that manual assessors answer for each summary. The linguistic quality markers are: *Grammaticality*,

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 736–747, 2010.

Non-Redundancy, *Referential Clarity*, *Focus* and *Structure and Coherence*. Readability assessment is primarily a manual method following Likert Scale rating given by human assessors for each of the linguistic quality markers. All the above linguistic quality markers are rated individually for each summary and an average score for linguistic quality of various peers are compared against each other. An ANOVA (Analysis of Variance) is performed on the linguistic quality markers to show which set of peers fall in a statistically similar range. Manual evaluation, of a large number of documents necessary for a relatively unbiased view is often unfeasible, especially since multiple evaluations are needed in future to track incremental improvement in systems. Any kind of manual evaluation is time-consuming, expensive and possibly not repeatable. Such assessment doesn't help system developers – who would ideally like to have fast, reliable and most importantly *automated* evaluation metric that can be used to keep track of incremental improvements in their systems. So despite the strong manual evaluation criterion for readability, there is an urgent need for reliable automatic metrics that can perform evaluation in a fast and consistent manner.

Automated Readability Evaluations. Early studies on readability assessment have provided us with numerous techniques that are based on approximations of complexities of syntactic structures. Flesch Reading Ease[2], for example, used the number of words in a sentence and the number of syllables in a word as approximations of syntactic complexity of a sentence and semantic complexity of a word respectively. On the other hand, Gunning-Fog Index [3] used 'number of complex words' as a criterion to rate readability of a text. Flesch Readability Ease, Gunning-Fog Index, Automated Readability Index and Simple Measure of Gobbledygook (SMOG) have been used as baselines for our experiments.

For the readability aspects of summary evaluation there hasn't been much of dedicated research with text summaries. Discourse-level constraints on adjacent sentence, have been relatively fairly investigated, indicative of *coherence* and *good text-flow* [4,5,6]. In a lot of applications, like in "*overall responsiveness*" for text summaries, readability is assessed in combination with other qualities. In machine translation scenarios, approaches such as BLEU use n-gram overlap [7] with a reference sentence to judge "*overall goodness*" of a translation. With BLEU, higher sized n-grams' overlap were meant to capture fluency considerations, while all the n-gram overlaps together contribute to the translation's "*content goodness*". In some related work in NLG [8,9] directly set a goal of sentence level fluency regardless of content. Some recent work [10] performed a systematic study on how syntactic features were able to distinguish machine generated translations from human translations. And another related work [11] investigated the impact of certain *linguistic surface features*, *syntactic features*, *entity coherence features* and *discourse features* on the readability of Wall Street Journal (WSJ) Corpus. Further, [12,13] developed a tool for automatically rating the readability of texts for adult users with intellectual disabilities.

There has been no known work in the area of characterizing grammaticality of a summary automatically, from text summaries point of view. Application of the above methods and in particular syntactic and semantic features expressed

in [10,11] to create an automated metric to evaluate summaries would be an interesting area of research.

Readability of a summary depends on (at least) the five linguistic quality measures described earlier. Since all these linguistic quality measures are independent of each other, an evaluation of readability of a summary is dependent on how we are able to capture these independent measures. In this paper, we address the problem of capturing grammaticality of a summary.

2 Grammaticality in Summaries

Grammaticality is defined as the quality of a linguistic utterance of being grammatically well-formed. In theory, a sentence is either grammatically correct or grammatically incorrect. Generative grammar defines a set of rules which can classify a sentence to be grammatically well-formed or grammatically ill-formed. The grammar attempts to do this by taking into consideration the acceptance of a sentence structure by native speakers of the language. The Generative grammar rules formed based on a single native speaker's utterances is not sufficient for classification as he may not be familiar with all possible constructions of the language. Moreover, he might be familiar only with his own dialect. Therefore, Grammaticality can be better estimated by considering it to be the *degree of acceptance* by multiple native speakers rather than correctness/incorrectness of structure. Stochastic grammar is one such grammar which considers grammaticality as a probabilistic random variable, which we define in our approach as the degree of acceptance. By extracting a set of grammar rules using a corpora and by learning probabilistic language models over these rules, we calculate the degree of acceptance of an utterance.

In the context of this paper, 'grammaticality' refers to the degree of acceptance of a sentence based on its eligibility to be a part of a summary. The intuition here is that human written summary sentences are deliberately written to follow a definite style, and this pattern can be captured relatively easily using a corpus. Each sentence in a system summary might be a *sentence extract* from the source document. Though it might be grammaticality correct as part of a text, it may not be suitable as a summary sentence. We believe that this is the major reason why system generated summaries obtain a low score on human evaluation of grammaticality, and hence we try to compute this acceptability of a sentence in a summary in this paper.

The grammaticality of a summary solely depends on the grammaticality of the sentences that constitute the summary since grammaticality is more a syntactic property. We say a sentence is grammatically well formed when the adjacent words are in agreement with each other. That is, the parts-of-speech (POS) tags of adjacent words are mutually compatible, where the level of compatibility accounts to the degree of acceptance. Exploiting this property of compatibility, in this work, we give three methods to effectively rate the grammaticality of a summary generated by a summarization system ensuring that the rating is close to human judgment.

A set of generative grammar rules are generated using a corpus of manually written text by native speakers. Since these texts would cover many of the grammatical constructions accepted by a reasonable number of native speakers it would be a good source of learning probabilistic models. But our goal is to evaluate summaries — which have a constraint on length — and not texts. Hence, a corpus of manually written summaries by humans would be more effective. A system trained on the rules of the grammar based on such corpus would be the best method to determine the rating of grammaticality. This motivated us to use a corpus of reference summaries available from text summarization shared tasks[1].

The reference summaries are human written summaries with respect to a given topic against which we rate the system generated summaries, also called peer summaries. We consider the sequences of the POS tags for learning the language models from the reference summaries. These sequences are a set of generative grammar rules since they are generated from manually written summaries which are assumed to be a source of grammatically acceptable sentences.

As a part of training data generation, POS-tag sequences in reference summaries are extracted using the Stanford POS tagger [14,15] and chunk tags for the same are extracted using a standard chunker. We use these POS-tag sequences and chunk-tag sequences to build the POS-tag training corpus and chunk-tag training corpus, respectively.

As part of training, frequencies of unigram, bigram and trigram sequences of POS-tag sequences in the POS-tag training corpus are computed. Similarly frequencies of unigram, bigram and trigram sequences of chunk-tag sequences in the chunk-tag training corpus are computed. A bigram or trigram sequence is considered as a grammar rule and its frequency in the corpus shows to what level this rule is acceptable to the authors of the reference summaries. Trigram sequences of higher frequency are highly acceptable than a Trigram of lower frequency. The following approaches use these probabilistic models in three different ways to determine the grammaticality of a sentence.

2.1 Ngram Model

In this model, we estimate the probability of a sentence to be grammatically acceptable with respect to the corpus using language models. Sentences constructed using frequent grammar rules would have higher probability and are said to have a well accepted sentence structure.

In the preprocessing phase, for each sentence in the peer summary, POS-tag sequence is extracted using the POS tagger. All possible Trigram, Bigram and Unigram sequences are generated using the POS-tag sequence. The Trigrams, as they are better estimates than Bigrams or Unigrams, are considered to be the underlying grammar rules of the sentence without bringing in the sparsity issues of any higher ngrams. Probability of each Trigram is calculated and probabilities

[1] Document Understanding Conferences (DUC) and Text Analytics Conferences (TAC).

of all Trigrams together estimate the score of the sequence. The cumulative score of all the sentence sequences gives the final score of the summary.

The probability of a sequence is defined as follows:

$$G(S) = P(Seq) \tag{1}$$

$$P(Seq) = \prod_{i=1}^{n} P(K_i) \tag{2}$$

$$P(K_i) = P(t_{i-2}t_{i-1}t_i) \tag{3}$$

For example, To estimate the grammaticality $G(S)$ of a sentence S we estimate the probability of its POS-tag sequence Seq. The probability of Seq is calculated from the estimated probabilities of its trigram sequences K_1, K_2,.....,K_N, where K_i is $t_{i-2}t_{i-1}t_i$ and $\forall t_j$ are $t_j \in POS\ tags$. The additional tags t_{-1}, t_0 and t_{n+1} are the beginning-of-sequence and end-of-sequence markers. The probability of trigrams (See Eq 4) generated from training corpus usually cannot directly be used because of sparsity problem. This means that there are not enough instances for each trigram to reliably estimate the probability and leads to a zero probability which is undesirable.

$$P(t_1t_2t_3) = \frac{frequency\ of\ t_1t_2t_3\ in\ training\ corpus}{total\ number\ of\ trigrams\ in\ training\ corpus} \tag{4}$$

Hence, we try to interpolate the probability of trigram from its bigram and unigram frequencies as seen in Eq 5[2] reported in [16].

$$P(t_1t_2t_3) = \lambda_1 * P(t_3|t_1t_2) + \lambda_2 * P(t_3|t_2) + \lambda_3 * P(t_3) \tag{5}$$

where

$$\lambda_1 + \lambda_2 + \lambda_3 = 1$$

$$P(t_3|t_1t_2) = \frac{f(t_1t_2t_3)}{f(t_1t_2)}$$

$$P(t_3|t_2) = \frac{f(t_2t_3)}{f(t_2)}$$

$$P(t_3) = \frac{f(t_3)}{\sum_{\forall t_i} f(t_i)}$$

Here, function $f(T)$ gives the frequency counts of a sequence T in the training corpus. The score of the sequence is $\ll 1$ since it is a product of Trigram probabilities (each value ≤ 1). Hence, Grammaticality values of summaries vary by a very small factor. In order to overcome this problem we take the *logarithm* — a monotonically increasing function — of the probabilities. Hence the probability of a sequence (Seq) is

$$P(Seq) = \log(\prod_{i=1}^{n} P(K_i)) \tag{6}$$

[2] λ_1, λ_2 *and* λ_3 values are 0.5, 0.3 and 0.2 based on empirical tuning.

Let us consider a sentence X with 10 trigrams, each trigram probability = 0.8 and another sentence Y with 3 trigrams, with each trigram probability = 0.4. The scores of these two sequences X, Y would be 0.107 and 0.16 respectively, which says X is grammatically ill-formed compared to Y; this is a contradiction to the fact that X is more grammatical evident from its high trigram probabilities. Hence, longer sentences (with more number of trigrams) which are grammatically acceptable would have a lesser score than a shorter sentence which is ungrammatical. The resultant score of the sequence is biased towards sentence length, and the solution to this problem is to normalize the estimated probability.

$$P(Seq) = \log(\sqrt[n]{\prod_{i=1}^{n} P(K_i)}) \tag{7}$$

The average of the grammaticality scores of sentences in a summary gives the final grammaticality score of the summary.

$$G(S) = \frac{\Sigma_{i=1}^{k} G(s_i)}{k} \tag{8}$$

where $G(S)$ is the grammaticality score of summary S and $G(s_i)$ is the grammaticality score of sentence s_i. The above procedure estimates grammaticality of sentence using its POS tags and we call this run '*Ngram (POS)*'. A similar procedure is followed to estimate grammaticality using its chunk tag sequences and language models trained on chunk-tag training corpus. This run based on the chunk tags in called '*Ngram (Chunk)*' in the results.

2.2 Longest Common Subsequence (LCS)

In this model, we determine how structurally close a sentence is to a grammatically acceptable sentence in a non probabilistic manner. The closeness is determined by the length of the longest subsequence common between a sentence in peer summary and a sentence in the corpus. A subsequence is defined as a sequence that can be derived from another sequence by deleting some elements without changing the order of remaining elements. The common subsequence in this model is generated based on [17] considering the POS-tag sequence or chunk-tag sequence of sentences. For a sentence, if length of the generated subsequence (K) is equal to length of the sentence (N) we can say that the sentence is grammatically acceptable. If $K < N$ then the subsequence cannot effectively convey the extent to which the sentence is grammatically acceptable. So we use probabilistic models on top of the longest common subsequences (LCS) to estimate the extent of acceptability.

As a preprocessing step, for each sentence in peer summary a LCS is generated with respect to each sentence in training corpus using their POS-tag sequences. The grammaticality score of sentence is estimated using the following approaches on the generated longest common subsequences (LCSes).

LCS and Ngram Model. We apply Ngram model, described in Section 2.1, on the LCSes of a sentence to estimate its grammaticality score. The Ngram model estimates the degree of acceptance of grammaticality of the subsequence. Using the lengths of sentence and its subsequence we determine what percentage of a sentence is similar to the grammatically well-formed sentences.

$$G(s_i) = P(Q_i) * \frac{length\ of\ Q_i}{length\ of\ s_i} \tag{9}$$

Grammaticality $G(s_i)$ of sentence s_i is the product of acceptability value $P(Q_i)$ of its longest common subsequence Q_i and percentage correctness. The grammaticality of the summary is average of grammaticality of all sentences in the summary. This run described above based on the Ngram over LCS is referred as '*Ngram(LCS-POS)*'. A similar procedure is followed to estimate the grammaticality using its chunk tag sequences and language models trained on chunk-tag training corpus; this run based on chunk-tags is called '*Ngram(LCS-Chunk)*'.

2.3 Class Model

In this model, we view the task of scoring grammaticality as a classification problem. Sentences are classified into classes on the basis of acceptability of underlying grammar rules. The class boundaries are estimated according to the degree of acceptance of grammar rules and these boundaries classify grammar rules (trigrams) into K classes with each class contributing a different amount to the score. Hence, the cumulative score of the underlying grammar rules estimate the score of grammaticality of a sentence. As mentioned earlier, a trigram sequence is considered as a grammar rule and its frequency in the corpus shows the extent to which this grammar rule is acceptable to the authors of the reference summaries. Therefore, the class boundaries are defined on frequencies of trigrams.

All the summaries are manually evaluated based on five properties one among which is grammaticality. Each property is rated on a scale of 1 to 5 (score 1 being bad and 5 being good). Differently put, this means summaries are manually classified into *five classes* and each class has an assigned value. In the process, each sentence of the summary is also rated on a similar scale. Similarly in our approach, trigrams are classified into *five classes* C_1, C_2, C_3, C_4 and C_5 and each class is assigned a score on a similar scale ($\forall_j score(C_j) = j$). Class boundaries are estimated using the frequencies of trigrams in the training corpus. The most frequent trigram, for example, would fall into class C_5.

As a preprocessing step, class sequences are generated from POS-tag sequences using class boundaries. This is done by classifying each trigram in a POS-tag sequence into a class, say C_i, and ordering the respective class labels of trigrams in the order their occurrence. These class sequences constitute the POS-class training data. Similar procedure is followed on chunk trigrams of chunk-tag corpus generating Chunk-class training data.

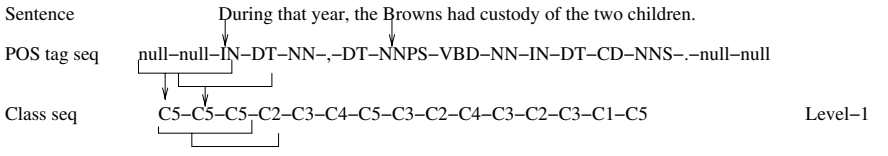


Fig. 1. Class labeling for terms in a sentence at various levels

Given a peer summary, for each sentence POS-tag sequence is generated. The trigrams of this sequences are classified into classes and each trigram is assigned a score depending on the class to which it belongs (See level-1 in Fig. 1). The sequence score is determined by trigram scores.

$$G(S) = AVG(H(C_{k1}), H(C_{k2}), ..., H(C_{kn})) \tag{10}$$

AVG is the average of $H(C_{ki})$, where $k1$, $k2$, $\ldots kn$ are POS trigrams, C_{ki} is class into which trigram ki falls into and $H(C_{ki})$ is score assigned to the class to which ki belongs. This run is referred as '*Class (POS level-1)*' in the results.

Our method above, *Class(POS level-1)*, models grammaticality of a sentence using *POS-tag trigrams* in isolation. That is, the compatibility of adjacent trigrams is not captured. To find the compatibility of a trigram with the adjacent trigrams we can use the Ngram frequencies of these *POS-tag trigrams*. Say for a sentence with 4 words (excluding the three nulls explained in Figure 1) we would be dealing with estimating the frequency of a 6 gram sequence, which is difficult due to sparsity issues. This problem can be resolved using the class sequences by projecting the problem of finding compatibility between *POS-tag trigrams* to finding compatibility between their *class trigrams*. This degree of compatibility of class trigrams can be estimated using the POS-Class training data; this would avoid the problem of sparsity since unlike POS-tag corpus, POS-Class corpus follows a tagset of size 5 and the corpus is also of the same size as POS-tag corpus. We estimate the probability each class sequence using the reference corpus by applying Ngram model on the class sequences. This approach captures the compatibility of a POS-trigram with adjacent trigrams.

For each sentence in peer summary a POS-tag sequence is generated. Based on frequencies of POS-tag trigrams in this sequence a class sequence is obtained. The Ngram model described in Section 2.1 is applied on this class sequence which estimates the degree of acceptance of the sentence. This run is referred as '*Ngram(Class(POS level-1))*' in the results.

Similar to above approach, the grammaticality of sentence can also be estimated using chunk-tag sequence and Chunk-Class training data, and the corresponding runs obtained are referred as '*Class(Chunk level-1)*' and '*Ngram(Class (Chunk level-1))*'.

3 Evaluation

This paper deals with methods that imitate manual evaluation metric for grammaticality. Each evaluation metric produces a score for each summary, and an evaluation of these new *summarization evaluation metrics* is done based on how well these new metrics correlate with manual evaluations at system level averages. This task, despite the complexity involved, boils down to a simpler problem, that of information ordering. We have a reference ordering and have various metrics that provide their own ordering for these systems. Comparing an ordering of information with another is a fairly well understood task and we would use correlations between these manual metrics and the metrics we proposed in this work to show how well our metrics are able to imitate human evaluations in being able to generate similar ordering of systems.

We use 3 types of evaluations each describing some aspect of ordering problems; we use Spearman's Rank Correlation, Pearson's Correlation and Kendall's Tau to evaluate the orderings produced by the metrics. Spearman's Rank Correlation (ρ) is a non-parametric measure of correlation that assesses how an arbitrary monotonic function might be able to describe the relationship between two variables, without making any other assumptions about the particular nature of the relationship between the variables. While Pearson's correlation (r) measures the correlation that assesses the linear dependence between two variables. The Kendall's Tau (τ) is another non-parametric method that captures the degree of correspondence between two rankings and assessing the significance of this correspondence.

The data we used for these experiments were drawn from DUC 2007 and TAC 2008, 2009. We used reference summaries from TAC 2008, 2009 for the reference corpus. And the experiments described were tested on DUC 2007 query-focused multi-document summarization datasets which have 45 topics and 32 system summaries for each topic apart from 4 human reference summaries. Apart from distinguishing the grammaticality of system summaries against each other, distinguishing a reference summary from a system summary is also an equally important task, which may be easy for automated approaches. To comprehend the capability of a metric in distinguishing a reference summary from a system summary we use two cases, AllPeers and NoModels. If a metric is able to obtain high correlations in AllPeers case when compared to NoModels case then it means the metric is able to distinguish reference summaries from system summaries.

Table 3 shows the system level correlations of our approaches to grammaticality assessment with that of human ratings. We have used four baseline approaches: Gunning Fox Index, Flesch Reading Ease, Automatic Readability Index, Simple Measure of Gobbledygook (SMOG). Our approaches constitute of the following runs:

- *Ngram(POS)*
- *Ngram(Chunk)*
- *Ngram(LCS-POS)*
- *Ngram(LCS-Chunk)*

Table 1. System level correlations of automated and manual metrics for grammaticality

RUN	Spearman's ρ		Pearson's r		Kendall's τ	
	AllPeers	NoModels	AllPeers	NoModels	AllPeers	NoModels
Baselines						
Gunning Fox	-0.2542	0.2309	-0.1367	0.2883	-0.1644	0.17930
Flesch Reading Ease	0.0576	-0.3552	-0.0354	-0.3953	0.0442	-0.2613
Automatic Readability index	-0.1583	0.2816	-0.0769	0.3356	-0.1055	0.2181
SMOG	-0.0668	0.3021	0.0028	0.3298	-0.0491	0.2228
Our experiments						
Ngram (POS)	**0.8500**	**0.6993**	**0.8386**	**0.7368**	**0.6626**	**0.5204**
Ngram(Chunk)	0.2041	**0.6035**	0.2623	0.5022	0.1227	0.3866
Ngram(LCS-POS)	0.2257	**0.6495**	0.3343	0.5701	0.1718	0.4428
Ngram(LCS-Chunk)	0.2006	**0.6182**	0.2570	0.5144	0.1301	0.4125
Class(POS level-1)	0.5978	0.3967	**0.6554**	0.5189	0.4123	0.2527
Ngram(Class(POS level-1))	**0.7155**	0.4223	**0.7464**	0.5108	**0.5104**	0.2743
Class(Chunk level-1)	**0.5996**	**0.6688**	**0.6069**	**0.6378**	0.4000	04903
Ngram(Class(Chunk level-1))	**0.6322**	0.3427	**0.6007**	0.4516	0.4712	0.2916

- *Class(POS level-1)*
- *Ngram(Class(POS level-1))*
- *Class(Chunk level-1)*
- *Ngram(Class(Chunk level-1))*

4 Conclusion and Discussion

We have argued that automated readability assessment of text summaries is important in the context of text summarization evaluations since the quality of a summary is directly dependent on both the content and the form of presentation. Manual readability assessment is done based on five distinct attributes of readability, one among which is *grammaticality*. In this paper, we addressed the problem of identifying the degree of acceptance of grammatical formations at sentence level. We used surface features like Ngrams and LCS on the POS-tag sequences and chunk-tag sequences which have produced impressive results.

Our approaches have produced high correlations to human judgment on grammaticality in judging differences in system level performances based on the measures of association of cross-tabulations like Spearman's Rank Correlation, Pearson's Correlation and Kendall's Tau. Results in Table 1 show that Ngram approach on the POS-tag sequences outperforms all the other approaches on all categories. It is interesting to note that '*Ngram(chunk)*', '*Ngram(LCS-POS)*' and '*Ngram(LCS-Chunk)*' have better correlations in the NoModels case where we rank only the system summaries. For example in the case of '*Ngram(chunk)*', which is based on chunk-tag sequences, due to the relatively smaller size of chunk tagset the difference between a highly frequent Ngram and the highest frequent Ngram is lower when compared to the difference between a high frequent Ngram and a low frequency Ngram. Hence we are able to distinguish bad summaries

from good summaries but not good summaries from best summaries; this also explains low scores in the AllPeers case. In case of using the LCS, the difference in *percentage correctness* of a good sentence to the best sentence is marginal, while the same between a good sentence and a bad sentence is high. This property leads to the better performance of *Ngram(LCS-POS)* and *Ngram(LCS-Chunk)* in NoModels case.

In this work we have used Ngram and LCS approaches built on POS tagged data. This is a relatively surface level approach to the problem, we could use more complex syntactic approaches based on parse structures to identify grammaticality of a sentence and hence a summary. Given the relatively abundant data, the combination of the surface level and syntactic features could be used to learn the degree of acceptance of a sentence. We ignore the effect of the position of an ungrammatical sentence in a summary. In theory, however, the presence of an ungrammatical sentence at the beginning of a summary shall effect the readability heavily when compared to a summary in which the last sentence (or a sentence in the middle) is ungrammatical. The position of an ungrammatical sentence must have an impact on the averaging techniques we employ for this task.

The focus of this paper was on capturing the grammaticality aspects of readability of a summary and in future other aspects of readability like focus, coherence, non-redundancy and referential clarity can be addressed. When we are able to achieve the above, we would have independently automated both content evaluations and readability evaluations which would lead to the automated appreciation of overall responsiveness of a summary.

References

1. Hirschman, L., Mani, I.: Evaluation (2001)
2. Flesch, R.: A new readability yardstick. Journal of Applied Psychology 32, 221–233 (1948)
3. Gunning, R.: The technique of clear writing. McGraw-Hill International Book Co., New York (1952)
4. Lapata, M.: Probabilistic text structuring: Experiments with sentence ordering. In: Proceedings of the annual meeting of the Association for Computational Linguistics, pp. 545–552. The Association of Computational Linguistics (2003)
5. Lapata, M., Barzilay, R.: Automatic evaluation of text coherence: Models and representations. In: Kaelbling, L.P., Saffiotti, A. (eds.) IJCAI, pp. 1085–1090. Professional Book Center (2005)
6. Barzilay, R., Lapata, M.: Modeling local coherence: An entity-based approach. Comput. Linguist. 34, 1–34 (2008)
7. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: ACL 2002: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, Morristown, NJ, USA, pp. 311–318. Association for Computational Linguistics (2002)
8. Wan, S., Dale, R., Dras, M.: Searching for grammaticality: Propagating dependencies in the viterbi algorithm. In: Proceedings of the Tenth European Workshop on Natural Language Generation (ENLG 2005). Association for Computational Linguistics (2005)

9. Mutton, A., Dras, M., Wan, S., Dale, R.: Automatic evaluation of sentence-level fluency. In: ACL. The Association for Computer Linguistics (2007)
10. Chae, J., Nenkova, A.: Predicting the fluency of text with shallow structural features: Case studies of machine translation and human-written text. In: EACL, pp. 139–147. The Association for Computer Linguistics (2009)
11. Pitler, E., Nenkova, A.: Revisiting readability: A unified framework for predicting text quality. In: EMNLP, pp. 186–195. ACL (2008)
12. Feng, L., Elhadad, N., Huenerfauth, M.: Cognitively motivated features for readability assessment. In: Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009), Athens, Greece, pp. 229–237. Association for Computational Linguistics (2009)
13. Feng, L.: Automatic readability assessment for people with intellectual disabilities. In: SIGACCESS Accessibility and Computing, pp. 84–91 (2009)
14. Toutanova, K., Manning, C.D.: Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In: Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora, Morristown, NJ, USA, pp. 63–70. Association for Computational Linguistics (2000)
15. Toutanova, K., Klein, D., Manning, C.D., Singer, Y.: Feature-rich part-of-speech tagging with a cyclic dependency network. In: NAACL 2003: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, Morristown, NJ, USA, pp. 173–180. Association for Computational Linguistics (2003)
16. Brants, T.: Tnt: a statistical part-of-speech tagger. In: Proceedings of the sixth conference on Applied natural language processing, Morristown, NJ, USA, pp. 224–231. Association for Computational Linguistics (2000)
17. Cormen, T.H., Leiserson, C.E., Rivest, R.L.: Introduction to Algorithms. The MIT press and McGraw-Hill (1990)

Integrating Contrast in a Framework for Predicting Prosody

Pepi Stavropoulou, Dimitris Spiliotopoulos, and Georgios Kouroupetroglou

Department of Informatics and Telecommunications,
National and Kapodistrian University of Athens,
Panepistimiopolis, Ilisia, GR-15784, Athens, Greece
{pepis,dspiliot,koupe}@di.uoa.gr

Abstract. Information Structure (IS) is known to bear a significant effect on Prosody, making the identification of this effect crucial for improving the quality of synthetic speech. Recent theories identify contrast as a central IS element affecting accentuation. This paper presents the results of two experiments aiming to investigate the function of the different levels of contrast within the topic and focus of the utterance, and their effect on the prosody of Greek. Analysis showed that distinguishing between at least two contrast types is important for determining the appropriate accent type, and, therefore, such a distinction should be included in a description of the IS – Prosody interaction. For this description to be useful for practical applications, a framework is required that makes this information accessible to the speech synthesizer. This work reports on such a language-independent framework integration of all identified grammatical and syntactic prerequisites for creating a linguistically enriched input for speech synthesis.

Keywords: Information Structure, Contrast, Prosody Prediction, Speech Synthesis, Annotation Framework.

1 Introduction

It is generally acknowledged that there is a significant interaction between Information Structure (IS) and Prosody. Identifying this interaction is, therefore, very important in the case of practical applications such as speech synthesizers, whereas the quality of the prosody of the utterance greatly determines the overall quality, naturalness and legibility, of the synthetic speech. In addition to the fundamental information-structural partition of the utterance into topic and focus (or theme and rheme, or topic and comment etc. depending on the approach) recent theories [3, 6, 13] identify contrast as a significant IS element claimed to affect accentuation. Furthermore, several researchers [5, 8] propose the existence of different types – or alternatively a hierarchy – of contrast, based on evidence from various languages that grammatically encode different levels of this contrast hierarchy. This paper presents an empirical study of the effect of the various levels of contrast on the prosody of

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 748–758, 2010.

Modern Greek and further discusses the integration of this meta-information for creating a linguistically enriched text description for prosody prediction in speech synthesis into an appropriate framework.

1.1 Theoretical Background

Two-dimensional views of Information Structure identify: (i) a high level partition of the utterance into complementary parts, such as topic and focus, and (ii) a lower level mechanism that functions both within the topic and the focus part of the utterance and is associated with some notion of contrast [6, 13, 16] or givenness [3]. Contrast, in this case, is related to the possibility of different, alternative referents made available by the context, and is marked by a pitch accent as opposed to background material, which remains unmarked. Sentence (1) illustrates this two-level distinction. Prosodically prominent words are capitalized.

(1) What did the tourists want?
The British tourist wanted to rent the blue car. [The ITALIAN$_C$ tourist]$_{TOPIC}$ [wanted to rent the RED$_C$ car]$_{FOCUS}$.

In this more semantically-oriented, quantification-based view of contrast, every focus is contrastive as it triggers the presupposition of a set of alternatives to the focused element. Even in cases of broad focus, one may argue that it is one state of affairs that is contrasted with another [4, 9]. Some researchers, however, combining a more pragmatic or "informational" approach, argue for the existence of different types of contrast, each one of which may be differently encoded in the structure of the language, bearing distinct prosodic, morphological or syntactic correlates. [8] proposes the following criteria for the definition of a hierarchy of contrast (from weaker to stronger): mere *highlighting* through accentuation → existence of a *dominant contrast*, dividing the utterance into a focus and background part → existence of an *open set of alternatives* → existence of a *limited closed set of alternatives* → *explicit mentioning of alternatives* in the context (i.e. existence of a salient directly accessible set). In addition to these criteria, *correction* has been proposed as a special case of contrast that has distinct prosodic markers [5, 6]. It is actually the case that – in some languages at least – only correction as opposed to other sub-notions of contrast is expressed differently.

The different levels of this contrast hierarchy are associated with different types of topics or foci as shown in Table 1. Accordingly, the primary descriptive goal of the study presented here is to examine the prosodic correlates of the different types of topics and foci, ultimately identifying the levels of contrast that are encoded in the prosody of Modern Greek. Furthermore, this study aims to assess the range of interaction between contrast and the topic–focus partition, in order to identify the type of information that should be integrated in a framework for predicting prosody. That is, if the notion of contrast alone is enough to determine accentuation, then it should be formally represented as an autonomous IS feature and there would be no need to resolve to the identification of different types of foci or topics.

Table 1. Association of contrast with different types of topic and focus

	High-lighting	**Dominant Contrast – Open Set of Alternatives**	**Salient Closed Set of Alternatives**	**Correction**
All New / Topic-less Utterances, Broad Information Focus	+	-	-	-
Narrow Information Focus	+	+	-	-
Simple Topic	(+)	+	-	-
Contrastive Focus	+	+	+	-
Contrastive Topic	+	+	+	-
Corrective Focus	+	+	+	+
Corrective Topic	+	+	+	+

2 Experimental Setup

To address these issues two pilot experiments were carried out, the first one investigating the effect that different types of topics have on prosody, and the second one investigating the effect of different types of foci.

2.1 Experiment A - Topics

Three types of topics were tested: simple, contrastive and corrective topics. Sentences (2), (3) and (4) are examples of each type respectively.

(2) What did the Italian tourist want?
[The Italian tourist]$_{ST}$ wanted to rent a car [Simple Topic]
(3) What did the tourists want?
The British tourist wanted to rent a room,
[the ITALIAN tourist]$_{ConT}$ wanted to rent a car [Contrastive Topic]
(4) What did the British tourist want?
[The ITALIAN tourist]$_{CorT}$ wanted to rent a car [Corrective Topic]

All types were compared against all new / topic-less utterances as well. Therefore four pragmatic conditions in total were examined. Test material consisted of 7 utterances per condition. Each utterance was produced twice, once following a narration and once following a Q/A disambiguating context. All utterances were produced by 9 speakers of Athenian Greek resulting in 504 (4x7x2x9) tokens in total. Speakers read the material in random order. Topics were sentence-initial, one and two content-word phrases. To avoid topic accommodation in all new sentences, a generic

version of the utterances was used for the no topic condition; that is an indefinite noun phrase was used instead of a definite one, as definitiveness is often assumed to signal knowledge already present in the hearer's knowledge store.

The four pragmatic conditions were compared on the basis of both phonological and phonetic criteria. In the first case, utterances were annotated for pitch accent type based on the GRToBI annotation scheme [1]. In the second case, measurements were taken of mean F0 (vowel), vowel duration and mean intensity (vowel). Statistical significance was tested using chi-square tests and ANOVAs for phonological and phonetic values respectively.

2.2 Experiment B - Foci

As in the case of topics, 4 pragmatic conditions were tested for focus as well: all new sentences, (narrow) information focus, contrastive focus and corrective focus. Sentences (5)-(8) are examples of the focus types examined.

(5) (What's going on?)
The mailman is looking for HELEN [Broad Focus-All New]
(6) Who is the mailman looking for?
The mailman is looking for $\text{HELEN}_{\text{InfF}}$ [Information Focus]
(7) Who is the mailman looking for? Michael or Helen?
The mailman is looking for $\text{HELEN}_{\text{ConF}}$ [Contrastive Focus]
(8) The mailman is looking for Michael
(No), the mailman is looking for $\text{HELEN}_{\text{CorF}}$ [Corrective Focus]

Seven sentences per condition were embedded in disambiguating contexts to be produced in random order by 5 speakers of Athenian Greek, resulting in a total of 140 (4x7x5) tokens. Focus phrases were always sentence final. Materials were annotated for pitch accent type, and measurements of mean F0 (vowel), vowel duration and mean intensity (vowel) were taken and subjected to analysis of variance. Chi-square tests were used to calculate the effect on pitch accent.

3 Results

The L+H* pitch accent was the predominant choice for both corrective topic and corrective focus. The L* and H* were the accents most commonly used for the remaining types of topic and focus respectively. Figure 1 shows the distribution of nuclear pitch accents over the four pragmatic conditions examined in experiments A and B. Accent distribution proved to be statistically significant for all speakers in the case of topics ($p<0.0005$) and for all speakers (ranging from $p<0.001$ to $p<0.008$ depending on speaker) but one ($p<0.634$) in the case of foci. That one speaker resorted to an emphatic rendition for all utterance types.

Moreover, corrective topics were uttered with increased intensity, duration and F0. All dependent variables showed statistically significant effect ([$F(3)=47.825$, $p<0.0005$], [$F(3)=23.505$, $p<0.0005$], [$F(3)=417.944$, $p<0.0005$] for mean intensity, duration, and mean F0 respectively). Post hoc Turkey tests revealed that only

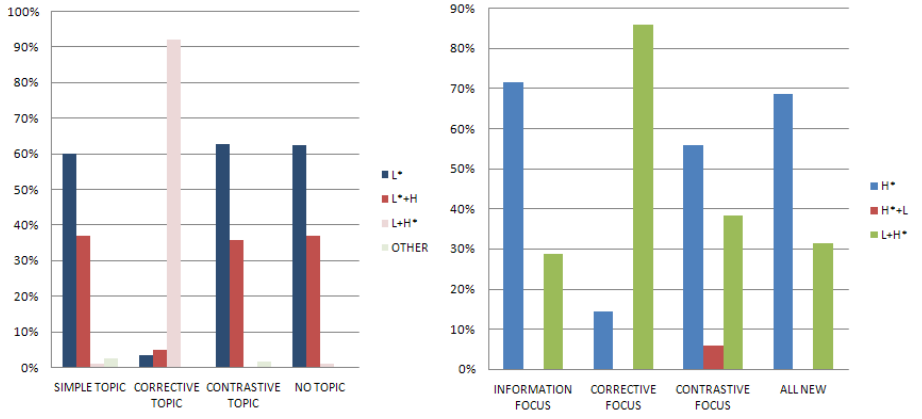


Fig. 1. Distribution of pitch accents over topic and focus types

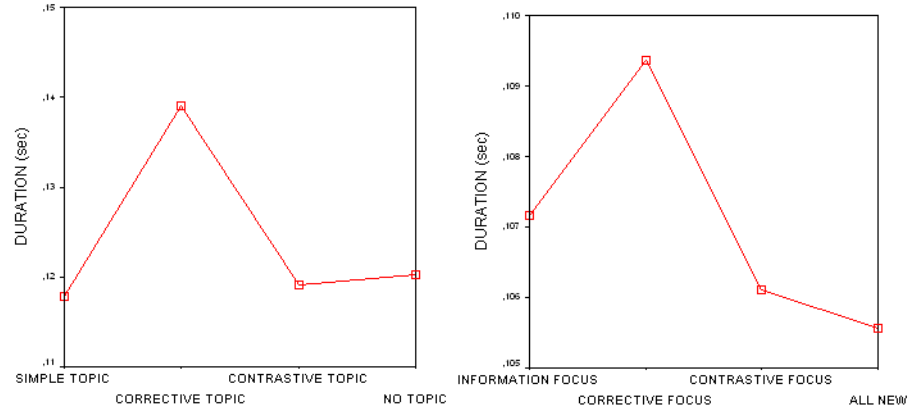


Fig. 2. Mean Vowel Duration for different topic and focus types

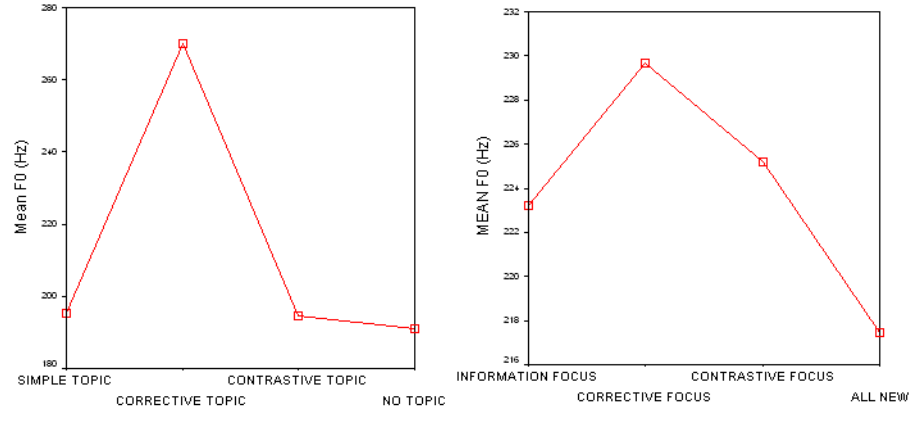


Fig. 3. Mean F0 (vowel) for different topic and focus types

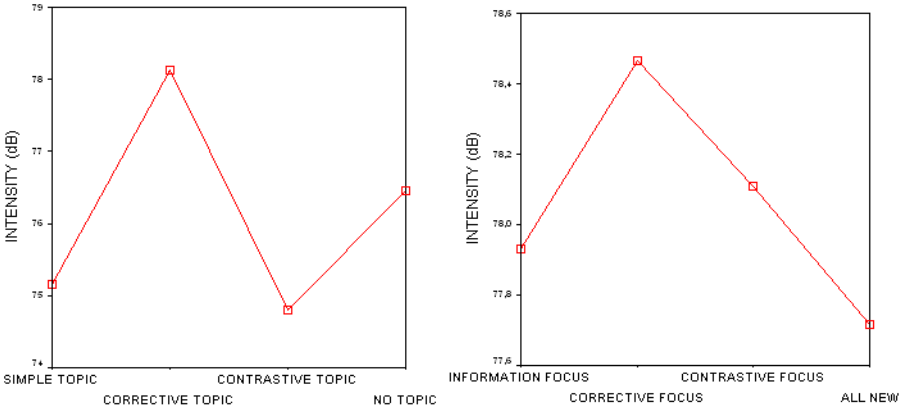


Fig. 4. Mean intensity (vowel) for different topic and focus types

corrective topics significantly differed in pair-wise comparisons, except for the case of intensity, whereas topic-less phrases also differed. In the case of focus, on the other hand, only F0 differed with marginal statistical significance [$F(3)=1756$, $p<0.018$]. Figures 2-4 summarize the results.

4 Discussion

The results of the experiments presented here show that only corrective topics and foci are clearly and consistently distinguished from the other three conditions on the basis of both phonological (L+H* pitch accent) and phonetic (increased intensity, duration, F0 for topics, and F0 for foci) properties. Therefore, Greek only seems to mark correction – with regards to intonation at least – as opposed to other levels of contrast. This does not come as a surprise, as – from an "informational" point of view [15] – correction is the most cognitively loaded procedure, involving subtraction as well as addition of information to the hearer's knowledge store. Similar behavior has been observed in several languages, whereas only corrective focus – as opposed to other types of foci – has distinct phonological correlates, and is therefore structurally contrastive [5]. Moreover, correction is associated with the feature of exhaustivity [7] (i.e. the identification of a unique and maximal subset from the set of alternatives, for which subset only, the predicate phrase actually holds), which in turn has been associated with identificational focus [17]. Identificational focus is an additionally marked case of focus as, on top of being contrastive, is exhaustive as well.

Furthermore, analysis showed that the same nuclear pitch accent (NPA) was used for corrective topic as well as corrective focus, suggesting that the marked effect of correction is independent of the topic-focus articulation, at least with regards to the type of NPA employed. The significant increase in duration and intensity that was observed for corrective topics only, could be explained on the basis of their sentence-initial position (cf. [10]) rather than as being a reflection of topichood. In short, one

could argue that it is not corrective topic or focus per se that is expressed differently, but that the difference is due to the low-level contrast feature that functions within both topic and focus and that topichood or focusing do not determine accent type in contrast to what has been suggested in the literature [13, 14]. The above argument is corroborated by the fact that previous work [2] has shown that, for Greek, the tonal pattern for topic in declaratives is the same as the tonal pattern for focus in interrogatives and vice versa, suggesting that it is the boundary tone that "selects" NPA type, ultimately associating the latter to the discourse role of the former, further disassociating NPA type from topichood or focusing. Similarly, our analysis showed that the contour used for all new phrases was the same for simple and contrastive topics, further supporting the claim that it is not the topic-focus distinction that is conveyed through pitch accent type. As a result, the L* and H* accents that were the predominant choice for the remaining types of topic and focus in our corpus cannot be considered as a constant marker of topichood or focusing.

Even though only correction, compared to other types of contrast, seems to be able to determine the NPA type, identifying what is contrastive in the broad semantically oriented view of contrast, is still necessary in order to define the location of the nuclear pitch accent. That becomes clear in the case of deaccenting, whereas the word which distinguishes the focused element from other alternatives carries the Nuclear Pitch Accent causing all following words to surface de-accented. In some models of Information Structure [3, 11, 12], this function of contrast is ascribed to the function of givenness, whereas a given element is informally defined as an element that has been previously mentioned or can be entailed from another previously mentioned constituent. The prosodic effect is the same, whether it is alternative entities that are distinguished or new vs. given elements. In a similar vein, [12] proposes the postulation of two different features, G and F, in the syntactic representation of the utterance, which correspond to givenness and contrast respectively. It is claimed that the combination of these two features can adequately describe different, structurally motivated types of topics or foci.

In the following section, we will present a markup framework for prosody prediction, whereas pragmatic contrast – i.e. correction in the case of Greek – is represented as an autonomous feature and semantic contrast in the broad sense is conveyed through the given-new distinction. It should be noted that while correction seems to be the minimum pre-requisite for contrastive marking in Greek, other languages may still be "structurally sensitive" to other levels lower in the contrast hierarchy.

5 Integration to an Annotation Framework

Speech synthesizers traditionally perform a part-of-speech analysis and build the syntactic tree of the text in order to assign prosody [18]. General purpose Text-to-Speech (TtS) systems use certain language processing subsystems, such as sentence segmentation and part-of-speech tagging, for the analysis of the written text input.

Depending on the actual system, such analysis may suffer from inherent statistical error accuracy that may be due to the design and implementation of the respective modules or language ambiguity. However, TtS systems may employ language analysis modules that are designed for high accuracy in specific thematic domains for which they seem to perform adequately. The respective accuracy when used for generic or other thematic domains may fall under unacceptable levels. Additionally, the language processing modules embedded in TtS systems are not usually designed to identify and extract higher-level linguistic information, such as semantic or pragmatic factors, that may be used to aid speech synthesis.

Previous works that have explored prosody and speech synthesis show that linguistically enriched annotated text input to a speech synthesizer can lead to improved naturalness of speech output [19, 20]. Generation of tones and prosodic phrasing from high level linguistic input produces better prosody than plain texts do [21]. When such input can be provided, the language processing from the TtS system can be superseded. In this respect, integrating contrast into a framework for language analysis and semantic annotation is important in order to produce an enriched text description as input for speech synthesizers. Text annotation is a procedure where certain meta-information gets identified and associated with the entities in a text corpus. Such information is commonly used in computational linguistics for language analysis, speech processing, natural language processing, speech synthesis, and other areas. The type of information that is analyzed and associated to text units may span the linguistic analysis tree (grammatical, syntactic, morphological, semantic, pragmatic, phonological, phonetic), as well as include any other description that may be of use.

Existing frameworks included the feature and annotation of *contrast* as a process rule [22]. The other features that are currently used for determining the intonational focus prominence include *newness (new or old information)*, *explicit emphasis*, *first or second argument to verb, proper- or common-noun*. Extending that description, based on the aforementioned results, contrast may be included as two distinct features, each providing a more accurate respective prosodic manipulation. Consider the following sentences taken from [22]:

(9) This exhibit was made$_{New}$ in Beotea$_{New}$.
[It was found$_{New}$ in Beotea$_{Giv}$ but it was made$_{Giv}$ in Athens$_{New}$]$_{CONSTRAST}$

The analysis of the corrective vs informational contrast dictates that the contour of sentence 9 should be treated differently to the prototypical contour of the corresponding all-new sentence. Focus prominence and pitch accent prediction shifts from the proper-noun “Beotea” to the verb “found” in the first clause, and proper-noun “Athens” receives special emphasis when corrective contrast is introduced. Providing a distinction between corrective and all other types of contrast, the annotation of this feature can result in proper prosody prediction of those instances. Informational contrast can be described by the newness factor while corrective should be a distinct feature. For Greek, as a generalisation rule, contrast is used for correction while all other instances are described by association with new/given information feature.

```
<utterance>
<relation name="Word" structure-type="list">
<wordlist>
<w id="w01">It</w>
<w id="w02">was</w>
<w id="w03">found</w>
<w id="w04">in</w>
<w id="w05">Beotea</w>
<w id="w06">but</w>
<w id="w07">it</w>
<w id="w08">was</w>
<w id="w09">made</w>
<w id="w10">in</w>
<w id="w11" punct=".">Athens</w>
</wordlist>
</relation>
<relation name="Group" structure-type="list">
</relation>
<relation name="Syntax" structure-type="tree">
<elem phrase-type="S">
<elem phrase-type="prosody" event="contrast">
<elem lex-cat="PRONOUN" href="#w01"/>
<elem lex-cat="AUX" href="#w02"/>
<elem phrase-type="prosody" newness="true" class="mid-emphasis-verb">
<elem lex-cat="VERB" href="#w03"/>
</elem>
<elem lex-cat="PREPOS" href="#w04"/>
<elem phrase-type="prosody" newness="false" arg="arg2" class="proper-
noun">
<elem lex-cat="NOUN" href="#w05"/>
</elem>
<elem phrase-type="prosody" class="mid-emphasis-conj">
<elem lex-cat="CONJNCT" href="#w06"/>
</elem>
<elem lex-cat="PRONOUN" href="#w07"/>
<elem lex-cat="AUX" href="#w08"/>
<elem phrase-type="prosody" newness="false", class="mid-emphasis-verb">
<elem lex-cat="VERB" href="#w09"/>
</elem>
<elem lex-cat="PREPOS" href="#w10"/>
<elem phrase-type="prosody" newness="true" arg="arg2" class="proper-
noun">
<elem lex-cat="NOUN" href="#w11"/>
</elem>
</elem>
</elem>
</relation>
</utterance>
```

Fig. 5. The XML description

Figure 5 shows the XML output for the sentence *"It was found in Beotea but it was made in Athens"* as annotated within the framework and exported to XML. First part is a wordlist of all tokens (words) and punctuation values (<*wordlist*>), followed by the syntax tree, prosodic features, and other high-level information (<*relation*>). This is the input for the speech synthesizer that contains meta-information about how

contrast is assigned as a property of the whole phrase and is subsequently associated with the particular new word within the sentence.

6 Conclusion

The empirical evidence presented in this paper favors the postulation of two different types of contrast as predictors for prosody generation. The two types are associated with a semantic view of contrast, whereas all utterances are in a broad sense contrastive, and a pragmatic one respectively. The latter is a feature of certain utterances only that fulfill specific conditions. The minimum conditions required are subject to typological parameterization, as different languages may express different levels of pragmatic contrast. Greek in particular seems to be sensitive to correction, the level with the highest cognitive load. In the text processing framework described here semantic contrast is accommodated through the given-new distinction and pragmatic contrast is represented as an additional autonomous feature.

Acknowledgments

The work described in this paper has been funded by the Special Account for Research Grants of the National and Kapodistrian University of Athens under the KAPODISTRIAS program.

References

1. Arvaniti, A., Baltazani, M.: Intonational Analysis and Prosodic Annotation of Greek Spoken Corpora. In: Jun, S.-A. (ed.) Prosodic Typology: The Phonology of Intonation and Phrasing, pp. 84–117. Oxford University Press, Oxford (2005)
2. Baltazani, M., Jun, S.-A.: Focus and topic intonation in Greek. In: Proceedings of the 14th International Congress of Phonetic Sciences, vol. 2, pp. 1305–1308 (1999)
3. Büring, D.: Semantics, Intonation and Information Structure. In: Ramchad, G., Reiss, C. (eds.) The Oxford Handbook of Linguistic Interfaces. Oxford University Press, Oxford (2007)
4. Dretske, F.J.: Contrastive statements. The Philosophical Review 81, 411–437 (1972)
5. Gussenhoven, C.: Types of Focus in English. In: Lee, C., Gordon, M., Büring, D. (eds.) Topic and Focus: Cross-linguistic Perspectives on Meaning and Intonation, pp. 83–100. Springer, Heidelberg (2007)
6. Krifka, M.: Basic notions of information structure. In: Fery, C., Krifka, M. (eds.) Interdisciplinary Studies of Information Structure, Potsdam, vol. 6 (2007)
7. Van Leusen, N., Kalman, L.: The Interpretation of Free Focus. In: ILLC Computational Linguistics (1993)
8. Molnár, V.: Contrast from a contrastive perspective. In: Kruiff- Korbayová, I., Steedman, M. (eds.) ESSLLI 2001 Workshop on Information Structure, Discourse Structure and Discourse Semantics (2001)
9. Rooth, M.: A Theory of Focus Interpretation. Natural Language Semantics 1, 75–116 (2001)

10. Rump, H., Collier, R.: Focus Conditions and the prominence of pitch-accented syllables. Language & Speech 39, 1–17 (1996)
11. Schwarzschild, R.: GIVENness, AvoidF and Other Constraints on the placement of Accent. Natural Language Semantics 7(2), 141–177 (1999)
12. Selkirk, E.: Contrastive Focus, Givenness and the Unmarked Status of "Discourse-New". In: Féry, C., Fanselow, G., Krifka, M. (eds.) Working Papers of the SFB632, Interdisciplinary Studies on Information Structure (ISIS), vol. 6, pp. 125–146. Universitätsverlag Potsdam, Potsdam (2007)
13. Steedman, M.: Information structure and the syntax-phonology interface. Linguistic Inquiry 31, 649–689 (2000)
14. Steedman, M.: Information-Structural Semantics of English Intonation. In: Gordon, M., Büring, D., Lee, C. (eds.) LSA Summer Institute Workshop on Topic and Focus, Santa Barbara, pp. 245–264. Kluwer Academic, Dordrecht (2002)
15. Vallduví, E.: The Informational Component. Garland Publishers, New York (1992)
16. Vallduví, E., Vilkuna, M.: On Rheme and Kontrast. In: Culicover, P., Wagner, M. (eds.) Givenness and Locality. The Limits of Syntax, pp. 79–108. Academic Press, San Diego (1998)
17. Kiss, K.E.: Identificational focus versus information focus. Language 74, 245–273 (1998)
18. Taylor, P., Black, A., Caley, R.: The architecture of the festival speech synthesis system. In: Proc. 3rd ESCA Workshop on Speech Synthesis, Australia, pp. 147–151 (1998)
19. Pan, S., McKeown, K., Hirschberg, J.: Exploring features from natural language generation for prosody modeling. Computer Speech and Language 16, 457–490 (2002)
20. Xydas, G., Spiliotopoulos, D., Kouroupetroglou, G.: Modeling Improved Prosody Generation from High-Level Linguistically Annotated Corpora. IEICE Trans. of Inf. and Syst., Special Section on Corpus-Based Speech Technologies 88(3), 510–518 (2005)
21. Black, A., Taylor, P.: Assigning intonation elements and prosodic phrasing for English speech synthesis from high level linguistic input. In: Proc. 3rd Int. Conf. on Spoken Language Processing, Yokohama, Japan, pp. 715–718 (1994)
22. Spiliotopoulos, D., Petasis, G., Kouroupetroglou, G.: A Framework for Language-Independent Analysis and Prosodic Feature Annotation of Text Corpora. In: Sojka, P., Horák, A., Kopeček, I., Pala, K. (eds.) TSD 2008. LNCS (LNAI), vol. 5246, pp. 517–524. Springer, Heidelberg (2008)

Author Index