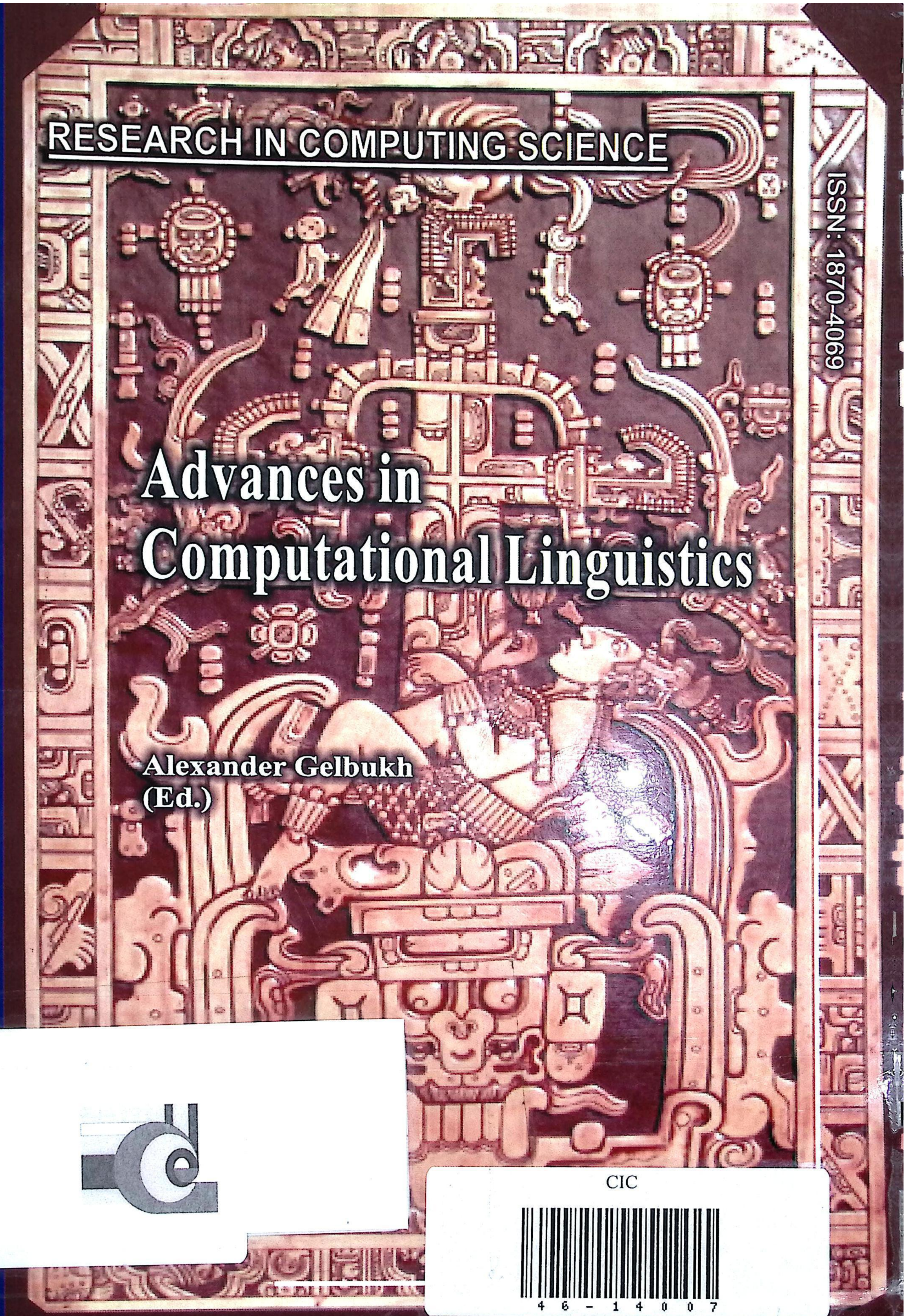


RESEARCH IN COMPUTING SCIENCE

Advances in Computational Linguistics

Alexander Gelbukh
(Ed.)

ISSN: 1870-4069

Advances in Computational Linguistics

Research in Computing Science

Research in Computing Science es una publicación trimestral, de circulación internacional, editada por el Centro de Investigación en Computación del IPN, para dar a conocer los avances de investigación científica y desarrollo tecnológico de la comunidad científica internacional. **Volumen 41** Febrero, 2009. Tiraje: 500 ejemplares. *Certificado de Reserva de Derechos al Uso Exclusivo del Título* No. 04-2004-062613250000-102, expedido por el Instituto Nacional de Derecho de Autor. *Certificado de Licitud de Título* No. 12897, *Certificado de licitud de Contenido* No. 10470, expedidos por la Comisión Calificadora de Publicaciones y Revistas Ilustradas. El contenido de los artículos es responsabilidad exclusiva de sus respectivos autores. Queda prohibida la reproducción total o parcial, por cualquier medio, sin el permiso expreso del editor, excepto para uso personal o de estudio haciendo cita explícita en la primera página de cada documento. Imagen de la portada: http://tierramayaimports.com/images/PL3terraA.jpg. Impreso en la Ciudad de México, en los Talleres Gráficos del IPN – Dirección de Publicaciones, Tres Guerras 27, Centro Histórico, México, D.F. Distribuida por el Centro de Investigación en Computación, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, México, D.F. Tel. 57 29 60 00, ext. 56571.

Editor Responsable: *Juan Humberto Sossa Azuela, RFC SOAJ560723*

***Research in Computing Science** is published by the Center for Computing Research of IPN. **Volume 41**, February, 2009. Printing 500. The authors are responsible for the contents of their articles. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without prior permission of Centre for Computing Research. Printed in Mexico City, February, 2009, in the IPN Graphic Workshop – Publication Office.*

Volume 41
Volumen 41

Advances in Computational Linguistics

Volume Editor:
Editor de Volumen

Alexander Gelbukh

Instituto Politécnico Nacional
Centro de Investigación en Computación
México 2009

ISSN: 1870-4069

Instituto Politécnico Nacional (IPN)
Centro de Investigación en Computación (CIC)
Av. Juan de Dios Bátiz s/n esq. M. Othón de Mendizábal
Unidad Profesional "Adolfo López Mateos", Zacatenco
07738, México D.F., México

http://www.ipn.mx
http://www.cic.ipn.mx

The editors and the Publisher of this journal have made their best effort in preparing this special issue, but make no warranty of any kind, expressed or implied, with regard to the information contained in this volume.

Indexed in LATINDEX and Periodica / Indexada en LATINDEX y Periodica

Printing: 500 / Tiraje: 500

Printed in Mexico / Impreso en México

Preface

Computational linguistics is an interdisciplinary research area that combines the ideas and methods of linguistics, computer science, and artificial intelligence and has twofold goal: on the one hand, to study human language by means of modern computational methods, and on the other hand, to develop computer programs capable of human-like activities related to understanding or producing texts or speech in human language, such as English or Chinese.

The most important technical applications of computational linguistics include information retrieval and information organization, machine translation, and natural language interfaces, among others. However, as in any science, the activities of the researchers are mostly concentrated on its internal art and craft, that is, on the solution of the problems arising in analysis or synthesis of natural language text or speech, such as syntactic and semantic analysis, disambiguation, or compilation of dictionaries and grammars necessary for such analysis.

This volume presents 25 original research papers written by 64 authors representing 23 different countries: Algeria, Brazil, Canada, China, Czech Republic, France, Germany, Greece, Hong Kong, India, Islamic Republic of Iran, Italy, Jordan, Lithuania, Macao, Mexico, Portugal, Russian Federation, Spain, Switzerland, Turkey, United Kingdom, and United States. The volume is structured in 8 thematic areas of both theory and applications of computational linguistics:

- Computational lexicography and lexical resources
- Morphology and syntax
- Semantics
- Anaphora and co-reference
- Text classification
- Text summarization
- Speech generation
- Applications

The papers included in this volume were selected on the base of rigorous international reviewing process out of 65 submissions considered for evaluation; thus the acceptance rate of this volume was 38%.

I would like to cordially thank all people involved in the preparation of this volume. In the first place I want to thank the authors of the published paper for their excellent research work that gives sense to the work of all other people involved, as well as the authors of rejected papers for their interest and effort. I also thank the members of the Editorial Board of the volume and additional reviewers for their hard work on reviewing and selecting the papers. I thank Sulema Torres, Ignacio Garcia Araoz, Oralia del Carmen Pérez Orozco, and Raquel López Alamilla for their valuable collaboration in preparation of this volume. The submission, reviewing, and selection process was supported for free by the EasyChair system, www.EasyChair.org.

Alexander Gelbukh

February 2009

Table of Contents
Índice

Page/Pág.

Computational Lexicography and Lexical Resources

Morphology and Syntax

Semantics

Anaphora and Co-reference

Text Classification

Text Summarization

Speech Generation

Applications

Computational Lexicography and Lexical Resources

Scaling to Billion-plus Word Corpora

Jan Pomikálek[1], Pavel Rychlý[1] and Adam Kilgarriff[2]

[1] Masaryk University, Brno, Czech Republic
[2] Lexical Computing Ltd, Brighton, UK

Abstract. Most phenomena in natural languages are distributed in accordance with Zipf's law, so many words, phrases and other items occur rarely and we need very large corpora to provide evidence about them. Previous work shows that it is possible to create very large (multi-billion word) corpora from the web. The usability of such corpora is often limited by duplicate contents and a lack of efficient query tools.
This paper describes BiWeC, a Big Web Corpus of English texts currently comprising 5.5b words fully processed, and with a target size of 20b. We present a method for detecting near-duplicate text documents in multi-billion-word text collections and describe how one corpus query tool, the Sketch Engine, has been re-engineered to efficiently encode, process and query such corpora on low-cost hardware.

1 Introduction

There's no data like more data, and one place to get more data almost without limit (for general English and some other languages and varieties) is the web. One way to use the web is to create a local corpus by downloading web pages: in [1] we argue that it is the optimal way to use the web for linguistic research. A number of corpora have been built in this way: Baroni and colleagues developed web corpora with nearly 2 billion words for German, Italian and English [2, 3] and have made them available for research as tar archives. Liu at al. [4] describe the creation of a 10 billion word corpus. In this paper we introduce BiWeC, a Big Web Corpus currently of 5.5b words, with a target size of 20b.

Very large corpora can be created on low cost hardware in a few person-months. Most of the steps have linear complexity and scale up well. The two outstanding issues we focus on in this paper are:

1. removing duplicate content
2. efficient querying.

The article is organized as follows. In section two our motivation for creating larger corpora is discussed and the advantages of using more data for various tasks is explained. Section three describes the process of creating BiWeC, with a focus on removing duplicate and near-duplicate documents. Section four deals with corpus processing and querying using the Sketch Engine corpus manager. Then we present some figures about BiWeC and outline future plans.

© A. Gelbukh (Ed.)
Advances in Computational Linguistics.
Research in Computing Science 41, 2009, pp. 3-14

Received 03/11/08
Accepted 08/12/08
Final version 04/02/09

2 Motivation

Bigger corpora provide more information. To illustrate why a BNC[3]-sized corpus is often not enough we take a sample of headwords from one page of Macmillan English Dictionary [5] and compare their frequencies in the 50,000 word Suzanne corpus, the 100m word BNC, and BiWeC (Big Web Corpus), our new web corpus containing 5.5 billion words of general English.

Table 1. Word frequencies comparison for Susanne, BNC and BiWeC

	Susanne	BNC	BiWeC	BiWeC/BNC
Size [m words]	0.15	111	5,500	49×
heavy (adj)	11	9,089	252,305	27×
hector (v)	0	37	956	25×
hedge (n)	0	1,525	19,526	12×
hedonism (n)	1	63	1,757	27×
heebie-jeebies	0	0	151	-

Corpora like Susanne are not usable for lexical knowledge, BNC is good for high frequency words but scarcely provides enough information to make informed generalisations on *hector* (verb) and certainly does not for *heebie-jeebies.* BiWeC provides ample evidence in both cases. The issue is more acute still for phrasal and collocational items.

One common role for corpora is exploration of variation in language, according to, for example, genre, domain or region, or over time. To support such studies, a corpus should be big enough to have subcorpora where the expected frequency for the item being studied is at least thirty or forty. Here, the BNC is often too small. Consider DOMAIN: the written part of the BNC can be divided into ten broad domains, with subcorpus sizes between three and nineteen million words. So a word like *hedonism* has an expected frequency of just two in the smallest of these subcorpora, and there is not enough data in the BNC to support a study of the use of the word across domains.

As we make more and more use of corpora, so the merits of having ample data even for non-core phenomena are more and more evident. In recent work we have developed GDEX, a "good dictionary example extractor" [6]. We aim to find good candidate sentences to be used to exemplify collocations in dictionaries and language teaching. So, we gather all the example sentences in the corpus for a given collocation, reject those that have undesirable features (such as non-words, 'bad' words, many long words, excessive punctuation, or where the sentences are long) and choose the most desirable of the remainder (preferring sentences that are short, contain common words, appear to have standard grammar and contain other words typical of the settings in which the collocation is found). If there

[3] British National Corpus, see `http://natcorp.ox.ac.uk/`, comprising 100m words.

was a set of 100 sentences to start with, we are likely to find a good candidate for a dictionary or teaching example. If we only had five sentences to start with, we are unlikely to.

2.1 Relations with Google

One response to the argument for size above is "why not, then, use the web via Google". This is an entirely pertinent response: Google is a spectacularly fast query engine looking at a spectacularly large corpus. For Google, the 'local corpus' is as much of the web as it has succeeded in indexing. (It would like to index the whole web, but in the course of its crawling it does not find everything. It probably finds a greater proportion of everything than any of its competitors: for a review of these issues and discussion on index sizes see [7].) Google, Yahoo and other web search engines share many of the concerns of linguistic corpus developers and corpus tool developers. They want large corpora, with wide coverage, with duplication intelligently handled and with spam weeded out. They want to index on terms in the content of the page rather than in navigation bars and advertisements. 'Text-heavy' pages and pages that have lots of readers are or highest value to them. They would like to operate (by default) on lemmas; word class information would be useful to them. While the goals are distinct, since search engines help people find out about things whereas corpus research looks at the language denoting the things, the route to those goals is often shared.

The Google-indexed web is far larger than any corpus developed for linguists: in Table 2 we repeat Table 1 with Google counts added.

Table 2. Comparing corpus frequencies with the web via Google. Note that: Google hit counts are for document counts not word counts; they are for word forms not lemmas and are not word-class-specific; and they can vary from one hour or day to the next. The searches were undertaken with default settings except that the language was set to English and 'allintext' was set so that the search term had to be in the text rather than, for example, in a link. The differences in what is being counted probably account for the high ratio for *hector*, which often occurs as a name.

	Susanne	BNC	BiWeC	BiWeC/BNC	Google	Google/BiWeC
Size [tokens]	.15m	111m	5 500m	49×		
heavy (adj)	11	9 089	252 305	27×	242.0m	955×
hector (v)	0	37	956	25×	22.1m	23,000×
hedge (n)	0	1 525	19 526	12×	25.8m	1,321×
hedonism (n)	1	63	1 757	27×	2.4m	1,343×
heebie-jeebies	0	0	151	-	20,900	138×

A cluster of researchers have used Google and other search engines in this way, see e.g. [8]. The great benefit is the size. Where the phenomena under investigation are too rare to have a reasonable number of hits in 5.5b words of English, this is currently the only course available. But there are costs and problems relating to the strategy [1]:

- the query language is limited
- no searching for lemmas or by word class or other linguistic parameters
- search hits are ordered in a very specific manner which is not relevant to linguistic research and cannot (in Google) be turned off
- searches are limited: over-users of Google may have their access blocked, and also a maximum of 1000 hits are given per query
- methods are not published
- Google may change methods and the query language, without saying they are doing so or offering any explanation how or why, at any point.

For these reasons, wherever the corpus is big enough to provide enough data to support the research question, we advocate the use of a corpus prepared using the methods described here and loaded into a query tool specialised for linguistic research. So – the bigger the corpus, the more research questions can be addressed without recourse to Google or other commercial search engines, with all the associated disadvantages.

3 Building BiWeC

3.1 Crawling

We retrieve textual data from the web using procedures similar to [2, 3]. We used the Heritrix web crawler developed by the Internet Archive[4] and started the crawl from the same list of URLs as Ferraresi et al.[5] The crawl was restricted to domains considered likely to contain English texts, such as `.uk` or `.au`. For practical reasons we only processed HTML pages (`content-type: text/html`). We also applied restrictions to the size of the processed documents and filtered out all web pages smaller than 5 kB or larger than 2 MB. The rationale is that pages smaller than 5 kB contain little if any textual content and pages larger than 2 MB are usually not what we want but are logfiles, lists, catalogues or similar. Unlike Ferraresi et al. we configured the web crawler to perform this basic filtering for us rather than storing all crawled data and filtering as a postprocessing step, so reducing bandwidth requirements and disk space.

3.2 Cleaning

'Cleaning' the retrieved web pages involves stripping out the HTML mark-up and removing content such as navigation links, copyright notices, advertisements, etc. To perform this step we considered the participating systems in the CleanEval

[4] `http://www.archive.org/`

[5] We would like to thank the ukWaC corpus team for providing us with the list of seed URLs.

competition for cleaning webpages [9]. However, our experiments revealed that even the winning system of CleanEval is matched by the BTE algorithm [10].[6]

BTE works from the observation that the material we wish to remove is usually rich in markup. It establishes the ratio of text to markup for different chunks of the page. The ratio is most often high at the beginning and end of the page and lower in the middle. BTE retains only that part of the page where the ratio is low.

3.3 Character Encoding Conversion

To unify the character encoding of the corpus contents we converted all downloaded pages to UTF-8. As long as we only processed HTML pages, we were able to determine the original character encoding from the HTML headers. We are well aware that the header information is not always reliable and that techniques exist for guessing the character encoding from textual contents of a web page. However, for English, occasional errors in character encoding conversion do not cause problems. Therefore we considered using any advanced encoding detection techniques unnecessary.

3.4 Language Detection

Language detection was performed to filter out non-English documents which occasionally occur in the crawled domains. The problem of language detection has been well described in the literature (e.g. [11]) and many freely available language detection systems exist. Having the postprocessing procedure fully implemented in Python, we chose the Trigram class[7] which performs the language identification based on frequencies of triples of characters. As a by-product of the language detection, noisy texts were also filtered out, such as documents full of JavaScripts which the cleaning phase failed to remove.

3.5 Removing Duplicate and Near-duplicate Documents

Duplicate documents in text corpora do damage to corpus derived statistics. Much corpus use is based around identifying patterns which are much more common that one would expect by chance. For example, collocation studies are premised on a word and its collocate appearing together remarkably often. Collocation-finders are repetition-spotters. If a corpus contains many duplicate texts, then supposed collocation lists will often have contents that result, not from a *bona fide* collocation, but from the duplication. Because much corpus research is regularity-spotting, it is easily derailed by the regularities provided by duplication. Duplicate concordance lines are also an irritant, and potentially

[6] BTE achieved a score of 85.41 on the CleanEval test set in the text-only cleaning task, while the CleanEval winner (Victor) scored 84.07.

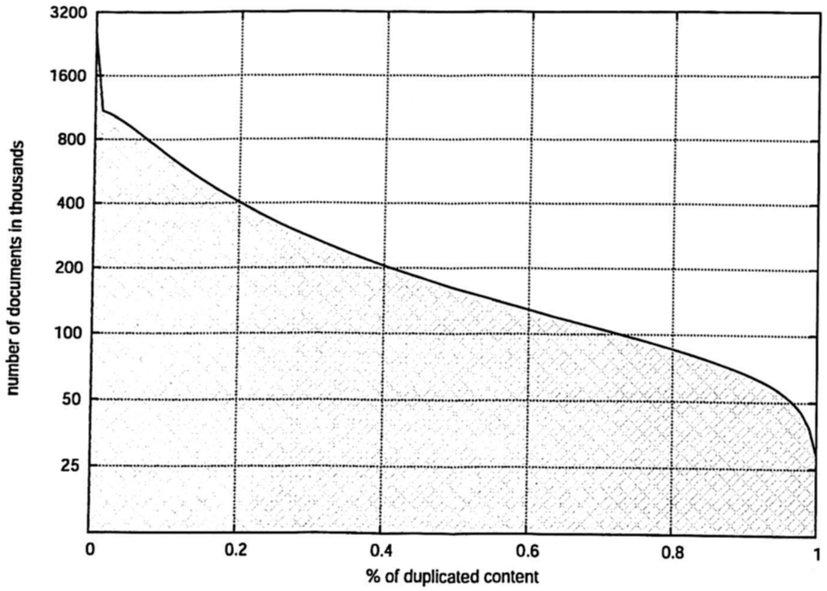
[7] `http://code.activestate.com/recipes/326576/`

misleading, in the manual exploration of corpora. For a high-quality corpus, removing duplicates is essential.

Identical web pages can be easily detected by comparing their checksums. This, however, does not work for near-duplicate web pages as even a small change to the contents changes the checksum.

Liu and Curran [4] describe how they created a 10 billion words corpus but do not discuss duplication. Ferraresi et al. [3] removed near-duplicate documents from the ukWaC corpus using a technique based on Broder's fingerprinting algorithm [12]. This method works well only for very similar documents but does not detect documents which contain both significant identical parts and different parts. This is illustrated in Fig 1. We have analysed ukWaC to establish the duplication it retains, and while it does not include many 100 % duplicates, it does contain large numbers of partial duplicates. Of a total of 2.58m documents, 28,716 have 100 % duplicate content, but 85,693 have at least 80 % duplicate content.

Fig. 1. Duplicate and near-duplicate documents in the ukWaC corpus



Many variations of fingerprinting algorithms for near-duplicate documents exist. These are mostly based on finding shared sequences of words among documents. To make the process feasible for very large data, only small samples of each document are used. This, however, constitutes a loss of data which means that only very close documents are identified, such as a regular web page and a version of it for printing. The technique is commonly used by search engines for grouping near-duplicates but is not so suitable for us since it lets through a substantial amount of duplicate text.

In our previous work [13] we developed a new algorithm for detecting duplicated text, and this is what we have applied in the building of BiWeC. It works as follows.

Let us assume that undesirable duplication occurs wherever a string of n or more words is duplicated (we have used $n = 10$). An exhaustive method of duplicate detection would work with all 10-word strings in the corpus. This is prohibitive for large corpora, but we can reason that we do not need to work with all of them. We only need to find the duplicate 10-word strings.

The core idea of our algorithm is to use an external sort method to generate all n-grams together with their counts directly, in one step (as opposed to the SPEX algorithm [14] which iterates through 3-grams, 4-grams, ... n-grams, removing non-duplicated items at each stage: our earlier paper shows that SPEX does not scale well). The program splits the input text into chunks which fit into a fixed amount of memory. For each chunk, a sorted list of n-grams is generated and saved to a temporary file on disk. The final phase joins all temporary files and outputs any n-gram with total count higher than one. This is a typical external sort method which would require a huge amount of disk space and a lot of runs (chunks) to process the whole input text.

Having the list of duplicate 10-grams available, we can easily determine the amount of duplicated data in each document. This allows us to explore the tradeoff between partial-duplicate-removal and corpus size. We may choose to remove all documents with over 10 % of data duplicated, or over 50 %, or over 90 %. The optimal figure will depend on the sensitivity of the applications and users to duplication, versus the demand for a large corpus. For BiWeC, after some experimentation, we have used a threshold of 50 %. The size of BiWeC before and after de-duplication is shown in Table 3.

Table 3. BiWeC size before and after removing duplicate content

	before deduplication	after deduplication
number of documents	≈ 9 mil.	4.3 mil.
number of tokens	**9 bil.**	**5.5 bil.**
number of different word forms	36 mil.	28 mil.
number of different lemmas	32 mil.	27 mil.

The algorithm does not give rise to a database which can be used to ask, of a new document, does it duplicate material already in the corpus, so is not suitable where a corpus is frequently to be added to. Such a database is provided by fingerprinting algorithms such as Broder's.

In the paper cited above we demonstrate that the algorithm scales well to billions of words. Also the de-duplication was a one-off exercise: if the corpus is regularly to be added to, we shall revisit suitable methods.

3.6 Corpus Annotation

For tagging the corpus we have used TreeTagger [15], a widely-used state-of-the-art part-of-speech tagger based on decision trees. For English, it lemmatizes as well as POS-tags. We have used it with the English tagset and parameters as distributed, and have not re-trained it. We have good experiences of its performance and results over the past seven years: the accuracy has proved to be adequate for many corpus based tasks such as building word sketches [16, 17] and distributional thesauruses. In addition, the TreeTagger is fast, processing about one million words per minute on a single machine: sufficient even for a corpus with several billion words.

4 Query Engines for Multi-Billion-Word Corpora

Many interesting linguistic results can be computed by simple batch processing of the whole corpus. They include word list and bigram (or n-gram) lists with frequencies, collocation lists for a particular word or lemma, lists of words or lemmas which are particularly salient in a particular subcorpus ('keywords') and lists of words particularly associated with a grammatical construction or pattern, such as "nouns with a strong tendency to occur in the plural" [18].

Studying such lists, linguists usually find data which are surprising and need some explanation. This is best done by browsing concordances. Users need to be able to query the corpus interactively as well as in batch mode. This demands fast query evaluation.

Most current computers use 32-bit numbers and even on 64-bit machines the default numeric type in many programming languages is 32-bit only. The maximum number for a signed 32-bit numeric type is $\hat{2}31$, which is slightly more than 2 billion: the computer cannot (readily) count higher than 2 billion. Many off-the-shelf algorithms use the default numeric type and are not prepared for more than 2 billion items. For example, advanced implementations of data processing systems often use *memory-mapping*, which enables mapping of data in files to main memory addresses without a long sequential read of the whole data file. The operating system is responsible for reading the respective data block from files to the main memory. This feature can greatly improve the speed of a program and also simplify the implementation. However 32-bit machines cannot memory-map more than 2 GB of data.

While the revision of the code to use 64-bit numbers at all points is not intrinsically a hard technical challenge, there are many fixes that need to be made to cover the code of the whole system.

We have recently re-engineered the Manatee corpus query engine so that it can handle corpora of over 2 billion words [19].[8]

[8] Manatee is incorporated in a few derived products; it is the core engine of the Sketch Engine [17].

4.1 Corpus Encoding

In Manatee, the binary files holding the encoded and indexed corpus are roughly the same size as the input plain or annotated text in Manatee's one-word-per-line input format. Each variety of annotation (e.g., wordform, lemma, POS-tag, sentence-markup) increases the size of the input data and the size of the encoded data in similar ways. The same amount of disk space again is needed for temporary storage during the encoding process. Data sizes are shown in Table 4.

Table 4. BiWeC data sizes (after removing duplicates)

downloaded data size	≈ 1 TB
cleaned data in vertical format (POS-tagged, lemmatised)	72 GB
encoded data	69 GB

The total encoding time depends not only on the number of words in the corpus but also on the amount of annotation.

The first stage of encoding on a standard server take about 10 hours per billion tokens. Encoding of 9.5 billion token corpus ran for four days on a 2 GHz AMD Opteron machine with 2 GB of memory dedicated to the encoding process.

After this stage, the corpus can be queried for concordances, concordance-based functions including frequency distributions, collocations and concordance sorting, and word lists. Additional indexes can be computed to speed up some type of queries (for example ignoring upper and lower case of lemmas). Computation time for such indexes depends on the size of the lexicon, that is, the number of different words in the corpus.

An encoded corpus can be used by users to create concordances and word lists. A powerful query language can be used for queries. We are using the Sketch Engine to build word sketches (one-page, automatic, corpus-derived summary of a word's grammatical and collocational behavior [17]). The usefulness of such information depends on how many instances of grammatical relations we find in the corpus for the given word.

For the BNC there are 7,414 words for which more than 1000 grammatical relation instances were found. In the current fully-processed version of BiWeC, comprising 5.5b tokens, there are 76,689 such words. We have ten times as many words for which we can generate a detailed and thorough account of the word's behavior.

Table 5. BiWeC processing times

computing word sketches	16 hours
computing thesaurus	40 minutes

Table 6. The number of lemmas for which rich, data-driven analyses are available in each of three corpora. The analyses are built on the evidence of the grammatical relations (gramrels) that a word occurs in, and the other words it occurs with. At least one hundred, and preferably several hundred instances are required for a good word sketch.

gramrel hits	>100	>1000
BNC (100m)	29,931	7,414
ukWaC (1,500m)	74,293	21,614
BiWeC (5,500m)	335,294	76,689

5 Conclusions and Future Work

We are able to build multi-billion word corpora which are suitable for linguistic research, and we have tools which can encode and query them efficiently. We have introduced one such corpus, BiWeC, a corpus of general English, currently of 5.5 billion words fully prepared and accessible in our tools (Manatee and the Sketch Engine) and with a target size of 20 billion words.

We have a long agenda for further developments to both the corpus and the tool. Our highest priority for the tool is to address hardware limitations relating to disk access speed. We shall be exploring Amazon's 'cloud computing' which is rumored to offer very fast disk access. In relation to the corpus, in addition to gathering more data, we wish to classify documents using text classification methods along a number of dimensions including at least domain, formality and region. In a companion paper [20] we describe a parallel stream of work in which we are developing a 100m word "New Model Corpus", half of which is taken from BiWeC (with data-driven classification) and the other half taken from web sources which we know to be fiction, or chatshow transcripts, or film transcripts, so we have a corpus which, like the BNC and before it LOB and Brown, supports a variety of studies across language varieties. We also want to explore additional textual markup, including for named entities, time phrases, and semantic categories of nouns: we plan to do this in an open, collaborative model, working with other groups who have tools and expertise in these forms of annotation. The New Model Corpus will be made freely available for research purposes. Our plan is that the two streams should merge, to give a multi-billion word corpus with many interesting and still very large subcorpora and rich textual markup.

Acknowledgements. This work has been partly supported by the Academy of Sciences of the Czech Republic under the projects T200610406, T100300419 and by the Ministry of Education of CR within the Centre of Basic Research LC536 and National Research Programme 2C06009.

References

1. Kilgarriff, A.: Googleology is Bad Science. Computational Linguistics **33**(1) (2007) 147–151
2. Baroni, M., Kilgarriff, A.: Large linguistically-processed web corpora for multiple languages. Proceedings of European ACL (2006)
3. Ferraresi, A., Zanchetta, E., Baroni, M., Bernardini, S.: Introducing and evaluating ukWaC, a very large web-derived corpus of English. In: Proceedings of the 4th Web as Corpus Workshop (LREC 2008). (2008)
4. Liu, V., Curran, J.: Web Text Corpus for Natural Language Processing. EACL. The Association for Computer Linguistics (2006)
5. Rundell, M., Fox, G.: Macmillan English Dictionary: For Advanced Learners. Macmillan (2002)
6. Kilgarriff, A., Husák, M., McAdam, K., Rundell, M., Rychlý, P.: GDEX: Automatically finding good dictionary examples in a corpus. Proceedings of Euralex (2008)
7. Sullivan, D.: End Of Size Wars? Google Says Most Comprehensive But Drops Home Page Count. Search Engine Watch (3551586) (2005)
8. Nakov, P.: Noun compound interpretation using paraphrasing verbs: Feasibility study. In: Proc. 13th International Conference on Artificial Intelligence: Methodology, Systems, Applications (AIMSA'08). (2008)
9. Baroni, M., Chantree, F., Kilgarriff, A., Sharoff, S.: CleanEval: A competition for cleaning webpages. Proceedings of LREC 2008 (2008)
10. Finn, A., Kushmerick, N., Smyth, B.: Fact or fiction: Content classification for digital libraries. In: DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries. (2001)
11. Beesley, K.: Language identifier: A computer program for automatic natural-language identification of on-line text. Language at Crossroads: Proceedings of the 29th Annual Conference of the American Translators Association (1988) 12–16
12. Broder, A.: Identifying and filtering near-duplicate documents. Proceedings of the 11th Annual Symposium on Combinatorial Pattern Matching (2000) 1–10
13. Pomikálek, J., Rychlý, P.: Detecting co-derivative documents in large text collections. Proceedings of LREC 2008 (2008)
14. Bernstein, Y., Zobel, J.: A scalable system for identifying co-derivative documents. In: Proc. String Processing and Information Retrieval Symposium, Padua, Italy (2004) 55–67
15. Schmid, H.: Probabilistic part-of-speech tagging using decision trees. Proceedings of International Conference on New Methods in Language Processing **12** (1994)
16. Kilgarriff, A., Tugwell, D.: WORD SKETCH: Extraction and Display of Significant Collocations for Lexicography. Proceedings of the 39th ACL **10** (2001) 32–38
17. Kilgarriff, A., Rychlý, P., Smrž, P., Tugwell, D.: The Sketch Engine. Proceedings of Euralex (2004) 105–116
18. Kilgarriff, A., Rychlý, P.: Finding the words which are most X. Proceedings of Euralex (2008)
19. Rychlý, P.: Manatee/bonito - a modular corpus manager. In: Recent Advances in Slavonic Natural Language Processing(RASLAN), Masaryk University, Brno (2007) 65–70
20. Kilgarriff, A.: New model corpus. Technical report, Lexical Computing Ltd. (2009 in preparation)

Detecting and Grounding Terms in Biomedical Literature

Kaarel Kaljurand, Fabio Rinaldi, Thomas Kappeler and Gerold Schneider

Institute of Computational Linguistics, University of Zurich
kalju@ifi.uzh.ch, rinaldi@ifi.uzh.ch, kappeler@bluewin.ch,
gschneid@ifi.uzh.ch

Abstract. We present an approach towards the automatic detection of names of proteins, genes, species, etc. in biomedical literature and their grounding to widely accepted identifiers. The annotation is based on a large term list that contains the common expression of the terms, a normalization step that matches the terms with their actual representation in the texts, and a disambiguation step that resolves the ambiguity of matched terms. We describe various characteristics of the terms found in existing term resources and of the terms that are used in biomedical texts. We evaluate our results against a corpus of manually annotated protein mentions and achieve a precision of 57% and recall of 72%.

1 Introduction

The complexity of biological organisms and the success of biological research in describing them, have resulted in a large body of biological entities (genes, proteins, species, etc.) to be indexed, named and analyzed. Probably the most important entities are proteins. They are an essential part of an organism and participate in every process within cells. Most proteins function in collaboration with other proteins, and one of the research goals in molecular biology is to identify which proteins interact.

While the number of different proteins is large, the amount of their possible interactions and combinations is even larger. In order to record such interactions and represent them in a structured way, human curators who work for knowledge base projects, e.g. Molecular INTeraction database (MINT)[1], Human Protein Reference Database (HPRD)[2], IntAct[3] (see [4] for a detailed overview), carefully analyze published biomedical articles. As the body of articles is growing rapidly, there is a need for effective automatic tools to help curators in their work. Such tools must be able to detect mentions of biological entities in the text and tag them with identifiers that have been assigned by existing knowledge bases. As the names that are used to reference the proteins can be very ambiguous, there is a need for an effective ambiguity resolution.

[1] http://mint.bio.uniroma2.it

[2] http://www.hprd.org/

[3] http://www.ebi.ac.uk/intact

© A. Gelbukh (Ed.)
Advances in Computational Linguistics.
Research in Computing Science 41, 2009, pp. 15-26

Received 10/11/08
Accepted 09/12/08
Final version 02/02/09

In this paper, we describe the task of automatically detecting names of proteins, genes, species, and experimental methods in biomedical literature and grounding them to widely accepted identifiers assigned by three different knowledge bases — UniProt Knowledgebase (UniProtKB)[4], National Center for Biotechnology Information (NCBI) Taxonomy[5], and Proteomics Standards Initiative (PSI) Molecular Interactions (MI) Ontology[6].

The term annotation uses a large term list that is compiled on the basis of the entity names extracted from the mentioned knowledge bases. This resulting list covers the common expression of the terms. A term normalization step is used to match the terms with their actual representation in the texts. Finally, a disambiguation step resolves the ambiguity (i.e. multiple IDs proposed by the annotator) of the matched terms.

The work presented in this paper is part of a larger effort undertaken in the OntoGene project[7] aimed at improving biomedical text mining through the usage of advanced natural language processing techniques. The results of the protein detection approach described in this paper feed directly into the process of identification of protein interactions. Our approach relies upon information delivered by a pipeline of NLP tools, including sentence splitting, tokenization, part of speech tagging, chunking, and a dependency-based syntactic analysis of candidate sentences [6]. The syntactic parser takes into account constituent boundaries defined by previously identified multi-word entities. Therefore the richness of the annotation process (including a variety of domain entities) has a direct beneficial impact on the performance of the parser, and thus leads to better recognition of interactions.

This paper is structured in the following way. In section 2 we describe the terminological resources that we have used, in section 3 we describe an automatic annotation of biomedical texts using these resources, in section 4 we describe the evaluation method and results, in section 5 we review related work, and finally, in section 6 we draw conclusions and describe future work.

2 Term Resources

2.1 Introduction

As a result of the rapidly growing information in the field of biology, the research community has realized the need for consistently organizing the discovered information — assign identifiers to biological entities, enumerate the names by which the entities are referred to, interlink different resources (e.g. existing knowledge bases and literature), etc. This has resulted in large and ever-growing knowledge bases (lists, ontologies, taxonomies) of various biological entities (genes, proteins, species, etc.). These resources can be treated as linguistic resources

[4] http://www.uniprot.org

[5] http://www.ncbi.nlm.nih.gov/Taxonomy/

[6] http://psidev.sourceforge.net/mi/psi-mi.obo

[7] http://www.ontogene.org

which can function as the basis of large term lists that can be used to annotate existing biomedical publications in order to identify the entities mentioned in these publications. In the following we describe three resources: UniProtKB, NCBI Taxonomy, and PSI-MI Ontology.

2.2 UniProtKB

The UniProt Knowledgebase (UniProtKB)[8] assigns identifiers to 397,539 proteins and describes their amino-acid sequences. The identifiers come in two forms: numeric accession numbers (e.g. P04637), and mnemonic identifiers that make visible the species that the protein originates from (e.g. P53_HUMAN). In the following we always use the mnemonic identifiers for better readability.

In addition to enumerating proteins, possible names used in the literature to refer to the proteins are listed in UniProtKB. UniProt sees as one of its functions to help with the standardization of protein nomenclature and thus tries to cover all the common ways of referring to a protein[9], while at the same time specifying a single name as "recommended name", following certain naming guidelines[10]. In addition, the names of functional domains and components of proteins, and also names of genes that encode the proteins are provided. The set of names covers names with large lexical difference (e.g. both 'Orexin' and 'Hypocretin' can refer to protein OREX_HUMAN), but usually not names with minor spelling variations (e.g. replacing a space with a hyphen). UniProtKB attempts to cover proteins of all species. The top five species ranked by the number of their different proteins are *Homo sapiens* (Human) with 20,325 proteins, *Mus musculus* (Mouse) with 15,915, *Rattus norvegicus* (Rat) with 7170, *Arabidopsis thaliana* (Mouse-ear cress) with 6970, and *Saccharomyces cerevisiae* (Baker's yeast) with 6553.[11]

We extracted 626,180 (different) names from the UniProtKB XML file, using the XPath expressions listed in table 1. The ambiguity of a name can be defined as the number of different UniProtKB entries that contain the name. UniProtKB names can be very ambiguous. This follows already from the naming guideline which states that "a recommended name should be, as far as possible, unique and attributed to all orthologs"[12]. Thus, a protein that is found in several similar species has one name but each of the species contributes a different ID. For UniProtKB, the average ambiguity is 2.61 IDs per name. If we discard the species labels, then the average ambiguity is 1.05 IDs. Ambiguous names (because the respective protein occurs in multiple species) are e.g. 'Cytochrome b' (1770 IDs), 'Ubiquinol-cytochrome-c reductase complex cytochrome b subunit' (1757), 'Cytochrome b-c1 complex subunit 3' (1757). Ambiguous names (without species

[8] We use the manually annotated and reviewed Swiss-Prot section of UniProtKB version 14, in its XML representation

[9] http://www.uniprot.org/faq/9

[10] http://www.uniprot.org/docs/nameprot

[11] The amount of proteins for a species reflects the amount of research done on the given species, rather than the amount of proteins that the species has.

[12] http://www.uniprot.org/docs/nameprot

Table 1. Frequency ranking of paths to XML elements that contain terms in UniProtKB.

Frequency	XPath (starting with /uniprot/entry/)
752,019	gene/name
397,539	protein/recommendedName/fullName
284,782	protein/alternativeName/fullName
90,397	protein/recommendedName/shortName
65,500	protein/alternativeName/shortName
16,400	protein/component/recommendedName/fullName
8913	protein/domain/recommendedName/fullName
6339	protein/component/alternativeName/fullName
5269	protein/domain/alternativeName/fullName
5023	protein/component/recommendedName/shortName
1416	protein/CdAntigenName
1207	protein/domain/recommendedName/shortName
1069	protein/component/alternativeName/shortName
787	protein/domain/alternativeName/shortName

labels) are e.g. 'Capsid protein' (103), 'ORF1' (97), 'CA' (88). Interestingly, very ambiguous names are not necessarily short, as is usually the case with ambiguous words.

Table 2 shows the orthographic/morphological properties of the names in UniProtKB in terms of how much certain types of characters influence the ambiguity. Non alphanumeric characters or change of case, while increasing ambiguity, influence the ambiguity relatively little. But as seen from the last column, digits matter a lot semantically. These findings motivate the normalization that we describe in section 3.2. Table 2 also shows the main cause for ambiguity of the names — the same name can refer to proteins in multiple species. While these proteins are identical in some sense (similar function or structure), the UniProtKB identifies them as different proteins.

Table 2. ID_ORG stands for the actual identifiers (which also include the species ID). ID stands for artificially created identifiers where we have dropped the qualification to the species. "Unchanged" = no change done to the terms; "No whitespace" = all whitespace is removed; "Alphanumeric" = everything but alphanumeric characters is removed; "Lowercase" = all characters are preserved but lowercased; "Alpha" = only letters are preserved.

	Unchanged	No whitespace	Alphanumeric	Lowercase	Alpha
ID_ORG	2.609	2.611	2.624	2.753	10.616
ID	1.049	1.050	1.053	1.058	4.145

2.3 NCBI Taxonomy

The National Center for Biotechnology Information provides a widely used resource called NCBI Taxonomy[13], which describes all known species and also lists the various forms of species names (e.g. latin names and common names). As explained in section 2.2, knowledge of these names is essential for effective disambiguation of protein names.

We compiled a term list on the basis of the taxonomy names list[14], but kept only names whose ID mapped to a UniProtKB species "mnemonic code" (such as `ARATH`)[15]. The resulting list has very little ambiguity (one example of an ambiguous term is 'mink' which can refer to both the European and the American Mink, which are classified as different species in the NCBI Taxonomy, and have therefore different identifiers).

The final list contains 31,733 entries where the species name is mapped to the UniProtKB mnemonic code. To this list, 8877 entries were added where the genus name is abbreviated to its initial (e.g. 'C. elegans') as names in such form were not included in the source data. These entries can be ambiguous in general (e.g. 'C. elegans' can refer to four different species), but are needed to account for such frequently occurring abbreviation in biomedical texts. Furthermore, six frequently occurring names that consist only of the genus name were added. In these cases, the name was mapped to a unique identifier (e.g. 'Arabidopsis' was mapped to `ARATH`), as it is expected that e.g. 'Arabidopsis' alone is always used to refer to *Arabidopsis thaliana*, and never to e.g. *Arabidopsis lyrata.*

2.4 PSI-MI Ontology

Proteomics Standards Initiative (PSI) Molecular Interactions (MI) Ontology[16] contains 2207 terms (referring to 2163 PSI-MI IDs) related to molecular interaction and methods of detecting such interactions (e.g. 'western blot', 'pull down'). There is almost no ambiguity in these names in the ontology itself. Several reasons motivate including the PSI-MI names in our term list. First, names of experimental methods are very frequent in biomedical texts. It is thus important to annotate such names as single units in order to make the syntactic analysis of the text more accurate. Second, in some cases a PSI-MI name contains a substring which happens to be a protein name (e.g. 'western blot' contains a UniProtKB term 'blot'). If the annotation program is not aware of this, then some tokens would be mistagged as protein names. Third, some PSI-MI terms overlap with UniProt terms, meaning that the corresponding proteins play an important function in protein interaction detection, but are not the subject of the actual interaction. An example of this is 'GFP' (PSI-MI ID 0367, UniProtKB ID `GFP_AEQVI`), which occurs in sentences like "interaction between Pop2p and

[13] `http://www.ncbi.nlm.nih.gov/Taxonomy/`

[14] `ftp://ftp.ncbi.nih.gov/pub/taxonomy/taxdump.tar.gz` (file `names.dmp`)

[15] `http://www.uniprot.org/help/taxonomy`

[16] `http://psidev.sourceforge.net/mi/psi-mi.obo`

GFP-Cdc18p was detected" where the reported interaction is between POP2 and CDC18, and GFP only "highlights" this interaction.

2.5 Compiled Term List

We compiled a term list of 1,679,483 terms based on the terms extracted from UniProtKB, NCBI, and PSI-MI. The term list has a simple 3-column format listing the term name, the term ID, and the term type in each entry. The type corresponds roughly to the resource the term originates from. For UniProtKB, there are two types — PROT and GEN — first assigned to all the terms from the path /uniprot/entry/protein/, and second to all the terms from /uniprot/entry/gene/. For NCBI, there are six types, distinguishing between common and scientific names, and the rank of the name in the taxonomy. For the PSI-MI Ontology terms there is just one type — MI. The frequency distribution of types is listed in table 3.

Table 3. Frequency distribution of types in the compiled term list.

Frequency	Type	Description
884,641	PROT	UniProtKB protein name
752,019	GEN	UniProtKB gene name
16,979	ocs	NCBI common name, species or below
8877	oss	NCBI scientific name, species or below
8877	ogs2	oss name, genus abbreviated (e.g. 'A. thaliana')
3316	oca	NCBI common name, above species
2561	osa	NCBI scientific name, above species
2207	MI	PSI-MI term
6	ogs1	NCBI selected genus name (e.g. 'Arabidopsis')
1,679,483		Total

In this list, 934,973 of the terms are multi-word units (e.g. 257,379 contain two tokens, 189,751 three tokens, and a few terms even 20 tokens). We did not normalize the names to any canonical representation nor generate all possible spelling variations of the names. One is expected to apply such processing during term annotation to account for differences in spacing, hyphenation etc. with respect to the terms actually occurring in the texts that undergo annotation.

3 Automatic Annotation of Terms

3.1 Introduction

Using the described term list, we can annotate biomedical texts in a straightforward way. First, the sentences and tokens are detected in the input text.

We use the LingPipe[17] tokenizer and sentence splitter which have already been trained on biomedical corpora. The tokenizer produces a granular set of tokens, e.g. words that contain a hyphen (such as 'Pop2p-Cdc18p') are split into several tokens, revealing the inner structure of such constructs which would e.g. allow to discover the interaction mention in "Pop2p-Cdc18p interaction". We slightly modified the sentence splitter to take into account abbreviations common in species names (e.g. 'sp.', 'subsp.').

The processing then proceeds by annotating the longest possible and non-overlapping sequences of tokens in each sentence, and in the case of success, assigns all the possible IDs (as found in the term list) to the annotated sequence. The annotator ignores certain common English function words (we use a list of ~50 stop words), although, it is possible that some of them are UniProtKB terms. Also, figure and table references (e.g. 'Fig. 3a' and 'Table IV') are detected and ignored.

3.2 Normalization

In order to account for possible orthographic differences between the terms in the term list and the token sequences in the text, a normalization step is included in the annotation procedure. The same normalization is applied to the term list terms in the beginning of the annotation when the term list is read into memory, and to the tokens in the input text. In case the normalized strings match exactly then the input sequence is annotated with the IDs of the term list term. We currently apply the following normalization rules which were developed gradually over a training set. Many are based on similar rules reported in the literature, see e.g. [1,2,9].

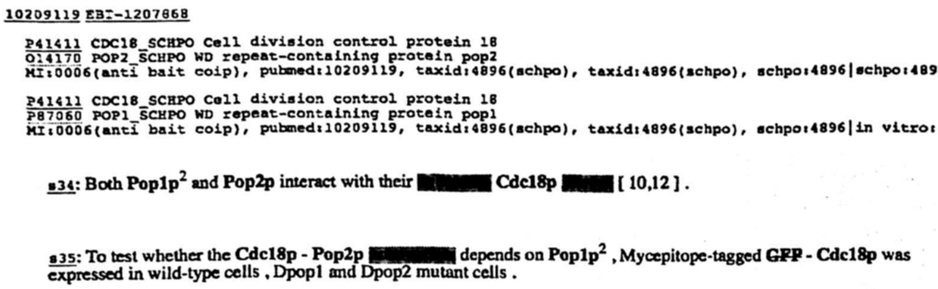
- Remove all characters that are not alphanumeric or space
- Normalize spaces, e.g. remove spaces between letters and numbers
- Normalize Greek letters, e.g. 'alpha' → 'a'
- Normalize Roman numerals, e.g. 'IV' → '4'
- Remove the final 'p' if it follows a number, e.g. 'Pan1p' → 'Pan1'
- Remove lowercase-uppercase distinction

In general, these rules increase the recall of term detection, but can lower the precision. For example, sometimes case distinction is used to denote the same protein in different species (e.g. according to UniProtKB, the gene name 'HOXB4' refers to HXB4_HUMAN, 'Hoxb4' to HXB4_MOUSE, and 'hoxb4' to HXB4_XENLA). However, the gain in recall seems to outweigh the loss of precision.

3.3 Disambiguation

A marked up term can be ambiguous for two reasons. First, the term can be assigned an ID from different term types, e.g. a UniProtKB ID and a PSI-MI

[17] http://alias-i.com/lingpipe/



```
10209119 EBI-1207868

  P41411 CDC18_SCHPO Cell division control protein 18
  O14170 POP2_SCHPO WD repeat-containing protein pop2
  MI:0006(anti bait coip), pubmed:10209119, taxid:4896(schpo), taxid:4896(schpo), schpo:4896|schpo:4896|

  P41411 CDC18_SCHPO Cell division control protein 18
  P87060 POP1_SCHPO WD repeat-containing protein pop1
  MI:0006(anti bait coip), pubmed:10209119, taxid:4896(schpo), taxid:4896(schpo), schpo:4896|in vitro:-1
```

s34: Both **Pop1p**^{2} and **Pop2p** interact with their [redacted] **Cdc18p** [redacted] [10,12] .

s35: To test whether the **Cdc18p** - **Pop2p** [redacted] depends on **Pop1p**^{2} , Mycepitope-tagged ~~**GFP**~~ - **Cdc18p** was expressed in wild-type cells , Dpop1 and Dpop2 mutant cells .

Fig. 1. Visualization of the annotation results. Terms of different type are highlighted with a different background color. The terms that were rejected by the disambiguator are crossed out. For ambiguous terms, the number of different IDs is shown in the superscript. At the top of the screenshot, the actual interaction information is show. This information originates from the IntAct protein-protein interaction knowledge base.

ID. This situation does not occur often and usually happens with terms that are probably not interesting as protein mentions (such as 'GFP' discussed in section 2.4). We disambiguate such terms by removing all the UniProtKB IDs. (Similar filtering is performed in [8].) Second, the term can be assigned several IDs from a single type. This usually happens with UniProtKB terms and is typically due to the fact that the same protein occurs in many different species. Such protein names can be disambiguated in various ways. We have experimented with two different methods: (1) remove all the IDs that do not reference a species ID specified in a given list of species IDs; (2) remove all IDs that do not "agree" with the IDs of the other protein names in the same textual span (e.g. sentence, or paragraph) with respect to the species IDs.

For the first method, the required species ID list can be constructed in various ways, either automatically, on the basis of the text, e.g. by including species mentioned in the context of the protein mention, or by reusing external annotations of the article (e.g. it might be possible to exploit MeSH annotations). We are developing and evaluating separately an approach to the detection of species names mentioned in the article. The species mentions are used to create a ranked list, which will then be used to disambiguate other entities in the text, such as the protein mentions. This recently emerged task, which is sometimes called TX task ("Taxonomy task"), is attracting growing interest as a crucial task in biomedical text mining. Currently our experimental results in this task are above 70% F-Score.

The second method is motivated by the fact that according to the IntAct database, interacting proteins are usually from the same species: less than 2% of the listed interactions have different interacting species. Assuming that proteins that are mentioned in close proximity often constitute a mention of interaction, we can implement a simple disambiguation method: for every protein mention, the disambiguator removes every UniProtKB ID that references a species that is not among the species referenced by the IDs of the neighboring protein mentions. Only in case the intersection of proposed species is empty, should all the IDs be kept — this would cover the case when the textual span contains unambiguous protein mentions which do not agree with each other with respect to their species. The neighborhood can be defined to be a textual unit such as a phrase, sentence, paragraph, etc. We currently use a sentence as the unit, as sentence splitting information is easily obtained from our linguistic pre-processing. We note that this form of disambiguation might be better applied after syntactic analysis, when we have a more granular information about potentially interacting proteins. For example, after syntactic analysis, the textual span that constitutes the neighborhood can be defined to be a relative clause or a predicate-argument structure.

It should be noted that the disambiguation result is not always a single IDs, but often just a reduced set of IDs which must be disambiguated by a possible subsequent step. Also, it can happen that none of the IDs matches a listed species. In this case all the IDs are removed. Thus, the disambiguation step can revert the decision made by the annotation step.

4 Evaluation

We evaluated the accuracy of our automatic protein name detection and grounding method on a corpus provided by the IntAct project[18]. This corpus contains a set of 6198 short textual snippets (of 1 to about 3 sentences), where each snippet is mapped to a PubMed identifier (referring to the article the snippet originates from), and an IntAct interaction identifier (referring to the interaction that the snippet describes). In other words, each snippet is a "textual evidence" that has allowed the curator to record a new interaction in the IntAct knowledge base. By resolving an interaction ID, we can generate a set of IDs of interacting proteins and a set of species involved in the interaction, for the given snippet. Using the PubMed identifiers, we can generate the same information for each mentioned article. By comparing the sets of protein IDs reported by the IntAct corpus providers, and the sets of protein IDs proposed by our tool, we can calculate the precision and recall values.

We annotated the complete IntAct corpus by marking up token sequences that the normalization step matched with an entry in the term list. Each resulting annotation includes a set of IDs which was further reduced by the two disambiguation methods described in 3.3, i.e. some or all of the IDs were removed. Figure 1 shows the visualization of the annotation output on IntAct snippets together with the actual interaction as specified in IntAct.

Results before and after disambiguation are presented in table 4. The results show a relatively high recall which decreases after the disambiguation. This

[18] `ftp://ftp.ebi.ac.uk/pub/databases/intact/current/various/data-mining/`

Table 4. Results obtained on the IntAct snippets, with various forms of disambiguation, measured against PubMed IDs. The evaluation was performed on the complete IntAct data (*all*), and on a 5 times smaller fragment of IntAct (*subset*) for which we automatically extracted the species information. Three forms of disambiguation were applied: IntAct = species lists from IntAct data; TX = species lists from our automatic species detection; span = the species of neighboring protein mentions must match. Additionally, combinations of these were tested: e.g. IntAct & span = IntAct disambiguation followed by span disambiguation. The best results in each category are in boldface.

Disamb. method	Corpus	Precision	Recall	F-Score	True pos.	False pos.	False neg.
No disamb.	all	0.03	0.73	0.05	2237	81,662	848
IntAct	all	0.56	0.73	0.63	2183	1713	804
span	all	0.03	0.71	0.06	2186	68,026	899
IntAct & span	all	0.57	0.72	**0.64**	2147	1599	840
span & IntAct	all	0.57	0.72	0.64	2142	1631	821
No disamb.	subset	0.02	0.69	0.04	424	20,344	188
IntAct	subset	0.51	0.71	0.59	414	397	170
span	subset	0.02	0.67	0.05	407	16,319	205
IntAct & span	subset	0.53	0.69	**0.60**	404	363	180
span & IntAct	subset	0.52	0.69	0.59	399	369	177
TX	subset	0.42	0.59	0.49	340	478	241
TX & span	subset	0.43	0.57	**0.49**	332	445	249
span & TX	subset	0.42	0.57	0.48	329	457	244

change is small however, compared to the gain in precision. False negatives are typically caused by missing names in UniProtKB, or sometimes because the normalization step fails to detect a spelling variation. A certain amount of false positives cannot be avoided due to the setup of task. The tool is designed to annotate all proteins contained in the sentences, but not all of them necessarily participate in interactions, and thus are not reported in the IntAct corpus.

5 Related Work

There is a large body of work in named entity recognition in biomedical texts. Mostly this work does not cover grounding the detected named entities to existing knowledge base identifiers. Recently, however, as a result of the BioCreative workshop, more approaches are extending from just detecting entity mentions to "normalizing" of the terms. In general, such normalization handles gene names (by grounding them to EntrezGene[19] identifiers). [5] gives an overview of the BioCreative II gene normalization task.

A method of protein name grounding is described in [9]. It uses a rule-based approach that integrates a machine-learning based species tagger to disambiguate protein IDs. The reported results are similar to ours.

[19] `http://www.ncbi.nlm.nih.gov/sites/entrez?db=gene`

There exist also two publicly available systems that return annotations together with UniProtKB identifiers. In the BioCreative Meta Server (BCMS)[20] [3], 2 out of 13 gene/protein taggers annotate using UniProtKB protein identifiers. The Whatizit[21] webservice annotates input texts with UniProtKB, Gene Ontology[22], and NCBI terms. A preliminary comparison showed that our approach gives results of similar quality.

6 Conclusions and Future Work

The main goal of the work described in this paper is to reliably identify protein mentions in order to identify protein-protein interactions in a subsequent processing step. We propose a method that uses large term lists extracted from various sources, and a set of normalization rules that match the token sequences in the input sentences against the term lists. Each matched term is assigned all the IDs that are possible for this term. The following disambiguation step tries to remove most of the IDs on the basis of the term context and knowledge about the species that the article discusses. The evaluation shows that a reasonably performing entity annotation system can be implemented in this way. For the evaluation, we have used the freely available IntAct corpus of snippets of textual evidence for protein-protein interactions. To our knowledge, this corpus has not been used in a similar evaluation before.

In the future, we would like to include more terminological resources in the annotation process. While the described three resources (UniProtKB, NCBI Taxonomy, PSI-MI Ontology) seem to contain the most important names used in biomedical texts, there exist also other names that are frequently used but that are not covered by these resources, e.g. cell line names (listed e.g. in CLKB [7]), names of certain chemical compounds, diseases, drugs, tissues.

We also intend to more conclusively evaluate our system against similar systems, such as BCMS and Whatizit.

Acknowledgements. This research is partially funded by the Swiss National Science Foundation (grant 100014-118396/1). Additional support is provided by Novartis Pharma AG, NITAS, Text Mining Services, CH-4002, Basel, Switzerland. The authors would like to thank the two anonymous reviewers of CICLing 2009 for their valuable feedback.

[20] http://bcms.bioinfo.cnio.es/

[21] http://www.ebi.ac.uk/webservices/whatizit/

[22] http://www.geneontology.org/

References

1. Jörg Hakenberg. What's in a gene name? Automated refinement of gene name dictionaries. In *Proceedings of BioNLP 2007: Biological, Translational, and Clinical Language Processing; Prague, Czech Republic*, 2007.
2. Jörg Hakenberg, Conrad Plake, Loic Royer, Hendrik Strobelt, Ulf Leser, and Michael Schroeder. Gene mention normalization and interaction extraction with context models and sentence motifs. *Genome Biology*, 9(Suppl 2):S14, 2008.
3. F Leitner, M Krallinger, C Rodriguez-Penagos, J Hakenberg, C Plake, C-J Kuo, C-N Hsu, RT-H Tsai, H-C Hung, WW Lau, CA Johnson, R Saetre, K Yoshida, YH Chen, S Kim, S-Y Shin, B-T Zhang, WA Baumgartner, L Hunter, B Haddow, M Matthews, X Wang, P Ruch, F Ehrler, A Ozgur, G Erkan, DR Radev, M Krauthammer, T Luong, and R Hoffmann. Introducing meta-services for biomedical information extraction. *Genome Biology*, 9(Suppl 2):S6, 2008.
4. Suresh Mathivanan, Balamurugan Periaswamy, TKB Gandhi, Kumaran Kandasamy, Shubha Suresh, Riaz Mohmood, YL Ramachandra, and Akhilesh Pandey. An evaluation of human protein-protein interaction data in the public domain. *BMC Bioinformatics*, 7(Suppl 5):S19, 2006.
5. AA Morgan, Z Lu, X Wang, AM Cohen, J Fluck, P Ruch, A Divoli, K Fundel, R Leaman, J Hakenberg, C Sun, H-h Liu, R Torres, M Krauthammer, WW Lau, H Liu, C-N Hsu, M Schuemie, KB Cohen, and L Hirschman. Overview of BioCreative II gene normalization. *Genome Biology*, 9(Suppl 2):S3, 2008.
6. Fabio Rinaldi, Thomas Kappeler, Kaarel Kaljurand, Gerold Schneider, Manfred Klenner, Simon Clematide, Michael Hess, Jean-Marc von Allmen, Pierre Parisot, Martin Romacker, and Therese Vachon. OntoGene in BioCreative II. *Genome Biology*, 9(Suppl 2):S13, 2008.
7. Sirarat Sarntivijai, Alexander S. Ade, Brian D. Athey, and David J. States. A bioinformatics analysis of the cell line nomenclature. *Bioinformatics*, 24(23):2760–2766, 2008.
8. Lorraine Tanabe and W. John Wilbur. Tagging gene and protein names in biomedical text. *Bioinformatics*, 18(8):1124–1132, 2002.
9. Xinglong Wang. Rule-Based Protein Term Identification with Help from Automatic Species Tagging. In *Computational Linguistics and Intelligent Text Processing, 8th International Conference, CICLing 2007*, pages 288–298, Mexico City, Mexico, 2007.

Automatic Word Clustering in Studying Semantic Structure of Texts

Olga Mitrofanova

St. Petersburg State University, Faculty of Philology and Arts,
Department of Mathematical Linguistics,
Universitetskaya emb., 11
199034 St. Petersburg, Russia
{alkonost-om@yandex.ru}

Abstract. The purpose of the study is to prove that results of automatic word clustering (AWC) may contribute much in investigating semantic structure of texts and in evaluating plot complexity. Experiments were carried out for Russian texts, mainly stories and short novels. Data obtained in course of study allowed to formulate and verify several linguistic hypotheses.

Keywords: Automatic Word Clustering, Russian Corpora, Semantic Structure of Texts

1 Introduction

Formalization of text structure and quantitative evaluation of semantic relations between text units prove to be of considerable importance in various fields of natural language understanding: modelling plot structure, text summarization, evaluation of translation adequacy in parallel texts, automatic text indexing, classification of texts in corpora, etc. (for a detailed analysis cf. [1], [2]).

One of the procedures providing linguistic data on semantic structure of texts is automatic word clustering (AWC). It is assumed that AWC results help to reveal semantic structure of texts and to determine plot complexity. To prove this assumption, AWC procedure was carried out with the help of a specialized AWC toolkit based on word space model. Experimental procedure implied processing Russian texts, mainly stories and short novels. A set of key words describing major topics of the plot was assigned to each text, clusters of words with similar distributions were created for each key word. Data extracted from texts through AWC procedure admit thorough linguistic interpretation. Further comparison of cluster content and structure allowed to distinguish texts characterized by a plot including a dominating topic with a number of subtopics and texts characterized by a plot including a set of major (independent or correlating) topics.

© A. Gelbukh (Ed.)
Advances in Computational Linguistics.
Research in Computing Science 41, 2009, pp. 27-34

Received 07/12/08
Accepted 11/12/08
Final version 05/02/09

2 AWC Procedure

From a linguistic point of view, AWC is based on the possibility of detecting semantic similarity of words by comparing their syntagmatic properties (co-occurrence or distribution analysis); from a technical standpoint, AWC involves construction of vector-space models for processed texts; it means that the sets of contexts for each word are represented as distribution vectors in N-dimensional space [5], [7].

It is possible to evaluate semantic similarity of words by measuring distances between their distribution representations. Numerous metrics are used for the given purpose. The selection of metrics often depends on qualitative parameters of processed texts. In our case, cosine measure (*Cos*) was chosen as a basic metric. Results of measuring semantic distances are applied in clustering: words having similar distribution representations as a rule reveal similarity of meaning and should be included into the same cluster.

General approaches to clustering are exposed in hierarchical (agglomerative, divisive), partitioning (K-means, K-medoid, etc.), hybrid algorithms. Certain linguistic tasks require application of special clustering techniques, e.g. CBC [4], MajorClust [6], etc. The choice of a particular algorithm is determined by experimental conditions (corpora size, required speed of clustering, constraints for the size of resulting clusters, etc.). In our research preference was given to agglomerative clustering algorithm as it seems to be applicable in case of limited data and appropriate for processing texts of small / medium size.

Experiments were carried out with the help of AWC tool [3]. Python-based AWC software maintains procedures of text preprocessing and agglomerative clustering. Such parameters as names of input files (processed texts and key words describing the content of a text), context window size, weight assignment for context items, size of clusters, etc. are determined by users.

Text preprocessing is performed at the first stage. Context segmentation is carried out in accordance with a particular context window size. Automatic weight assignment may be done for lexical items taking into account their positions in contexts. Then, distribution representations of words are formed, co-occurrence matrix is built, semantic distances are calculated at the second step. These data are necessary for agglomerative clustering which is performed at the third step. An output file contains clusters of words with similar distributions in a text, such clusters being formed for each key word.

3 Linguistic Data

Experiments were carried out for over 20 Russian texts, mainly stories and short novels (cf. table 1). The texts differ in authorship (A. Belyaev, M. Bulgakov, N. Gogol, A. Grin, E. Zamyatin, A. Žitinsky, etc.), in size ($N = 8\,491 \ldots 37\,217$ tokens), in lexical diversity (number of unique words $L = 3\,038 \ldots 6\,144$ tokens, Somers coefficient $S = \ln\ln L / \ln\ln N = 0.920 \ldots 0.936$). In some experiments both raw and morphologically tagged texts were subjected to analysis. Processing raw texts provides data on distribution of word forms (tokens), while processing morphologically tagged texts allows to reveal interrelations between words (lemmas) within texts. In particular cases original Russian texts and their translations were considered as well. The texts were extracted from M. Moškov digital library (http://lib.ru/). Frequency lists for each text were created, additional statistical information (frequencies of words from various parts of speech, average sentence length, amount of dialogues, etc.) was obtained with the help of FantLab linguistic processor (http://www.fantlab.ru/).

Table 1. Texts subjected to analysis.

Author, title	**Size (tokens) (*N*)**	**Number of unique words (*L*)**	**Somers coeffici-ent (*S*)**
Gogol N. *Taras Bul'ba*	37 217	6 144	0.920
Žitinsky A. *Časy s variantami* (*A Clock with Variants*)	28 092	5 197	0.922
Belyaev A. *Poslednij čelovek iz Atlantidy* (*The Last Man of Atlantis*)	26 892	5 160	0.924
Bulgakov M. *Sobačje serdce* (*Dog's Heart*)	25 218	5 321	0,928
Bulgakov M. *Rokovyje jajca* (*The Fatal Eggs*)	21 199	5 084	0.933
Zamyatin E. *Na kuličkah* (*In Kulički*)	20 832	4 544	0.928
Grin A. *Alyje parusa* (*Crimson Sails*)	20 366	4 984	0.933
Grin A. *Priklučenija Ginča* (*Ginč's Adventures*)	19 120	5 017	0.936
Belyaev A. *Večny hleb* (*Eternal Bread*)	17 103	3 640	0.924
Grin A. *Kolonija Lanfier* (*Lanfier Colony*)	15 532	3 943	0.932
Belyaev A. *Mertvaja golova* (*A Dead Head*)	14 820	3 519	0.928
Gogol N. *Povest' o tom, kak possorilis' Ivan Ivanovič s Ivanom Nikiforovičem* (*A Tale of How Ivan Ivanovič Quarrelled with Ivan Nikiforovič*)	14 052	3 071	0.923
Belyaev A. *Zolotaja gora* (*A Golgen Hill*)	12 505	3 008	0.927
Belyaev A. *Čelovek, kotoryj ne spit* (*A Sleepless Man*)	11 943	3 104	0.931
Gogol N. *Viy*	11 800	2 824	0.926
Bulgakov M. *Zapisky na manžetah* (*Notes on the Cuff*)	10 056	3 038	0.937
Belyaev A. *Ni žizn', ni sm'ert'* (*Neither Life nor Death*)	9 681	2 653	0.931
Bulgakov M. *Morphij* (*Morphia*)	8 491	2 493	0.934

4 Experimental Results

In course of experiments a set of five key words – frequent words describing major topics of the plot – was assigned to each text, e.g.:

Zamyatin E. *Na kuličkah* (*In Kulički*):
key words {*kapitan* (*captain*), *Tihmen', Marus'a, Andrej, Šmit*};

Žitinsky A. *Časy s variantami* (*A Clock with Variants*):
key words {*žizn'* (*life*), *vrem'a* (*time*), *časy* (*watch*), *ded* (*grandfather*), *ja* (*I*)}.

Clusters of lexical items with similar distributions were created for each key word. The following parameters of clustering were chosen in the experiments: similarity measure – *Cos*, context window size – ± 5, size of clusters – 10 items, no weight assignment. Previously it was found out that AWC performed with such parameters provides quite reliable data. Resulting clusters contain words or word forms associated with key words in a text and ordered according to *Cos* values. Distances between key words and their nearest neighbours in clusters (*D*) and difference between $D_{\max}$ and $D_{\min}$ in clusters (*Var*) were calculated for each text.

Table 2. Example (1): clusters of word forms extracted for key words in texts.

Text: Bulgakov M. *Morphij* (*Morphia*); key words *Polyakov*, *doktor* (*doctor*), *otdelenije* (*department*), *pis'mo* (*letter*), *Marja*; cluster elements are ordered in accordance with *Cos* values

Polyakov	*doktor (doctor)*	*otdelenije (department)*	*pis'mo (letter)*	*Marja*
pripiska (postscript) 0.328	*krasu (beauty)* 0.390	*terapevtičeskoje (therapeutic)* 0.589	*nelepoje (absurd)* 0.428	*Vlasjevna* 0.731
krupnymi (large) 0.293	*zastrelils'a (shot himself)* 0.390	*doktoru (doctor)* 0.490	*isteričeskoje (hysterical)* 0.349	*prolepetala (prattled)* 0.326
bukvami (letters) 0.293	*užas (horror)* 0.388	*hirurgičeskoje (surgery)* 0.431	153 0.349	*dviženije (movement)* 0.320
smerti (death) 0.289	*takoj (such)* 0.388	*Pavlu (Paul)* 0.423	*sarkoma (sarcoma)* 0.309	*šlepnula (slapped)* 0.320
umer (died) 0.285	*jehala (drove)* 0.379	*zaraznoje (infectious)* 0.409	*duše (soul)* 0.299	*bormotala (muttered)* 0.320
krasu (beauty) 0.254	*umer (died)* 0.333	*deckoje (infant)* 0.382	*načalo (beginning)* 0.295	*brauning (Browning)* 0.287
pomutneli (dimmed) 0.219	*drožala (trembled)* 0.323	*akušerskoje (obstetric)* 0.374	*roždalos' (was borning)* 0.284	*zadeta (touched)* 0.281
mimoletnuju (fleeting) 0.219	*doroga (road)* 0.231	*mašina (car)* 0.340	*ležalo (lay)* 0.259	*cepko (firmly)* 0.281
slyšno (audible) 0.215	*lampoj (lamp)* 0.231	*bol'šoj (big)* 0.272	*razdražat' (annoy)* 0.259	*boleznenno (painfully)* 0.281

Table 3. Example (2): clusters of word forms extracted for key words in texts.

Text: Belyaev A. *Čelovek, kotoryj ne spit* (*A Sleepless Man*); key word *preparat* (*medicine*), cluster elements are ordered in accordance with *Cos* values

***preparat* (*medicine*)**
himiki (*chemists*) 0.259
gotovyj (*ready*) 0.259
prodažu (*sale*) 0.236
uničtožavšij (*destroying*) 0.233
polučils'a (*came out*) 0.227
obnaružili (*discovered*) 0.227
polipeptidy (*polypeptides*) 0.195
vypuskalo (*produced*) 0.169
najdeny (*found*) 0.163

It seems that cluster elements often correspond to essential features of objects, persons or events denoted by key words and somehow emphasized in a text.

Relations between cluster elements can be characterized as syntagmatic and / or paradigmatic, e.g. synonymy & attributive relation: *terapevtičeskoje* (*therapeutic*), *hirurgičeskoje* (*surgery*), *zaraznoje* (*infectious*), *deckoje* (*infant*), *akušerskoje* (*obstetric*) – *otdelenije* (*department*); meronymy: *otdelenije* (*department*) – *doktor* (*doctor*); person – actions: *Marja* – *prolepetala* (*prattled*), *šlepnula* (*slapped*), *bormotala* (*muttered*), phraseological units and compounds: *Marja* – *Vlasjevna* (first name & second name), etc. (cf. table 2).

Those relations can be properly described in terms of semantic roles and lexical functions, e.g. action *obnaružili* (*discovered*) – agent *himiki* (*chemists*), result *preparat* (*medicine*) – attribute *gotovyj* (*ready*), *uničtožavšij* (*destroying*); action *prodažu* (*sale*) – theme *preparat* (*medicine*), etc. (cf. table 3).

Thus, AWC allows to reveal and analyze not only standard but also occasional relations between lexical items which may be specific for a particular text or a set of texts of the same author or dealing with the same topic.

In some tests clustering was performed in two modes: with weight assignment and without weight assignment. In most cases clusters contain similar elements – word forms (tokens) in raw texts or words (lemmas) in tagged texts. At the same time those words or word forms within clusters may be ordered differently as regards their *Cos* values. So, clusters may be similar in content, but they may differ in structure (cf. table 4). It should be noted that in experiments with weight assignment *Cos* values for nearest neighbours of key words in clusters (*D*) seem to be lower than in experiments without weight assignment.

Table 4. Example: clusters obtained in experiments with / without weight assignment.

Text: Gogol N. *Viy*; key word *bursak (seminarist)*, cluster elements are ordered in accordance with *Cos* values	
Clustering without weight assignment	**Clustering with weight assignment**
bursak (seminarist)	***bursak (seminarist)***
sodrognuls'a (*shuddered*) 0.479	*sodrognuls'a* (*shuddered*) 0.436
pozelenevšije (*green*) 0.442	*pozelenevšije* (*green*)0.405
otstupivši (*having stepped aside*) 0.420	*holod* (*cold*) 0.371
vperil (*stared*) 0.379	*izumlenija* (*amuzement*) 0.364
holod (*cold*) 0.359	*mertvyje* (*dead*) 0.338
čuvstvitel'no (*perceptibly*) 0.299	*čuvstvitel'no* (*perceptibly*) 0.305
izumlenija (*amuzement*) 0.295	*sv'atoj* (*saint*) 0.222
žizni (*life*) 0.259	*probežal* (*run*) 0.205
sv'atoj (*saint*) 0.200	*žizni* (*life*) 0.176

We also considered clustering results obtained in course of processing raw texts and morphologically tagged texts. Correspondence of word forms (tokens) and words (lemmas) in clusters created for raw and morphologically tagged texts (cf. table 5) proves the existence of stable intrinsic relations underlying text structure. These relations remain almost intact as the analysis moves from the level of word forms (tokens) to the level of words (lemmas). So, AWC procedure may furnish us with additional information on the integrity and continuity of the text as a complex of heterogeneous linguistic units.

AWC proves to be of much use in comparative analysis of original texts and translations, as it often allows to evaluate stylistic and semantic similarity of texts. Similarity of clusters formed for a word and its translation equivalent reveals correspondence between contexts of those words in the original and in translation, while differences of content and structure of such clusters imply syntactic / morphological / lexical differences of texts in question as well as inconsistency in the choice of translation equivalents for a particular word or for lexical items co-occurring with this word in contexts (cf. table 6).

As statistical parameters of texts may influence results of clustering, additional tests were required. We've studied the texts written by A. Belyaev which reveal common semantic structure and are characterized by a branching plot with numerous and frequently changing topics. The given texts differ in size and in number of unique words. At the same time, distances between key words and their nearest neighbours in clusters don't vary much for those texts ($D \in [0.088 \ldots 0.259]$). It turns out that such parameters as size and number of unique words play important but not decisive role in studying text structure by means of AWC.

Table 5. Example: clusters obtained in experiments with raw and tagged texts.

Text: Bestužev-Marlinsky A. *Strašnoje gadanje* (*A Scary Fortune-telling*); key word *neznakomec* (*stranger*), cluster elements are ordered in accordance with *Cos* values	
Clustering in a raw text (tokens)	**Clustering in a tagged text (lemmas)**
neznakomec (stranger)	***neznakomec (stranger)***
stenky (*wall*) 0.219	*drognut'* (*quaver*) 0.223
podjezdu (*entrance*) 0.219	***trost'* (*cane*) 0.198**
***vysadiv* (*having put off*) 0.219**	***vysadit'* (*pull off*) 0.197**
***večor* (*evening*) 0.218**	*zajti* (*overstep*) 0.196
***zahvatyvaja* (*seizing*) 0.216**	***večor* (*evening*) 0.196**
***rasseržen* (*angry*) 0.216**	*kalitka* (*gate*) 0.195
***trost'* (*cane*) 0.193**	***zahvatyvat'* (*seize*) 0.194**
***gorst'ami* (*in handfuls*) 0.188**	***rasserdit'*(*anger*) 0.192**
car'a (*tzar*) 0.172	***ironičeskij* (*ironical*) 0.173**
***ironičeskoju* (*ironical*) 0.165**	***gorst'*(*in handfuls*) 0.169**

Table 6. Example: comparison of clusters formed for test words in the original text and in translation.

Texts: Grin A. *Alyje parusa* (*Crimson Sails*); test words *Sekret* (*Secret*), *galiot* (*galliot*), cluster elements are ordered in accordance with *Cos* values			
Russian text	**English text**	**Russian text**	**English text**
Sekret (Secret)	***Secret***	***Galiot (Galliot)***	***galliot***
potr'asenija (*shock*) 0.239	*intimations* 0.213	*trehmačtovyj* (*three-mastered*) 0.800	*masted* 0.843
vdohnovennogo (*inspired*) 0.210	*hurries* 0.212	*dvesti* (*two hundred*) 0.700	*purchased* 0.556
dvesti (*two hundred*) 0.178	*agitation* 0.202	*šest'des'at* (*sixty*) 0.600	*sixty* 0.527
neuderžimymi (*uncontrollable*) 0.178	*rounding* 0.201	*kuplennyj* (*purchased*) 0.600	*ton* 0.509
slezami (*tears*) 0.178	*shock* 0.173	*tonn* (*ton*) 0.500	*brig* 0.316
nravits'a (*likes*) 0.149	*cape* 0.173	*Grejem* (*Gray*) 0.329	*hundred* 0.271
kamenistoj (*rocky*) 0.149	*uncontrollable* 0.144	*sobstvennikom* (*proprietor*) 0.3	*orion* 0.222
padajuš'im (*falling*) 0.147	*masted* 0.117	*kapitanom* (*captain*)0.291	*rugged* 0.211
golovokružitel'no (*astoundingly*) 0.117	*galliot* 0.093	*mačty* (*masts*) 0.290	*Arthur* 0.189

Thorough treatment of AWC results allowed us to distinguish three main types of texts with regard to their semantic structure (Types 1, 2, and 3).

Type 1 is represented by texts characterized by a plot including a dominating topic with a number of subtopics. For such texts distances between key words and their nearest neighbours in clusters (D) and difference between $D_{\max}$ and $D_{\min}$ in clusters (Var) are as follows: $D \geq 0.300$, $Var \geq 0.200$.

Type 2 is represented by texts characterized by a plot including a set of major (probably independent) topics. For such texts distances between key words and their nearest neighbours in clusters (D) and difference between $D_{\max}$ and $D_{\min}$ in clusters (Var) are as follows: $D < 0.300$, $Var < 0.200$.

Type 3 is represented by texts characterized by a plot including a set of major (probably correlating) topics. For such texts distances between key words and their nearest neighbours in clusters (D) and difference between $D_{\max}$ and $D_{\min}$ in clusters (Var) are as follows: $D \geq 0.300$, $Var < 0.200$.

Examples of texts representing Types 1, 2, and 3 are given in table 7.

Table 7. Texts representing Types 1, 2 and 3.

Type, author, title	***D***	***Var***
Type 1		
Gogol N. *Taras Bul'ba*	0.379	0.252
Grin A. *Priklučenija Ginča* (*Ginč's Adventures*)	0.406	0.231
Gogol N. *Povest' o tom, kak possorilis' Ivan Ivanovič s Ivanov Nikiforovičem* (*A Tale of How Ivan Ivanovič Quarrelled with Ivan Nikiforovič*)	0.453	0.357
Gogol N. *Viy*	0.547	0.424
Belyaev A. *Zolotaja gora* (*A Golgen Hill*)	0.566	0.471
Bulgakov M. *Morphij* (*Morphia*)	0.731	0.403
Type 2		
Žitinsky A. *Časy s variantami* (*A Clock with Variants*)	0.149	0.048
Zamyatin E. *Na kuličkah* (*In Kulički*)	0.174	0.068
Grin A. *Alyje parusa* (*Crimson Sails*)	0.204	0.091
Bulgakov M. *Sobačje serdce* (*Dog's Heart*)	0.212	0.103
Belyaev A. *Ni žizn', ni sm'ert'* (*Neither Life nor Death*)	0.222	0.070
Belyaev A. *Poslednij čelovek iz Atlantidy* (*The Last Man of Atlantis*)	0.224	0.115
Grin A. *Kolonija Lanfier* (*Lanfier Colony*)	0.224	0.139
Belyaev A. *Mertvaja golova* (*A Dead Head*)	0.243	0.155
Belyaev A. *Čelovek, kotoryj ne spit* (*A Sleepless Man*)	0.259	0.144
Belyaev A. *Eternal bread* (*Večny Hleb*)	0.268	0.130
Bulgakov M. *Rokovyje jajca* (*The Fatal Eggs*)	0.279	0.182
Type 3		
Bulgakov M. *Zapisky na manžetah* (*Notes on the Cuff*)	0.359	0.091

Our observations on semantic structure of texts require more detailed consideration and further verification.

5 Conclusion

In course of our experiments performed for Russian stories and short novels we proved that AWC may be of great help in distinguishing three types of texts as regards their semantic structure. We managed to describe texts of different plot complexity: texts revealing a dominating topic and a set of subtopics, texts revealing a set of major (probably independent) topics, and texts revealing a set of major (probably correlating) topics. Linguistic analysis of cluster content and structure allowed to study standard as well as occasional semantic relations between lexical items occurring in texts. Experiments on AWC performed for raw and morphologically tagged texts proved the existence of intrinsic relations underlying text structure, those relations being preserved at two levels of analysis: the level of word forms (tokens) and the level of words (lemmas). Comparison of AWC results obtained for the original texts and their translations proved to be relevant in the evaluation of stylistic and semantic similarity of texts.

Further research implies experiments carried out for texts of different size, genre and authorship, with expanded sets of key words, with changing parameters (context window size, cluster size, etc.).

Acknowledgements. The author expresses deep gratitude to CICLing–2009 Program and Organizing Committees for having accepted the paper, to reviewers for their kind attention and wise comments which helped to improve the paper, to Mikhail Alexandrov (Barcelona, Spain) and Elena Iagounova (St. Petersburg, Russia) for their deep interest in the given research and for their help and encouragement.

References

1. Bolshakov, I.A., Gelbukh, A.: Computational Linguistics: Models, Resources, Applications. IPN – UNAM – Fondo de Cultura Económica (2004)
2. Leontjeva, N.N.: Avtomatičeskoje Ponimanije Tekstov: Sistemy, Modeli, Resursy. Moscow (2006)
3. Mitrofanova, O., Mukhin, A., Panicheva, P., Savitsky, V.: Automatic Word Clustering in Russian Texts. In: Matoušek, V., Mautner, P. et al. (eds.): Text, Speech and Dialogue. Proceedings of the Tenth International Conference TSD–2007, Pilsen, Czech Republic, September 3–7, 2007. Lecture Notes in Artificial Intelligence, Vol. 4629. Springer-Verlag, Berlin Heidelberg New York (2007) 85–91
4. Pantel, P.: Clustering by Committee. Ph.D. Dissertation, Department of Computing Science, University of Alberta (2003) http://www.isi.edu/~pantel/Content/publications.htm
5. Sahlgren, M.: The Word-Space Model: Using Distributional Analysis to Represent Syntagmatic and Paradigmatic Relations between Words in High-dimensional Vector Spaces. Ph.D. dissertation, Department of Linguistics, Stockholm University (2006) http://www.sics.se/~mange/TheWordSpaceModel.pdf
6. Stein, B., Meyer zu Eissen, S.: Document Categorization with MajorClust. In: Proceedings of the 12th Workshop on Information Technology and Systems WITS–02. Barcelona, Spain (2002) 91–96
7. Widdows, D.: Geometry and Meaning. Center for the Study of Language and Information – Lecture Notes, Vol. 172. The University of Chicago Press (2004)

Semantically-Driven Extraction of Relations between Named Entities

Caroline Brun and Caroline Hagège

Xerrox Research Centre Europe, 6 chemin de Maupertuis 38240 Meylan, France
{Caroline.Brun, Caroline.Hagege}@xrce.xerox.com

Abstract. In this paper, we describe a method that automatically generates lexico-syntactic patterns which are then used to extract semantic relations between named entities. The method uses a small set of seeds, i.e. named entities that are a priori known to be in relation. This information can easily be extracted from encyclopedias or existing databases. From very large corpora we extract sentences that contain combinations of these attested entities. These sentences are then used in order to automatically generate, using a syntactic parser, lexico-syntactic patterns that links these entities. These patterns are then re-applied on texts in order to extract relations between new entities of the same type. Furthermore, the patterns that are extracted not only provide a way to spot new entities relations but also build a valuable paraphrase resource. An evaluation on the relation holding between an event, the place of the event occurrence and the date of the event occurrence has been carried out on French corpus and shows good results. We believe that this kind of methodology can be applied for other kinds of relation between named entities.

1 Introduction

In this paper we describe a system that extracts accurately semantic relations between named entities from raw text. Taking as input a small set of already known relations that can be extracted from encyclopedias or from databases, our system first learn from a large corpus a wide range of lexico-syntactic patterns conveying the desired semantic relation. These learned patterns are then further applied on texts, and as a result, new occurrences of the given semantic relations linking new entities are detected. As all the patterns extracted represent a comparable semantic situation, they can be considered as paraphrase patterns. These patterns can then be used both in generation and for information extraction tasks.

2 Related Work

Many research works on extraction of relation between entities have already been performed since this kind of information is useful for a wide range of applications of information extraction. For instance [8] describe an algorithm to extract relations between named entities and the resulting improvement of a question answering

© A. Gelbukh (Ed.)
Advances in Computational Linguistics.
Research in Computing Science 41, 2009, pp. 35-46

Received 07/11/08
Accepted 09/12/08
Final version 04/02/09

system. Semantic relation detection between named entities has also been investigated in the context of the semantic web (try to obtain a rich and accurate metadata annotation from web content) as for instance in [6]. In the biomedical domain, [7] and [14] present methods to automatically extract interaction relations between genes and/or protein using machine learning techniques.

Some of these approaches rely on pattern matching exploiting simple syntactic relations as the Subject-Verb-Object relation. Sometimes, additional ontological knowledge is also exploited. These approaches take advantage of the fact that a certain syntactic configuration can be mapped onto a semantic relation. This is particularly well described in [12] where a shallow parser and a deeper parser are used to extract SVO relations between triples. In [7] and [14], a previous dependency analysis is performed to derive necessary information for learning algorithms.

Other approaches based on syntactic analysis rely on the fact that the type of syntactic relations is predefined: it has to be stated beforehand that a SVO syntactic relation or an appositive relation is meaningful for the kind of semantic relation the system extracts.

Other approaches, which do not presupposes the type of syntactic relations (see [7] and [14]) holding between entities, take into account both positive and negative examples of possible relations (relying on the closed-world assumption, stating that if no link is found between two entities, then these entities are never related).

Our approach neither needs the a priori knowledge of relevant syntactic links conveying the semantic relational information of interest, nor assumes the presence of negative examples: in some cases, like the one we present for illustration, the notion of negative example is not pertinent since one of the entities in focus convey temporal information, which is not compatible with close world hypothesis. This approach has the advantage to extract relations between entities that are less predictable in advance.

In the following steps we describe our method by exemplifying it on a concrete case: the extraction of relations between events, the place where the event occurs and the date of the event occurrence. The same methodology can be applied for other kind of relations and in other contexts. It must be stressed that our approach needs quite limited resources, namely a small list of attested and trustable relations between entities, that we can find in freely available encyclopedia as Wikipedia and a linguistic engine, which is able to detect named entities and to syntactically relate linguistic units appearing in texts.

3 A System for NE Relation Extraction

In this section, we describe the general methodology we used to build our system; we also describe the robust parser we use as core component of the system, and then present the resulting prototype.

3.1 Description of the Method

The first step of our method is to extract attested related entities from a trustable external resource. In other words, we need a resource which stores n-uples of entities which are linked by some kind of semantic relation. For instance, from the French Wikipedia in the article about Olympic Games, we can obtain the following list of triples which links a date (date of the event), a place (place of the event at that date) and the event name (in this case, the Olympic Games).

1896	Grèce Athènes Jeux Olympiques.
1900	France Paris Jeux Olympiques.
1904	États-Unis Saint-Louis Jeux Olympiques.
1908	Royaume-Uni Londres Jeux Olympiques.
1912	Suède Stockholm Jeux Olympiques.
1916	Berlin Allemagne Jeux Olympiques.
1920	Belgique Anvers Jeux Olympiques.
1924	France Paris Jeux Olympiques.
1928	Pays-Bas Amsterdam Jeux Olympiques.
1932	États-Unis Los Angeles Jeux Olympiques.
1936	Allemagne Berlin Jeux Olympiques.
1940	Helsinki Finlande Jeux Olympiques.
1944	Londres Grande-Bretagne Jeux Olympiques.
1948	Royaume-Uni Londres Grande-Bretagne Jeux Olympiques.
1952	Finlande Helsinki Jeux Olympiques.
1960	Italie Rome Jeux Olympiques.
1964	Japon Tôkyô Jeux Olympiques.
1968	Mexique Mexico Jeux Olympiques.
1972	Allemagne Munich Jeux Olympiques.
1976	Canada Montréal Jeux Olympiques.
1980	Moscou URSS Jeux Olympiques.
1984	États-Unis Los Angeles Jeux Olympiques.
1988	Séoul Corée du Sud Jeux Olympiques.
1992	Espagne Barcelone Jeux Olympiques.
1996	États-Unis Atlanta Jeux Olympiques.
2000	Australie Sydney Jeux Olympiques.
2004	Grèce Athènes Jeux Olympiques.
2008	Chine Pékin Jeux Olympiques.
2012	Royaume-Uni Londres Jeux Olympiques.

The second step of the method consists in extracting from very large corpora all sentences that contain all or a part of one of the triples appearing in the list. In other words, for our example, we extract all the sentences containing the name of the event together with the date and/or the place(s). For example, for "1992 Espagne Barcelone Jeux Olympiques.", we extract all sentences containing Espagne/1992/Jeux Olympiques, Barcelone/1992/Jeux Olympiques, Espagne/Jeux Olympiques, Barcelone/Jeux Olympiques, and 1992/Jeux Olympiques.

Sentences like the following ones are extracted from our initial list:

1. Le CIO a fixé des objectifs de lutte contre le dopage durant les jeux Olympiques de Sydney. (Event + Place)
(IOC has given objectives for fighting doping during the Olympic Games of Sydney)

2. Toutefois, les premières mesures contre le dopage n'ont été prises qu'après les jeux olympiques d'Helsinki de 1952. (Event + Place + Date)
(The first measures against doping, however, have been only taken after the Olympic Games of Helsinki in 1952)
3. En 2008, la Chine accueillera les Jeux olympiques, et le pays est un membre permanent du Conseil de sécurité des Nations unies. (Event + Place + Date)
(In 2008, China will receive the Olympic Games, and the country is a permanent member of the United Nations Security Council.
4. Ce n'est qu'en 1928 que décision a été prise d'autoriser la participation des femmes aux Jeux olympiques. (Event + Date).
Etc.
(It is only in 1928 that the decision has been taken of authorizing the participation of women to the Olympic Games)

This extraction is a simple processing step, because only pattern matching is required (possibly with some normalization of case).

The third step consists in applying a robust syntactic dependency parser on these sentences in order to extract the syntactic relationships linking the attested elements. It is important to stress that there is no a priori about the possible syntactic relations that can hold between the entities. These links can also be direct links or indirect links, transitivity being taken into account.

In addition to the syntactic parsing, the linguistic engine also performs named entity recognition, in order to be able to generalize the patterns extracted.

The following example shows the relations that the linguistic engine extracts from sentence 3. Binary relations correspond to grammatical links between lemmatized lexical units of the sentence while unary relations correspond to the identification of Named Entities.

En 2008, la Chine accueillera les Jeux olympiques, et le pays est un membre permanent du Conseil de sécurité des Nations unies.
(In 2008, China will receive the Olympic Games, and the country is a permanent member of the United Nations Security Council.

SUBJ(accueillir,Chine)
OBJ(accueillir,Jeux olympiques)
VMOD(accueillir,2008)
EVENT(Jeux Olympiques)
DATE(2008)
PLACE(Chine)

The fourth step consists in generalizing the set of relations captured by the parser in order to obtain a generic lexico-syntactic rules pattern. This generalization is performed by abstracting the Named Entities by their type (in our example, event, date and place) and keeping the lemmas of the lexical elements which are present in the grammatical binary relations.

From our previous example, we obtain automatically from the set of extracted relationships the following lexico-syntactic pattern (where & represent a conjunction).

SUBJ(accueillir, PLACE(X)) & OBJ(accueillir,EVENT(Y)) & VMOD(accueillir,DATE(Z))
==>DATE-and-PLACE-of-EVENT(Z,X,Y)

Once learned, as our syntactic parser is rule-based, these learned rules can be integrated as such on top of the syntactic parser set of rules, and applied as any other kind of rules on any corpora.

As a result, thanks to the generalization of entity types, the application of these rules, will enable to extract new n-uples of related named entities that were not present in our initially extracted list (i.e. other kind of events that are not necessarily Olympic games or even sport event associated with their date of occurrence and their place of occurrence may be discovered).

The whole process is summarized on the following figure:

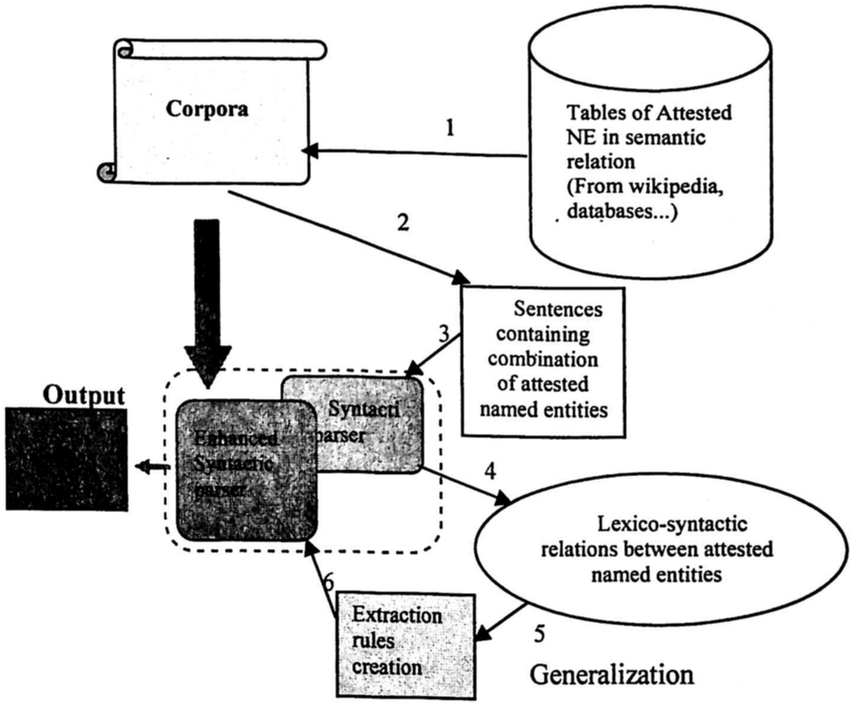

Corpora
1
Tables of Attested NE in semantic relation (From wikipedia, databases...)
2
Sentences containing combination of attested named entities
3
Output
Enhanced Syntactic parser
Syntacti parser
4
Lexico-syntactic relations between attested named entities
5
Generalization
6
Extraction rules creation

Fig. 1. Summary of the process

3.2 Robust and Deep Parsing using XIP

As a fundamental component of the system we designed for named entity relation extraction, we use the Xerox Incremental Parser (XIP, see [3]) in order to perform robust and deep syntactic analysis. Deep syntactic analysis consists here in the construction of a set of syntactic relations[1] from an input text. These relations link lexical units of the input text and/or more complex syntactic domains that are constructed during the processing (mainly chunks, see [1]). These relations are labeled with deep syntactic functions. More precisely, a predicate (verbal or nominal) is linked with what we call its deep subject (SUBJ-N), its deep object (OBJ-N), and modifiers.

In addition, the parser calculates more sophisticated and complex relations using derivational morphologic properties, deep syntactic properties (subject and object of infinitives in the context of control verbs), and some limited lexical semantic coding (Levin's verb class alternations, see [9], and some elements of the Framenet[2] classification [11]). These deep syntactic relations correspond roughly to the agent-experiencer roles that is subsumed by the SUBJ-N relation and to the patient-theme role subsumed by the OBJ-N relation (see [5] and [4]). Not only verbs bear these relations but also deverbal nouns with their corresponding arguments.

The use of such sophisticated relations in the pattern extraction process enables us to extract "normalized patterns" that have a wide coverage. For example, a single pattern extracted with the normalization grammar will match different surface realization such as a passive form, active form or nominalization of a given predicate.

This parser includes also a module for Named Entity recognition, i.e. detection of numerical expressions, dates, person, organization, location names, and events. This module is built within the XIP parser presented above, on top of a part-of-speech tagger. This system is purely rule-based, and consists in a set of ordered local rules that use lexical information combined with contextual information about part-of-speech, lemma forms and lexical features.

These rules detect the sequence of words involved in the entity and assign a feature (loc, org, date, event, etc.) to the top node of the sequence, which is a noun in most of the cases. This system has been evaluated internally and show a performance of .90 in F-measure on the whole set of types of entities.

Here is an example of an output (Named entities, chunk tree and deep syntactic relations) of the most sophisticated version of the grammar:

"Lebanon still wanted to see the implementation of a UN resolution."

TOP{SC{NP{Lebanon} FV{still wanted}} IV{to see} NP{the implementation} PP{of NP{a UN resolution}} .}

PLACE_COUNTRY(Lebanon)
ORGANISATION(UN)
MOD_PRE(wanted,still)
MOD_PRE(resolution,UN)
MOD_POST(implementation,resolution)
EXPERIENCER_PRE(wanted,Lebanon)
EXPERIENCER(see,Lebanon)
CONTENT(see,implementation)
EMBED_INFINIT(see,wanted)
OBJ-N(implement,resolution)

[1] Inspired from dependency grammars, see [10] and [13].

[2] http://framenet.icsi.berkeley.edu/

3.3 Description of the System

In order to validate our method, we implemented a prototype which uses as relation seeds the relation holding between the Olympic Games and the corresponding date and place of occurrence. As shown in the first subsection, we first extracted from Wikipedia a first list of triples.

To set up the prototype, we used a French corpus of about 1.3 million sentences, provided by the European community about "acquis communautaires" (i.e. "community acquis", the rights and obligations that EU countries share). We divide this corpus in two parts, and, then, on the first half, we extract sentences that contain the triplets "Olympic game/date/place" given by the Wikipedia attested list, and if the triplet is not present, then the couples "Olympic game/date" and "Olympic game/place". From the initial corpus, we extracted 150 sentences containing either a triplet or couple of attested entities.

We parse these sentences with XIP, which provided us with a list of dependencies involving the attested entities.

This list of dependencies is automatically transformed[3] into a set of XIP rules that can be applied on top of the parser previously used.

For example, when parsing:

« Londres organisera les Jeux Olympiques en 2012 »
(London will organize the Olympic Games in 2012)
XIP outputs the following dependencies:

```
SUBJ(organiser,Londres)
OBJ(organiser,jeux olympiques)
VMOD_POSIT1(organiser,2008)
PREPOBJ(2008,en)
DETERM_DEF_SPORT(jeux olympiques,le)
DATE(2008)
PLACE_CITY(Londres)
EVENT_SPORT(jeux olympiques)
```

We then select from this output all dependencies that:

- Involve one of the named entity in focus (here entities of type event, date & place)
- Involve only non-functional words (noun, verbs, adjectives and not preposition or determiner for example)

In this example, the selected dependencies are then:

```
SUBJ(organiser,Londres)
OBJ(organiser,jeux olympiques)
VMOD_POSIT1(organiser,2008)
PLACE_CITY(Londres)
DATE(2008)
EVENT_SPORT(jeux olympiques)
```

[3] By a python script developed for that purpose.

We then abstract on the named entity types, and consequently deduce the following XIP rule:

```
If(SUBJ(#1[lemma:="organiser"],#2) & PLACE(#2) & OBJ(#1,#3) & EVENT(#3)
& VMOD(#1,#4) & DATE(#4))
==>  DATE-and-PLACE-of-EVENT(#4,#2,#3)
```

This rule can then be applied as such incrementally on top of the parser, to discover new entities in semantic relation.

When applying the enhanced parser integrating the new learned rules, we get for example the following result, on a different event:

« Le Championnat d'Europe sera organisé en 2004 par le Portugal. »
(European Championship will be organized in 2004 by Portugal).

```
SUBJ_PASSIVE(organiser,championnat d'Europe)
SUBJ(organiser,Portugal)
OBJ(organiser,championnat de Europe)
VMOD_POSIT1(organiser,2004)
NMOD_POSIT1(2004,Portugal)
AUXIL_PASSIVE(organiser,être)
DATE(2004)
PLACE_COUNTRY(Portugal)
EVENT_SPORT(championnat d'Europe)
DATE-and-PLACE-of-EVENT(2004, Portugal, championnat d'Europe)
```

Additionally, this example shows the interest of using deep syntax ("syntactic normalization", cf. [6]), which enables to map active and passive cases.

On the 150 sentences about the Olympic Games that we extracted in the original corpus, we automatically build about 60 XIP rules that are incrementally applied on top of the parser, in a new layer of rules.

While many of the research focuses on extracting subject-verb-object patterns, our method does not make a priori hypothesis for the type of syntactic dependencies that links the entities in semantic relation (typically, dates have verbal or nominal modifier functions). It can therefore account for examples like:

« Les jeux olympiques de 1992 se déroulaient à Albertville. »
(The Olympic Games of 1992 were happening in Albertville).

```
SUBJ(dérouler,jeux olympiques)
VMOD_POSIT1(dérouler,Albertville)
NMOD_POSIT1(jeux olympiques,1992)
PREPOBJ(Albertville,à)
PREPOBJ(1992,de)
DETERM_DEF(jeux olympiques,le)
REFLEX(dérouler,se)
DATE(1992)
PLACE_CITY(Albertville)
EVENT_SPORT(jeux olympiques)
DATE-and-PLACE-of-EVENT(1992, Albertville, jeux olympiques)
```

Here, the entities of date and location have modifier syntactic functions and do not follow the pattern subject-verb-object of the above-mentioned two previous examples.

Applying the system on different kind of corpora, such as newspapers, shows that the system extract relations concerning other types of event, such as cultural events, since there are recognized by the named entity module and since the relation patterns hold also for them:

« La Biennale d'art contemporain aura lieu à Lyon du 16 septembre 2009 au 3 janvier 2010 »

(The Biennale of contemporary art will take place from the 16 September 2009 until the 3 January 2010)

SUBJ(aura,Biennale d'art contemporain)
OBJ2(aura,lieu)
VMOD_POSIT1(aura, du 16 septembre 2009 au 3 janvier 2010)
VMOD_POSIT1(aura, Lyon)
DATE_INTERVAL(du 16 septembre 2009 au 3 janvier 2010)
PLACE_CITY(LYON)
EVENT_CULTURAL(Biennale d'art contemporain)
DATE-and-PLACE-of-EVENT(du 16 septembre 2009 au 3 janvier 2010, Lyon, Biennale d'art contemporain)

3.4 Extraction of Paraphrase Patterns

One of the interesting points of the system we developed is the construction of a valuable resource of paraphrase templates that have been automatically constructed for extracting new relations between EVENTS, DATES and PLACES. Our templates correspond of XIP grammar rules as shown in section 3.3. and are very precise descriptions of the grammar links holding between the different NE.

As all our patterns semantically denote a situation where an event occurs in a certain place and possibly at a certain date, we can consider two things:

First, the text segments that enable to extract the templates are paraphrases of one another;

Second, the templates themselves can be considered are basis for the generation of paraphrases.

Let's take some example to illustrate:

if ((EVENT(#1) & SUBJ(#2[lemme:"avoir"],#1) & OBJ2(#2,#3[lemme:"lieu"]) & VMOD(#2,#4) & PLACE(#4) & PREPD(#4,?[lemma: "à"]))
==> PLACE-of-EVENT(#1,#4)

This template expresses the situation of an event taking place in a certain place (NEW-PLACE-EVENT situation).

It expresses that the EVENT is the subject of support verb "avoir lieu" and that this verb has a modifier of type PLACE which is introduced by preposition "à"

Following the canonical sentence order for French, this corresponds for instance to expressions like:

(1) <EVENT> a eu lieu à <PLACE>

(<EVENT> took place in <PLACE>)

Let's now take another example of pattern denoting the same situation:
if(SUBJ(#1[lemme:"accueillir"],#2) & OBJ(#1,#3) & EVENT(#3) & COREF[rel](#4,#2) & PLACE(#4))
==> PLACE-of-EVENT(#3,#4)

This time the template expresses that the verb "accueillir" has to have a subject of type PLACE and at the same time a direct object as type EVENT.

Following the canonical order for French, this corresponds for instance to expressions like:
(2) <PLACE> a accueilli <EVENT>
(<PLACE> received <EVENT>)

(1) and (2) are paraphrase patterns and linguistic realizations of those patterns convey the same information.

These two examples can be considered as exact paraphrases candidate, however we also extract approximate paraphrase patterns like:

<PLACE> organize <EVENT>
or
<PLACE> prepare <EVENT>,

that do not denote exactly the same situation (the occurrence of the event) but a situation which is presupposed by the second one. A manual verification or an automatic access to WordNet data could enable however to take into account this difference (organize and prepare are in a same WordNet synset). For the moment we did not experiment the automatic verification, but this is a possible future development.

Following the experiment described above, we extracted 35 paraphrase patterns expressing the situation of an event occurrence in a certain place and at a certain date. They include both exact and approximate paraphrases as explained before. We believe that this methodology can be applied to other and can be a valuable resource for different kind of information extraction application (like QA) or even for textual entailment.

4 Evaluation

In order to test our system, we applied it on the second half of the initial corpora. On this part of the corpora, we extract a subset that is potentially in focus for our prototype, i.e. sentences that contain at least a named entity event extracted by the robust parser XIP. This subset consists in about 1500 sentences. We annotate them manually in terms of relations (DATE-AND-PLACE-of-EVENT, DATE-of-EVENT, PLACE-of-EVENT). The confrontation of this corpus with our system gives the following results in term of precision and recall (see Table 1).

It shows that our system gets a very high precision, with an acceptable recall. The recall is a bit low for the triplet relation, because in many cases, our system didn't

Table 1. Evaluation results

	Date-and-place-of-event	Date-of-event	Place-of-Event	All
Precision	90.3	90.9	92.9	92.3
Recall	49.1	80.0	83.7	77
F-measure	63.6	85.1	88.1	83.9

catch the full semantic relation between the 3 elements, but on partial relation (date or place of event). This evaluation shows however that our method is promising. We can indeed relate in a certain way our results with the results obtained[4] by [8] where the authors show that they obtain an f-measure of 72.8 in the detection of most common binary relations between named entities using machine learning methods.

Related work for English (relation detection task between named entities obtained in last ACE competition) shows an overall F-score of 21.6. However it has to be stated that the kind of text provided by ACE included broadcast transcripts and web logs which make the task much more complicated. Furthermore the relations to be recognized are of different kind, see [2].

5 Conclusion

In this paper we present a method illustrated by an effective experiment to extract relations between named entities. Although this field has been studied a lot and that there exists many research work on that matter, our proposal has the particularity to be able to detect semantic relations that are not necessarily conveyed by prototypical syntactic relations (like Subject-Verb-Object relations). This is particularly suitable when one of the related elements is a date or a place as they are very often in very variable syntactic configurations (noun modifier, verb modifier). Our method relies on a first reliable small set of related NE, a parser and a NER system. The first set of related NE can be obtained from various sources. Online encyclopedias are one of them, but this kind of data is also available from specialized databases in specific domains which relate terms, or domain-dependence NE. The prototype we developed showed interesting results in terms of precision for the discovering of new entities relations and we believe that the methodology we adopted can be applied for other kind of relations between named entities. A side effect of the method is that it produces paraphrase or pseudo-paraphrases patterns.

From this first experiment, we expect future work to be done in different directions:

- Apply the method for other kind of relations;
- Try to extend the generalization of the lexico-syntactic patterns by replacing the specific lemmas that anchored the syntactic relations by word-senses;
- Study the effectiveness of paraphrase patterns for paraphrase detection and generation.

[4] Authors are working on the Korean language and they only are interested in binary relations.

References

1. Abney S.: Parsing by Chunks. In Robert Berwick, Steven Abney and Carol Teny (eds.). Principle-based Parsing, Kluwer Academics Publishers.(1991)
2. ACE 2007. NIST 2007 Automatic Content Extraction Evaluation Official Results. http://www.nist.gov/speech/tests/ace/ace07/doc/ace07_eval_official_results_20070402.htm
3. Aït-Mokhtar S., Chanod, J.P., Roux, C.: Robustness beyond Shallowness: Incremental Dependency Parsing. Special issue of NLE journal (2002)
4. Brun, C., Hagège C.: Normalization and Paraphrasing Using Symbolic Methods, Proceeding of the Second International Workshop on Paraphrasing. ACL 2003, Sapporo, Japan (2003)
5. Hagège C., Roux C.: Entre syntaxe et sémantique: Normalisation de l'analyse syntaxique en vue de l'amélioration de l'extraction d'information. Proceedings TALN 2003, Batz-sur-Mer, France. (2003)
6. Iria J., Ciravegna Fabio.: Relation Extraction for Mining the Semantic Web. In proceedings Machine Learning for the Semantic Web, Dagstuhl, DE (2005)
7. Kim S., Yoon J., Yang J.: Kernel approachees for genic interaction extraction. Bioinformatics, Vol. 24, no. 1. pp. 118—126 (2008)
8. Lee Changki, Yi-Gyu Hwang, Myung-Gil Jang.: Fine-Grained Named Entity Recognition and Relation Extraction for Question Answering. Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. Amsterdam, The Netherlands (2007)
9. Levin, B.: English Verb Classes and Alternations – A preliminary Investigation. The University of Chicago Press. (1993)
10. Mel'čuk I.: Dependency Syntax. State University of New York, Albany, N.Y.: The SUNY Press. (1988)
11. Ruppenhofer, J., Michael Ellsworth, Miriam R. L. Petruck, Christopher R Johnson and Jan Scheffczyk. Framenet II: Extended Theory and Practice (2006)
12. Specia Lucia, Motta Enrico.: A hybrid approach for extracting semantic relations from texts. Proceedings of the 2nd Workshop on Ontology Learning and Population: Bridging the Gap between Text and Knowledge, July 2006, Sydney, Australia (2006)
13. Tesnière L,: Eléments de Syntaxe Structurale. Klincksiek Eds. (Corrected edition Paris 1969). (1959)
14. Van Landeghem S., Saeys Y., Van de Peer Y.: Extracting protein-protein interactions from text using rich feature vectors and feature selection. Proceedings of Third International Symposium on Semantic Mining in Biomedicine (MBM 08). pp. 54--84 (2008)

Evaluation of Named Entity Extraction Systems

Mónica Marrero, Sonia Sánchez-Cuadrado,
Jorge Morato Lara and George Andreadakis

Computer Engineering Department, University Carlos III of Madrid
Av. de la Universidad 30, 28911 Leganés (Madrid), Spain
mmarrero@inf.uc3m.es, ssanchec@ie.inf.uc3m.es, jmorato@inf.uc3m.es, gand@ie.inf.uc3m.es

Abstract. The suitability of the algorithms for recognition and classification of entities (NERC) is evaluated through competitions such as MUC, CONLL or ACE. In general, these competitions are limited to the recognition of predefined entity types in certain languages. In addition, the evaluation of free applications and commercial systems that do not attend the competitions has been lightly studied. Shallowly studied have also been the causes of erroneous results. In this study a set of NERC tools are assessed. The assessment of the tools has consisted of: 1) the elaboration of a test corpus with typical and marginal types of entities; 2) the elaboration of a brief technical specification for the tools evaluated; 3) the assessment of the quality of the tools for the developed corpus by means of precision-recall ratios; 4) the analysis of the most frequent errors. The sufficiency of the technical characteristics of the tools and their evaluation ratios, presents an objective perspective of the quality and the effectiveness of the recognition and classification techniques of each tool. Thus, the study complements the information provided by other competitions and aids the choice or the design of more suitable NER tools for a specific project.

Keywords: Named entity extraction, named entity recognition and classification, information extraction, named entity extraction tools.

1 Introduction

There is currently a wide variety of named entities (NE) recognition systems. Competitive events are organized for the evaluation of NERC systems, in which the ability of identification and classification of the entities existing in a corpus is analyzed. Nevertheless, the competitions normally establish certain limitations such as:

- They focus on a limited group of NE types. This feature is quite variable due to the ambiguity in the use of the term *Named Entity* depending on the different forums or events. In the case of the MUC conferences, NEs were considered *personal names*, *organizations*, *locations* and at a later stage, *temporal entities* and *measurements* [1]. On the other hand, the CONLL-2002/2003 conferences defined the categories *person*, *organism, localization* and *miscellaneous* [2, 3]. The latter (*miscellaneous*), includes proper names of different nature with different categories: *gentilics, project names, team names*, etc. Finally, the ACE

Received 10/11/08
Accepted 09/12/08
Final version 04/02/09

conferences, used categories such as *arms, vehicles* or *facilities* (4). ACE also incorporates temporal expressions, but as an independent task. In addition, as far as the NE typology is concerned, there are at least two hierarchies of entity types: BBN categories [5] and Sekine's extended hierarchy [6]. The proposed hierarchical structures fluctuate respectively between 64 and 200 types and subtypes.

- The underlying concept behind every entity varies amongst the different competitions. For example, the entity type *person* includes different subtypes depending on the competition (e.g. titles of person, personal pronouns etc.). These differences along with the mismatch of entity types analyzed in every competition impede their comparison.
- Conferences normally focus on specific languages. Frequently, the languages present in a conference vary from year to year. Currently, ACE is the most competitive and prestigious event that evaluates the NERC tasks. At the same time, it is the most ample as far as the idiomatic coverage is concerned (Arabic, Chinese, English and Spanish).
- The usability and technical characteristics of the software tools is not a factor considered in the competitions.
- The evaluation algorithms are different in each competition. Generally, the precision-recall ratios for the identification and classification of all the criteria considered in the competition are presented in a single measurement. Evaluation varies from the simplicity of CONLL to the complexity of ACE. In CONLL, partial identifications are not considered and false positive errors are not penalized. ACE evaluation [4] is based on a complex algorithm where different named entities are weighted with different weights, making difficult the interpretation and the comparison of results to those of other competitions [7].
- Results obtained in these competitions are conditioned to the manually tagged training and test corpora, which are provided to the participants.
- Large tagged corpora may favor tools that possess larger gazetteers but nevertheless, this does not imply a superior tool quality.
- Tools presented are not necessarily available commercially or for research.
- Various research groups and commercial systems are presented in these competitions establishing a ranking of tools. Thus, the evaluation of NERC systems is limited to those that participate in these competitions. With the current resources, the comparison with tools that fail to attend in these events seems to be impossible.

This article proposes a framework that permits the assessment of NER systems. It intends to provide an evaluation system to those applications which for one reason or another do not attend official competitions. Even in the cases of tools that do participate, this analysis will permit obtaining a complementary vision of their results.

2 Analysis and Methodology

2.1 Characteristics of the Evaluated NERC Tools

There are many operating tools that have been located through references in scientific work or commercial documentation. However, for this study we have defined the following criteria for selecting a NERC system.

1. The system has to permit the processing of texts which are not domain dependent
2. It has to work independently. In other words, it shouldn't require the user to provide resources necessary for its operation.
3. It should process texts in a common language, since language dependency limits the applied techniques. English has been the selected language for this assessment, because it is widely used and supported by the tools.

Tools, such as Trifeed [8], have been discarded for not fulfilling the necessary requirements, as it only accepts predetermined newspaper articles. Other popular tools of the biomedical domain such as AbGene [9], Abner [10] and BioNer [11] have also been discarded. In other cases, some tools have been eliminated due to their reduced efficiency or lack of maintenance. Finally, a couple of tools with good results in the competitions were not considered since they did not dispose a free version: EROCS by IBM and the NERC system by BBM technologies.

Consequently, NERC evaluation will be performed on the following tools: Supersense – Model CONLL, Supersence – Model WNSS, Supersense – Model WSI, Afner, Annie, Freeling, TextPro, YooName, ClearForest and Lingpipe. Freeling is considered as a NERC system, but in the case of the English language it does not perform classification. It has been included because, according to its characteristics and previous evaluations in recognition, it has been giving out moderate results.

- *LingPipe* [12] is a set of Java libraries developed by *Alias-I* for natural language processing. It is by default prepared for the detection and the classification of NEs such as persons, organizations and locations in the English language, but it is also possible to train it through a corpus for other languages. Additionally to the detection and the identification of entities, it is also offering additional functionalities such as orthographic correction and text classification in English. It offers a user interface and various demos through which it is possible to test texts. It is open-source and free of charge for research causes, but it is possible to purchase it for commercial use.
- *ClearForest SWS* [13] is a commercial tool made by *ClearForest Ltd.*, currently acquired by *Reuters*. It allows the analysis of English texts and the identification of *ENAMEX* types, in addition to some other types such as products, currencies, etc. A web service, partially based on this tool, has been made for the capture of entities: *Gnosis*, a free plug-in based on this tool for the *Mozilla Firefox* browser, captures numerous types of different entities in web pages. They also offer a Web API that may be used freely under certain conditions. Currently, it has evolved to a tool called *Calais*, which amongst other additional services it

permits the establishment of relationships amongst entities and the detection of events and roles.

- *Annie* [14] is an entity extraction module incorporated in the *GATE* framework. It is open-source and under a GNU license, developed at the University of Sheffield. It is implemented in Java and incorporates in the form of plug-ins and libraries its own or external resources for a variety of aspects related to natural language processing (i.e. Lucene, MinorThird, Google, Weka etc.). It can be used as an API but it also provides its own interface for an independent use. Annie also offers as a module a set of default resources (i.e. tokenizer, sentence splitter, POS tagging, co-reference resolution, gazetteers, etc.) that can be used in combination for the capture of entities. This set can be substituted by other plug-ins or even be disabled. The evaluation of the tool has been realized using its default resources, which are adapted for the English language.
- *Freeling* [15] is a tool developed in C++ at the *TALP Research Center* of the Polytechnic University of Catalonia. It is an open source tool with GNU license that may be used as an API or independently. There is also a Web demo where you can type text. It offers various services related to natural language processing, amongst which the detection of entities. It supports English, Spanish, Catalan, Galician, and Italian. The tool recognizes the usual entities of person, organisms and locations as well as quantities of various types and dates. It separates the identification activities to those of classification, and utilizes automatic learning as well as linguistic (dictionaries, Word-net, lists) and heuristic resources.
- *Afner* [16] is an open-source NERC tool, under GNU license, developed in C++ at the University of Macquaire. Currently it is used as part of a Question Answering tool called *AnswerFinder*, which is focusing to maximizing recall. Afner can also be used as an API for other applications or can be used independently. It uses lists, regular expressions and a supervised learning model which amongst other features, can report the entity's membership to a list or the entity's match with a regular expression. It also allows the addition of lists and regular expressions, as well as the training of new models. It is by default capable of recognizing persons' names, organizations, locations, miscellanea, monetary quantities, and dates in English texts.
- *Supersense Tagger* [17] is an open-source tagger developed in C++ with a version 2.0 Apache license. It is designed for the semantic tagging of nouns and verbs based on WordNet categories which include persons, organizations, locations, temporal expressions and quantities. It is based on automatic learning, offering three different models for application: CONLL, WSJ and WNSS. Given the differences in the tagging and the behavior amongst these three models, they have been considered independently in this study.
- *TextPro* tools suite [18] is developed in C++ at the *Center for Scientific Research and Technology (ITC-irst)*, in Trento, and offers various NLP functionalities interconnected in a pipeline order. It is under a GNU license and uses automatic learning and gazetteers. It is available for English and Italian and offers a web demo for both these languages.
- *YooName* [19] is a tool developed at the University of Ottawa by David Nadeau. It incorporates semi-supervised learning techniques applied to the web, that

permit the identification of entities using a predefined classification of nine types of NEs (person, organization, location, miscellanea, facility, product, event, natural element and unit) and 100 subtypes. There is a web version for doing demos where you can also type English texts in order to be analyzed. The tool also offers a blog with news and information related to its operation and other NER subjects (http://yooname.wordpress.com/).

The main characteristics of each tool are presented on Table 1. As can be observed, the majority of these are developed in C++, offering a console user interface and an API. With respect to the degree of computer usage dexterity that is needed in order to operate each tool, the majority of them have been classified as Advanced and just one of them as Simple (Simple, indicates that it is enough downloading and executing the respective file, and Advanced refers to a more complex process -i.e. additional libraries, compilations, expert configurations etc-). The dash (-) indicates that there was not any information available.

Table 1. Tool features

Tool	**Develop. Language**	**Interface**	**License**	**Simple (S) /Advanced (A) Installation**	**Demo**	**Entity types**
Supersense-CONLL	C++	Console/ API	Apache 2.0	A	No	4
Supersense-WNSS	C++	Console/ API	Apache 2.0	A	No	27
Supersense-WSJ	C++	Console/ API	Apache 2.0	A	No	> 100
Afner	C++	Console/ API	GNU	A	No	6
Annie	Java	Graphical /API	GNU	A	Yes	~12
Freeling	C++	Graphical /API	GNU	A	Yes	0
TextPro	C++	Console/ API	GNU	A	Yes	4
YooName	-	-	-	-	Yes	>100
ClearForest	-	Web/API	Commerc.	-	Yes	6
Lingpipe	Java	API	Free/Devel op./Startup	S	Yes	3

2.2 Methodology

The data analysis has been realized having a triple focus:

- Comparison of the tools' characteristics: task realized through a brief technical specification based on usability aspects.

- Comparison of results obtained by the tools, for entities found in the test corpus. This evaluation has been realized through distinct measures of precision – recall based on :
  - Identification of the entities and false positives in the identification
  - Classification of entities
  - Classification by NE types that each tool recognizes.
- Comparison of the tools according to the typographic, lexical, semantic or heuristic factors that has been considered in the entities recognition. This analysis has been realized with data mining classification algorithms. For doing this, information referring to all the nominal elements (entities or not) of the corpus has been introduced into the Weka [20] tool and analyzed with the PART algorithm to extract rules reflecting the behavior of each tool.

In particular, the typographic, lexical, semantic and heuristic features analyzed in the entity recognition processes are:

- Words at the first position of the phrase.
- Words written with the first letter in uppercase.
- Words in quotes.
- Words written totally in uppercase.
- Words written totally in lowercase.
- Polysemic words.
- Noun phrases
- Entities previously identified/classified in the text
- Possible use of:
  - Verb argument (based on semantic roles)
  - Trigger word based recognition
  - Gazetteer based recognition
  - Regular expressions

An English test corpus has been made containing all the above features in order to evaluate the behavior of the tools. It has a total of 579 words, distributed in 13 paragraphs in which more than 100 occurrences of various types of entities have been accumulated. Some of these NEs may be recognized and classified using gazetteers (e.g. Spain), and some others may be recognizable through trigger words (e.g. Inc., Co., Mr.). These entity types were distributed in various phrases in the corpus with different typography (dash, quotes, etc.), the relative position in different sentences, and the orthographic form (e.g. upper or lower case letters).

Invented NEs (*dontknowhere, dontknowho*) and polysemic entities (e.g. *Rose*) allows the verification of the use of NLP techniques. On the other hand, the recognition of fictional entities in lower case and with no special features or contextual information that could assist in their identification, shows the influence of pre-processing stages on the tools.

Finally, it must be taken in account that entities in a tool could neither totally coincide in number nor in semantic with their equivalent entities in other tools so the analysis has to be specialized for every tool (It's the corpus that should be adapted to the tools and not the tools to the corpus).

3 Results

3.1 General Results

The results obtained for each of the parameters considered in the evaluation are presented next. The charts of precision-recall for both identification (Fig. 1) and classification (Fig.2), present a performance which is generally over 50%. Exceptions are the precision and recall values of the Afner tool, and the recall values of the YooName tool. ClearForest stands out with its behavior for obtaining precision rates that exceed 90%. Other tools such as Supersense Tagger and Annie achieve inferior values, although they exceed 70% and seem to be more equilibrated in respect to their recall.

A detailed analysis should additionally take in account the false positive errors, i.e. the elements erroneously identified as entities, as this could result more damaging in a project than partial identification or erroneous classification. Therefore, the tools that obtain a greater number of false positive errors are Freeling and Annie, whilst WNSS model of SupersenseTagger does not identify erroneously any element as an entity.

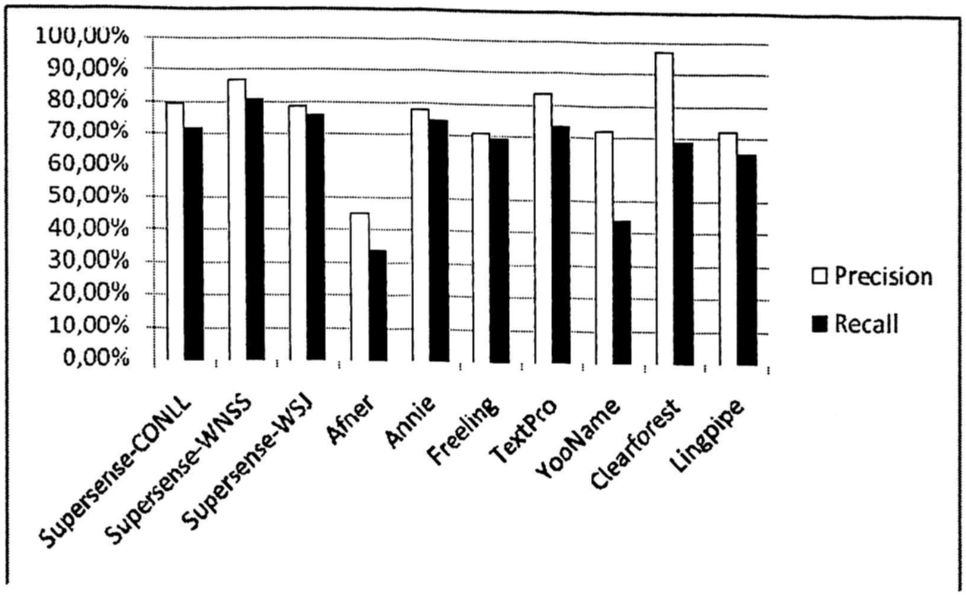


Fig. 1. Precision-Recall in entity identification

Given that classification is a process that depends on the identification of entities, the f-measure in identification is always superior to that of the classification's (Fig. 3). However it is generally observed that the values are similar. The most notable differences appear with the TextPro tool and to a lesser degree, with the WSJ Model of SupersenseTagger, which stand out in their identification processes but not in the classification of entities that have previously managed to identify.

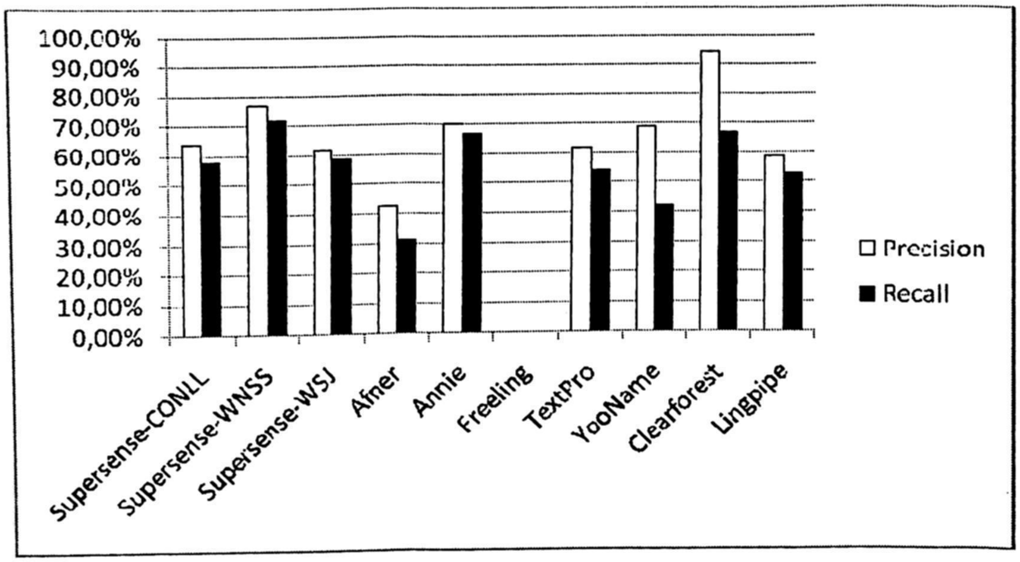


Fig. 2. Precision-Recall in entity classification (Freeling has not been evaluated in this process)

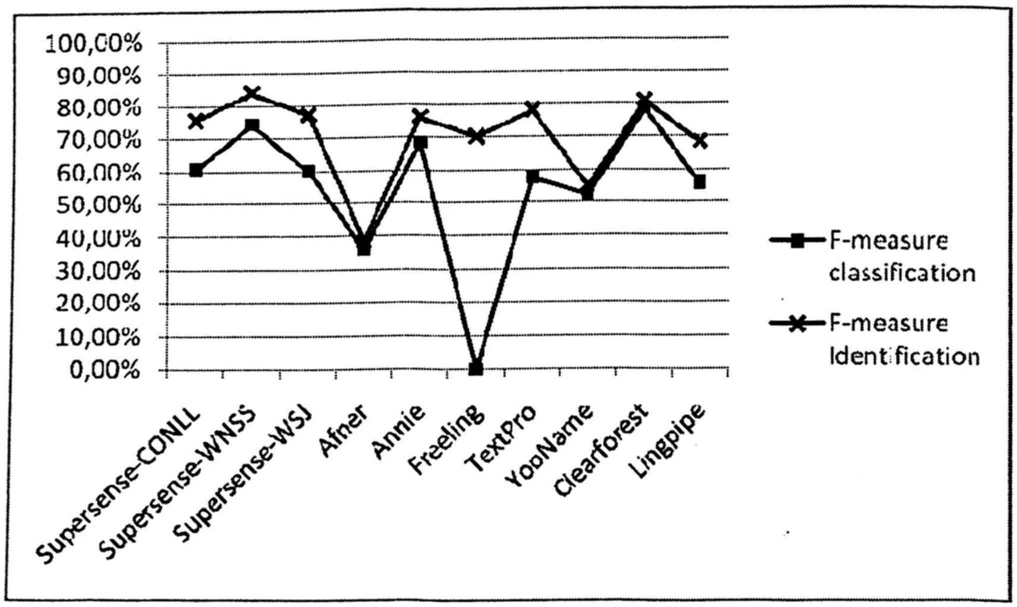


Fig. 3: F-measure in entity identification and classification (Freeling has not been evaluated in classification).

3.2 Results by Entity Type

The number of categories that each tool can recognize (Table 2) is an important factor for the evaluation of a tool. It is quite different having a tool able to recognize over one hundred different types of entities, to having a tool that can only recognize three.

However, the utility and difficulty of recognition of some types against some others is different, which demonstrates the need for a study based on the entity's types. In this case the study was carried out for each one of the entity types that the tool was able to recognize in the corpus. Thus, given the ambiguity relative to the term *entity* [21] and the lack of uniform use of tags, we have to previously analyze the precise significance of each tag in each tool and make them uniform.

The analysis illustrated in Table 2 allows us to observe some differences to the global analysis. Afner, which initially had worst results that the other tools, is performing better on person recognition than Supersense-WSJ or YooName. Additionally it is remarkable how YooName has an f-measure on the entity type *Company* of 0.08, whilst ClearForest achieves 0.95.

Table 2. Results by entity type

Tool	Entity type	N	F
Supers. CONLL	Person	32	0'63
	Location	43	0'64
	Org.	13	0'72
	Miscelanea	4	0
Supersense-WNSS	Person	38	0'65
	Location	48	0'78
	Group	25	0'88
	Time	9	0'66
	Quantity	9	0
	Food	6	1
	Communicat.	1	1
	Cognition	2	0'66
	Substance	1	0
	Relation	1	0
	Plant	6	1
	Object	1	1
	Other	1	1
Supersense-WSJ	Person	32	0'30
	Person-Desc.	5	1
	Geo-Pol.(other)	20	0'10
	Country	18	0'70
	State/Province	6	0'66
	Geo-Pol-Desc.	9	1
	Corporation	12	0'69
	Org.-Descrip.	12	0'86
	Date	10	1
	Money	4	0'28
	Food	7	1
	Ordinal	1	1
	Cardinal	7	0'92
YooName	Person	33	0'30
	Location	44	0'51
	Org.	13	0'88
	Vocation	4	1
	Country	21	0'66
	State/Prov.	4	0'75
	City	7	0'70
	Loc (other)	11	0
	Company	12	0'08
	Month	2	1
	Week Day	2	1
	Food	6	1
	Mineral	1	0
	Vegetal	1	1
Clear Forest	Person	53	0'72
	Country	19	0'97
	State/Prov.	6	1
	City	10	0'18
	Company	12	0'95
Text Pro	Person	32	0'59
	Location	44	0'51
	Org.	13	0'88
Lin gpipe	Person	32	0'67
	Location	47	0'47
	Org.	12	0'78
Afner	Person	32	0'50
	Location	44	0'36
	Org.	12	0
	Date	2	0'50

3.3 Inference

With the aid of Weka, inferences have been made about the behavior of the tools. The typographic, lexical, semantic and even contextual characteristics of every corpus element susceptible into been captured as an entity, have been annotated. Using an automatic learning algorithm (PART) applied to the results of each tool we have obtained rules that characterize the behavior of the tools in the combined task of identification and classification. Finally, we have applied this algorithm to the aggregate of results of all tools in order to detect common behavioral patterns. Those rules demonstrate the features most involved in the errors obtained by each tool during the processes of identification and classification.

One of the most important features seems to be the orthographic form of the entities: Supersense-CONLL, Supersense-WNSS, Afner, Annie and Freeling have remarkable problems in the recognition of entities written in lowercase, and Supersense-WSJ, Afner, Annie and Freeling have a significant number of false positives with words written totally in uppercase.

On the other hand, the existence of noun-phrase entities influences the errors committed by many tools: Supersense-WNSS, Afner, Annie, YooName and LingPipe have problems in the recognition of noun-phrase entities mainly when the typography or orthographic form of the terms in the noun-phrase, are different. The triggers work fine in all the tools except for Supersense-CONLL, which which does not seem to handle them well. Finally, the existence of polysemic entities is a handicap for all tools, but the rules make this handicap to stand out in the case of ClearForest. This does not necessarily mean that ClearForest performs worst with polysemic entities, but yet it is the only noticeable problem that this tool has.

4 Conclusion

An analysis of various NERC tools has been presented in this study. The evaluation proposes a model that eliminates some of the competitions' limitations into assessing these tools. This model is based on the creation of a small corpus, and the adaptation of the evaluation methodology to the NERC typology of the tools, not the contrary as it is common in the major competitions. The analysis of all the identified entities and the errors committed during this process permits a study using data mining in order to discover the most frequent errors in the identification and classification of NEs.

All the evaluated tools are oriented to experts who may integrate them in other systems. The major programming languages utilized are C++ and Java. The election of these languages could be related to their efficiency, portability, or their abundance in libraries.

At first sight as far as the performance of the recognition of entities is concerned, the NERC tools that performed best were Supersense-WNSS and Clearforest. It can be observed that the variety of entity types that the tools can recognize does not determine the results: tools that recognize the largest number of entities, such as Supersense-WSJ or YooName, do not achieve very good results; on the other hand, the lowest ratios are achieved by Afner, which recognizes a few different entity types.

In other words, an important factor in the evaluation of the different systems is not only the number of different entity types recognized but also their "quality". Metrics presenting the average performance in the identification of entity types is not always representative of its success. The performance of every tool in the identification of individual entity types should be examined in order to extract better conclusions.

The errors committed by all tools have been analyzed using data mining in order to determine which could be their cause and identify common patterns. This information was rarely analyzed in the competitions. The most common difficulties and the deficiencies detected in NERCs denote a handicap in the management of noun phrases and reveal a strong dependency on gazetteers. Tools that focus on gazetteers (as in the case of Afner and YooName) seem to produce poor results. This deficiency seems to be due to the scarce importance given to context analysis. Another deficiency is the lack of a preprocessing stage during which the tools could acquire knowledge useful in the tagging of ambiguous entities. This may lead to the failure of identifying an entity that previously has been successfully recognized (TextPro, LingPipe). An exception to this was YooName, although in this case, if the typography of the same entity through the corpus is different, this tool can conclude that it is not an entity.

NERC systems present an elevated dependency to uppercase characters, not being able to recognize the same entity if written in lowercase, even though it does with other typographic elements such as quotes. The management of dashes (-) and full stops (.) can significantly influence the recognition process, separating parts of a multi-word entity or uniting terms of different entities even if those are located in different paragraphs and are separated by full stops.

Moreover, the results point to semantic problems such as the inability of the NERCs to recognize polysemic entities and the inconsistency in the detection of cardinal or ordinal types, which are only recognized when they are written numerically but not when written alphabetically.

The techniques that have given the best results in the experiment have been the consideration of linguistic information (in the case of Supersense-WNSS), and the triggers (in the case of Supersense-CONLL). On the other hand, being limited to typography and gazetteers does not seem to improve results (Afner). The identification seems to be based mostly on the capitalization of words (Freeling, Annie).

Acknowledgements. This work was carried out within the framework of a research project (Ref. CCG07-UC3M/TIC-3401) financed by the Comunidad Autonoma de Madrid (CAM) and University Carlos III of Madrid (UC3M). We acknowledge in particular Raúl Matallanos to realize previous studies and test about evaluated systems.

References

1. Grishman, R., Sundheim, B.: Message Understanding Conference-6: A Brief History. In: Proc. 16th Conference on Computational Linguistics, vol. 1, pp. 466—471. ACL: NJ (USA) (1996)

2. Tjong Kim Sang, Erik. F. Introduction to the CoNLL-2002 Shared Task: Language-Independent Named Entity Recognition. In: Proc. Conference on Natural Language Learning, pp. 155--158. Taipei, Taiwan (2002)
3. Tjong Kim Sang, Erik. F.; De Meulder, F. 2003. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In Proc. Conference on Natural Language Learning. pp. 142-147. Edmonton, Canada (2003).
4. NIST: ACE08 Evaluation Plan. http://www.nist.gov/speech/tests/ace/2008/doc/ace08-evalplan.v1.2d.pdf (2008)
5. Brunstein, A. Annotation Guidelines for Answer Types" BBN technologies (2002). http://www.ldc.upenn.edu/Catalog/docs/LDC2005T33/BBN-Types-Subtypes.html
6. Sekine, S., Sudo, K., Nobata, C.: Extended named entity hierarchy. In: Proc. of the Third International Conference on Language Resources and Evaluation (LREC-2002), pp. 1818--1824. Las Palmas (SP) (2002)
7. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. Linguisticae Investigationes, vol 30, n.1, pp. 3--26 (2007)
8. Trifeed http://www.trifeed.com/
9. AbGene ftp://ftp.ncbi.nlm.nih.gov/pub/tanabe/AbGene/
10. Abner http://pages.cs.wisc.edu/~bsettles/abner/
11. BioNer. POSBioTm Postech Biological Text-Mining system http://isoft.postech.ac.kr/Research/BioNER/POSBIOTM/NER/main.html
12. Baldwin, B., Carpenter, B.: LingPipe. http://www.alias-i.com/lingpipe/
13. ClearForest Semantic Web Services (SWS), http://sws.clearforest.com/
14. Cunningham, H., Maynard D., Tab-lan V., Ursu C., Bontcheva K.: Developing Language Processing Components with GATE. GATE v5 User Guide, http://www.gate.ac.uk/sale/tao/tao.pdf
15. Atserias J., Casas B., Comelles E., González M., Padró Ll., Padró M.: FreeLing 1.3: Syntactic and semantic services in an open-source NLP library. In: Proceedings of the fifth international conference on Language Resources and Evaluation. ELRA , Paris (2006).
16. Zaanen M., Molla D.: A named entity recogniser for question answering. In: Proceedings PACLING (2007)
17. Ciaramita, M., Altun, Y.: Broad-coverage sense disambiguation and information extraction with a supersense sequence tagger. In: Proceedings of EMNLP-06, pp. 594--602, Sydney, Australia, 2006
18. Pianta, E., Girardi, C. and Zanoli, R.: The TextPro tool suite, Proceedings of Sixth International Language Resources and Evaluation. ELRA, Paris (2008)
19. Nadeau, D., Turney, P.; Matwin, S.: Unsupervised Named Entity Recognition: Generating Gazetteers and Resolving Ambiguity. In Proc. Canadian Conference on Artificial Intelligence (2006)
20. Witten, I.H., Franck, E., Trigg, L., Hall, M., Holmes G., Cunningham S.J.: Weka: Practical machine learning tools and techniques with Java implementations, Proc. ANNES'99 International Workshop on emerging Engineering and Connectionnist-based Information Systems, pp. 192–196 (1999).
21. Borrega, O, Taulé, Mariona and Martí, Mª. Antonia. What do we mean when we speak about Named Entities? Proceedings of the Corpus Linguistics Conference, CL2007. University of Birmingham (2007)

Building a Brazilian Portuguese Parallel Corpus of Original and Simplified Texts

Helena M. Caseli, Tiago F. Pereira, Lucia Specia, Thiago A. S. Pardo, Caroline Gasperin and Sandra M. Aluisio

Center of Computational Linguistics (NILC)/ Department of Computer Sciences, University of São Paulo, Av. Trabalhador São-Carlense, 400. 13560-970 - São Carlos/SP, Brazil
helenacaseli@dc.ufscar.br, tiagofrepereira@yahoo.com.br, lspecia@icmc.usp.br, taspardo@icmc.usp.br, cgasperin@icmc.usp.br, sandra@icmc.usp.br

Abstract. In this paper we address the problem of building the necessary tools and resources for performing Brazilian Portuguese text simplification. We describe our efforts on the design and development of: (a) a XCES-based annotation schema, (b) an annotation edition tool, and (c) a portal to access parallel corpora of original-simplified texts. These contributions were intended to (i) allow the creation and public release of a corpus of original and simplified texts with two different versions of simplification (called here *natural* and *strong*), targeting two levels of functional illiteracy and (ii) register simplification decisions during the creation of such corpus. We also provide an analysis of the first corpus created using the resources presented here: 104 newspaper texts and their simplified versions, produced by an expert in text simplification.

Keywords: Text Simplification, Brazilian Portuguese, annotation standards, annotation edition tool.

1 Introduction

In Brazil, "letramento" (literacy) is the term used to designate people's ability to use written language to obtain and record information, express themselves, plan and learn continuously [1]. In Brazil, according to the index used to measure the literacy level of the population (*INAF - National Indicator of Functional Literacy*), a vast number of people belong to the so called *rudimentary* and *basic* literacy levels. These people are able to find explicit information in short texts (rudimentary level) and also process slightly longer texts and make simple inferences (basic level).

The PorSimples project (*Simplificação Textual do Português para Inclusão e Acessibilidade Digital*)[1] aims at producing text simplification tools for promoting digital inclusion and accessibility for people with such levels of literacy, and possibly other kinds of reading disabilities. More specifically, the goal is to help these readers to process documents available on the web. Additionally, it could help children learning to read texts of different genres or adults being alphabetized. Two tools are

[1] http://caravelas.icmc.usp.br/wiki/index.php/Principal

Received 10/11/08
Accepted 11/12/08
Final version 05/02/09

envisioned: (1) a browser plugin, which automatically simplifies texts on the web for the end-user, and (2) an authoring tool, which supports authors in the process of producing simple texts. The focus is on texts published in government sites or by relevant news agencies, both expected to be of importance to a large audience with various literacy levels. The language of the texts is Brazilian Portuguese, for which there are no text simplification systems, to the best of our knowledge.

The project follows three main text processing strategies to produce simplified texts: (i) text summarization, (ii) highlighting of the text structure/organization, named entities and verb-argument structure, aiming to provide visual and explanatory information about important concepts appearing in the text, and mainly (iii) text simplification itself, which includes operations at the lexical, syntactic and discourse levels. The simplification operations proposed in the project aim to preserve most of the information in the input text, and thus the deletion of a sentence or parts of it was rarely adopted. For that reason, summarization techniques play an important role.

Text simplification has been exploited in other languages for helping poor literacy readers [2], [3] and [4] and special kinds of readers such as aphasics [5]. It has also been used for improving the accuracy of other Natural Language Processing (NLP) tasks [6] and [7], like parsing. One important step towards building text simplification tools is the analysis and comparison of general-use, non-simplified texts, with their corresponding simplified versions, that is, a parallel corpus of original-simplified texts. This allows investigating which kinds of changes should be applied, what resources are necessary to allow them, and how to evaluate the simplification task. Moreover, such a corpus can be directly used with statistical techniques to learn simplification rules.

A corpus of original and manually simplified sentences has been created for English but it is no longer available [8]. However, such a resource does not contain any explicit information about how and why the simplifications were performed, and therefore only limited learning from this corpus is possible. Two other studies have used parallel aligned corpus of original and simplified English texts. [9] uses parallel corpora of TV program transcripts and subtitles (documentaries and talk shows broadcasted by the BBC World Service) to automatically generate subtitles for hearing-impaired people. [10] uses a corpus of original news articles with corresponding abridged versions developed by Literacyworks[2] to aid teachers by automatically proposing ways to simplify texts.

Such parallel corpora of original and simplified texts do not exist for Portuguese. Moreover, given the differences between the two languages, a parallel corpus of English simplifications would not be appropriate. So, in the scope of the PorSimples project we have: (1) built a parallel corpus of original and simplified texts for Brazilian Portuguese, (2) developed a tool to assist human annotators in this inherently manual task — the Simplification Annotation Editor[3] — and (3) specified a new schema for representing the original-simplified information, based on the XCES standard[4]. The parallel corpora resulting from the simplification process can be queried in a public Portal of Parallel Corpora of Simplified Texts[5].

[2] http://literacynet.org/cnnsf/index cnnsf.html

[3] http://caravelas.icmc.usp.br/anotador/

[4] http://www.xml-ces.org

[5] http://caravelas.icmc.usp.br/portal/index.php

The Simplification Annotation Editor facilitates the manual simplification task, by guiding the annotator and providing the necessary linguistic resources, besides recording the simplification operations made by the annotator. Moreover, as a consequence, it guarantees the consistency of the annotated corpora. The annotation process, on the other hand, also helps our understanding of the simplification task which can bring improvements to the tool, making it more comprehensive and compact..

This paper is organized as follows. In Section 2 we present the background and technologies related to this work. In Section 3 we describe the Simplification Annotation Editor and the Portal of Parallel Corpora of Simplified Texts, which shows all the simplification decisions taken in the annotation process for a given corpus. We also describe our XCES-based schema proposed to annotate simplification operations and present some statistics on a parallel corpus built using the Editor. In Section 4 we discuss some final remarks and present directions for future work.

2 Background and Related Work

2.1 Support Tools for Text Annotation and Simplification Editors

Text annotation is the process of adding new information to existing language data/corpora [11]. This is an inherently manual task, but it can be supported by tools. Some tools, such as GATE[6] and its several plugged-in systems, were developed to automatically annotate a corpus. MMAX (MultiModal Annotation in XML), another linguistic annotation tool, allows multi-level annotation of (potentially multi-modal) corpora [11]. Although very useful for several applications, the existing tools could not be used in for our purposes. GATE would require a system to be developed from scratch and MMAX is not able to specify the relations between different texts - the original and the simplified -, an essential piece of information in the text simplification annotation process.

There are also tools called *simplification editors*, such as SIMPLUS[7] and StyleWriter[8]. SIMPLUS is a generic tool for helping writing simplified (or controlled) English. Simplified English implies the use of limited vocabulary of Standard or Plain English words and restricted sentence structure. StyleWriter has also features to help users to write using Plain English. It guides the user on how to produce a well-written English text and also focus on simplifying and clarifying such text. Some simplification features present in these previous tools are included in our editor. However, instead of helping authors to write simple texts, currently, our editor is intended to support the building of a parallel corpus of original-simplified texts to be used in corpus-driven approaches to text simplification. Therefore, besides the result of the simplification process, we need also to record the simplification operations that were performed. Other motivations for creating our own editor are that it is intended

[6] http://gate.ac.uk/

[7] http://www.linguatechnologies.com

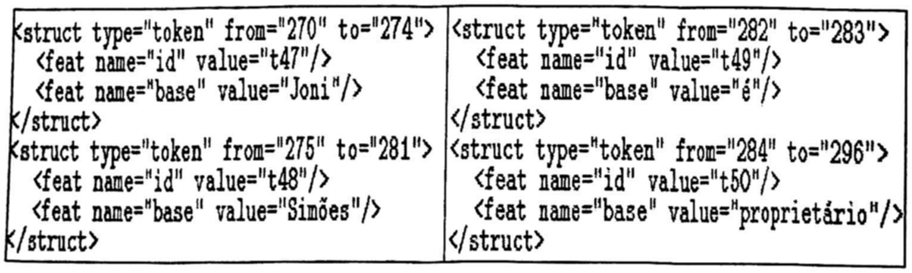
[8] http://www.editorsoftware.com/writing-software

to be freely available to the research community and to evolve with the project, ultimately becoming a text simplification editor itself.

2.2 XCES

XCES is a corpus encoding standard in which the source documents are plain texts and all the annotations are stored in stand-off XML[9] documents [12]. The stand-off format for annotations is a graph representation in which the nodes are virtually placed between the characters in the plain text and the edges define regions between nodes, represented by XML annotations which are associated with feature structures [13]. For example, Figure 1 shows an excerpt of a stand-off annotation document containing the tokens of the Portuguese sentence in (snt$_1$). In this example, each <*struct*> element represents an edge in the graph and the values specified by the *from* and *to* attributes are the nodes in the source text document over which the edge spans. For example, the first token, "*Joni*" spans from node 270 (placed before character 'J') to node 274 (placed after character 'i') in the text document. The <*feat*> elements allow specifying any other relevant information about the element, such as its identifier and the actual word it represents.

(snt$_1$) *Joni Simões é proprietário de uma empresa da Capital que vende equipamentos de DVD. (Joni Simões owns a company in the capital which sells DVD devices).*



```
<struct type="token" from="270" to="274">
  <feat name="id" value="t47"/>
  <feat name="base" value="Joni"/>
</struct>
<struct type="token" from="275" to="281">
  <feat name="id" value="t48"/>
  <feat name="base" value="Simões"/>
</struct>
```

```
<struct type="token" from="282" to="283">
  <feat name="id" value="t49"/>
  <feat name="base" value="é"/>
</struct>
<struct type="token" from="284" to="296">
  <feat name="id" value="t50"/>
  <feat name="base" value="proprietário"/>
</struct>
```

Fig. 1. Excerpt of a stand-off XCES annotation document

XCES has been used in projects involving both only one language, e.g.: American National Corpus (ANC)[10] (English) and PLN-BR[11] (Brazilian Portuguese); and multiple languages as parallel data, e.g.: CroCo[12] (English-German) and Swedish-Turkish [14]. However, to our knowledge, PorSimples is the first project to use XCES to encode original-simplified parallel texts and also the actual simplification operations. Two annotation layers have been added to the traditional stand-off annotation layers, in order to store the information related to simplification.

In our XCES schema, each plain text document is related to at most other eight annotation documents, which contain the following information: (1) the header (specifies the origin of the document content and the stand-off annotation files), (2)

[9] http://www.w3.org/XML/

[10] http://americannationalcorpus.org

[11] http://www.nilc.icmc.usp.br/plnbr

[12] http://fr46.uni-saarland.de/croco/index_en.html

the logical division (markup of the structure of the document), (3) the sentences (markup of the sentence boundaries), (4) the tokens, (5) the part-of-speech of the tokens, (6) the syntactic chunks (phrases), (7) the alignment between original and simplified sentences, and (8) the simplification operations performed to transform the original sentences into simplified sentences. The first five files follow the same formats of ANC and PLN-BR corpora. The sixth file is particularly important to build syntactic simplification systems both rule-based and statistical ones. The last two files also follow the XCES guidelines but were created specifically for this project (see Section 3.2).

2.3 The Use of Corpus for Text Simplification

Parallel corpora of original and simplified texts can be used for automatic text simplification considering: (1) the information obtained from the annotation process, and (2) the final result of this process (the actual annotated corpus). The first refers to the insights about the range of operations performed in order to simplify a text. These insights can guide the specification of a comprehensive and consistent set of simplification rules for rule-based simplification systems. The second refers to the several ways the parallel corpus can be used to design automatic text simplification systems by means of statistical or machine learning techniques.

[8] investigates the automatic induction of syntactic simplification rules from a parallel corpus. Syntactic correspondences are extracted and generalized into rules, for example, replacing words by variables. The work only covered isolating relative clauses and no evaluation was provided. [9] applies a case-based learning algorithm to a parallel corpus, focusing on the summarization of subtitles by the removal of elements and lexical substitution. A very low performance was reported and the system seems to make serious mistakes, such as removing the subject of the sentences. Both corpora developed in such investigations aim at the simplification of English texts. Details about the creation of these corpora are not discussed in the published materials, but since fewer simplification operations were covered, as compared to our set of operations, we believe that such a process was simpler. It appears that no tool was designed to help the annotators.

[3] and [10] present a detailed corpus analysis of original and manually simplified news articles aiming at learning how people simplify texts in order to develop better automatic tools. They focus on the features of sentences that are split and on position and redundancy information in decisions about which sentences to keep and which to drop. However, they did not develop a simplification system based on the outcome of the corpus analysis; instead they used the syntactic simplifier of [4].

We believe that with a well designed and appropriately annotated corpus of original-simplified texts, covering enough examples of the simplification operations aimed by the PorSimples project, we will be able to further investigate the learning techniques which can be applied (and most likely adapted) to this application.

3 Text Simplification Annotation in the PorSimples Project

3.1 The Annotation Editor and the Portal of Parallel Corpora of Simplified Texts

As described in Section 1, readers with literacy at basic level may need different type of help from those with literacy at rudimentary level, and the same goes to children learning to read or people with cognitive disabilities. To attend the needs of people with different levels of literacy, we propose two subsets of simplifications called *natural* and *strong* simplifications. In our annotation tool, when performing a natural simplification, the annotator is free to choose which operations to use, among the ones available, and when to use them; there may be cases where the annotator decides not to simplify a sentence. Strong simplification, on the other hand, is driven by explicit rules from a manual of syntactic simplification also developed in the project [15] and [16], which state when and how to apply the simplification operations. Table 1 shows examples of an original text from an on-line Brazilian newspaper (translated here from Portuguese) in (a), its natural simplification in (b) and its strong simplification in (c). Clearly, the sentence in (b) can be further simplified if broken in shorter ones, as shown in (c). Although (c) may look less cohesive and somehow redundant, it can be useful for people with very low literacy levels [17].

Table 1. An example of an original text (a) and its simplified versions (b and c)

A	*In a press conference called to answer corruption charges during his term as Mayor of the city of Ribeirão Preto, Minister Antonio Palocci Filho (Treasury) said he made his position available, but with the recommendation of President Luiz Inácio Lula da Silva, would remain in government.*
B	*Minister Antonio Palocci (Treasury) said in a press conference that he will leave his position, although President Lula advised him to remain in the government.*
C	*Minister Antonio Palocci is the Treasury Minister. Antonio Palocci said in a press conference that he will leave his position. But he said that President Lula advised him to remain in the government.*

The Simplification Annotation Editor was used by the human annotator to create the parallel corpus following the 3-step architecture shown in Figure 2.

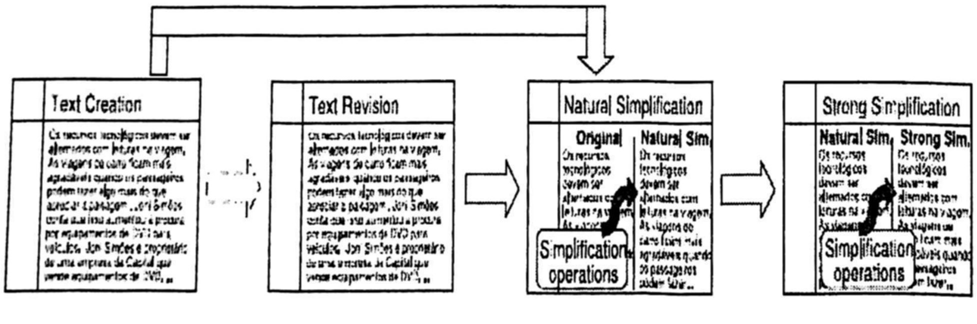


Fig. 2. Architecture of the Simplification Annotation Editor

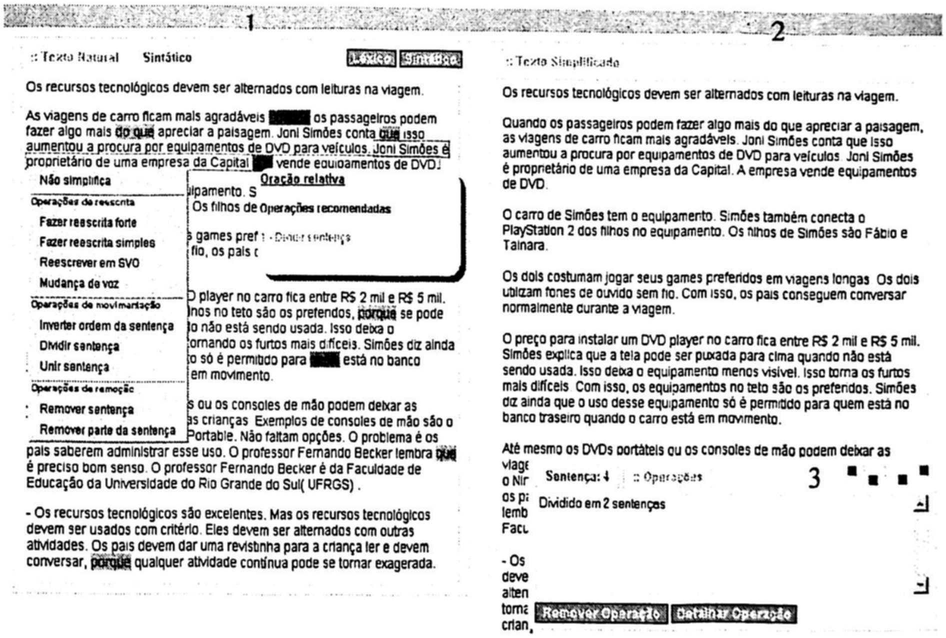
In the first step, the source text (original version) is created (or simply opened from a file) and possibly revised. In the revision step, the human annotator may manually correct punctuation and spelling mistakes. In the second step, natural simplifications are produced and logged, and from these the strong simplifications are generated (step3) (this sequence, first natural then strong, is not enforced in the Editor, that is, it allows strong simplifications from the original text as well). All the text versions (original, revised, natural and strong simplified) are stored in a database (DB).

To explain how the annotation is performed by a human using the Editor, consider the simplification example presented in Figure 3. This figure shows a screenshot of the Editor in the strong simplification step. As the numbers in Figure 3 show, the editor has three main areas: (1) the text being simplified, (2) the simplified version being produced, and (3) the log of simplification operations performed so far. In Figure 3, it is registered that the fourth original sentence, shown here in (snt$_1$) ("*Sentença: 4*") was divided in 2 sentences, as shown in snt$_2$ and snt$_3$).

(snt$_2$) *Joni Simões é proprietário de uma empresa da Capital (Joni Simões owns a company in the capital).*

(snt$_3$) *A empresa vende equipamentos de DVD (The company sells DVD devices).*

The simplification operations that can be applied encompass lexical and syntactic modifications and are performed for each original sentence separately. The syntactic operations, which are accessible via a pop-up menu, are the following: (1) non-simplification; (2) simple or (3) strong rewriting (as defined in [10]); (4) putting the sentence in its canonical order (subject-verb-object); (5) putting the sentence in the active voice; (6) inverting the clause ordering; (7) splitting or (8) joining sentences; (9) dropping the sentence or (10) dropping parts of the sentence. The lexical operations consist in replacing words found to be complex by simpler synonyms.



Os recursos tecnológicos devem ser alternados com leituras na viagem.

As viagens de carro ficam mais agradáveis [redacted] os passageiros podem fazer algo mais do que apreciar a paisagem. Joni Simões conta que isso aumentou a procura por equipamentos de DVD para veículos. Joni Simões é proprietário de uma empresa da Capital [redacted] vende equipamentos de DVD.

...ipamento. S
Os filhos de
...s games pref
fio, os pais c

...D player no carro fica entre R$ 2 mil e R$ 5 mil.
...nos no teto são os preferidos, porque se pode
...o não está sendo usada. Isso deixa o
...ornando os furtos mais difíceis. Simões diz ainda
...o só é permitido para [redacted] está no banco
em movimento.

...s ou os consoles de mão podem deixar as
...as crianças Exemplos de consoles de mão são o
...Portable. Não faltam opções. O problema é os pais saberem administrar esse uso. O professor Fernando Becker lembra que é preciso bom senso. O professor Fernando Becker é da Faculdade de Educação da Universidade do Rio Grande do Sul(UFRGS) .

- Os recursos tecnológicos são excelentes. Mas os recursos tecnológicos devem ser usados com critério. Eles devem ser alternados com outras atividades. Os pais devem dar uma revistinha para a criança ler e devem conversar, porque qualquer atividade contínua pode se tornar exagerada.

Os recursos tecnológicos devem ser alternados com leituras na viagem.

Quando os passageiros podem fazer algo mais do que apreciar a paisagem, as viagens de carro ficam mais agradáveis. Joni Simões conta que isso aumentou a procura por equipamentos de DVD para veículos. Joni Simões é proprietário de uma empresa da Capital. A empresa vende equipamentos de DVD.

O carro de Simões tem o equipamento. Simões também conecta o PlayStation 2 dos filhos no equipamento. Os filhos de Simões são Fábio e Tainara.

Os dois costumam jogar seus games preferidos em viagens longas. Os dois utilizam fones de ouvido sem fio. Com isso, os pais conseguem conversar normalmente durante a viagem.

O preço para instalar um DVD player no carro fica entre R$ 2 mil e R$ 5 mil. Simões explica que a tela pode ser puxada para cima quando não está sendo usada. Isso deixa o equipamento menos visível. Isso torna os furtos mais difíceis. Com isso, os equipamentos no teto são os preferidos. Simões diz ainda que o uso desse equipamento só é permitido para quem está no banco traseiro quando o carro está em movimento.

Até mesmo os DVDs portáteis ou os consoles de mão podem deixar as
viage...
o Nir...
os pa...
lemb...
Facu...

- Os
deve...
alten...
torna...
crian...

Sentença: 4

Dividido em 2 sentenças

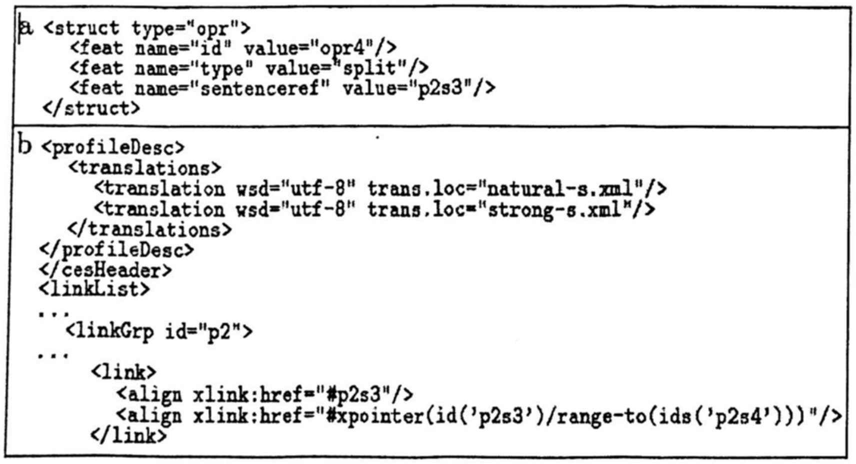
Fig. 3. Screenshot of the Simplification Annotation Editor (in the *Sintático* mode)

The Annotation Editor has two modes to assist the human annotator: the *Léxico* and the *Sintático* modes. In the *Léxico* mode, the editor proposes changes in words and discourse markers by simpler and/or more frequent ones. The annotator decides whether to accept or not the suggestions to simplify the highlighted words. Lexical simplifications are performed based on two linguistic resources: (1) a list of simple words and (2) a list of discourse markers. The first list is composed of words supposed to be common to youngsters, extracted from [18], frequent words from news texts for children, and concrete words [19]. The discourse markers were extracted from [20]. The *Sintático* mode proposes the 10 previously mentioned syntactic operations based on syntactic information provided by a parser for Portuguese [21]. As an example, in Figure 3, the system recommends (in the recommendation box) splitting snt_1 ("*1- Dividir sentença*"), since it has a relative clause (introduced by the relative pronoun "*que*"). This operation can be either selected from the recommendation box or from the pop-up menu. When chosen, the operation is recorded (area (3) of Figure 3) and for each simplification operation it is possible to specify (in "*Detalhar operação*") what has been changed in the simplified version.

The resulting parallel corpus can be queried in the Portal of Parallel Corpora of Simplified Texts, which shows all the simplification operations performed. For example, one can recover all the original sentences that were split during simplification or see all the lexical substitution pairs composed of complex and simple words. The Portal also makes available the XCES annotation and the resources that were used, including the dictionaries of simple words and discourse markers. It allows searching the corpus for the original and simplified texts, the alignment between such texts, the syntactical constructions that were considered in the project, and the actual texts that underwent the simplification operations.

3.2 The XCES Output

The output of the simplification process consists of eight XCES files, as described in Section 2.2.



```
a <struct type="opr">
     <feat name="id" value="opr4"/>
     <feat name="type" value="split"/>
     <feat name="sentenceref" value="p2s3"/>
  </struct>
```

```
b <profileDesc>
     <translations>
        <translation wsd="utf-8" trans.loc="natural-s.xml"/>
        <translation wsd="utf-8" trans.loc="strong-s.xml"/>
     </translations>
  </profileDesc>
  </cesHeader>
  <linkList>
  ...
     <linkGrp id="p2">
  ...
        <link>
           <align xlink:href="#p2s3"/>
           <align xlink:href="#xpointer(id('p2s3')/range-to(ids('p2s4')))"/>
        </link>
```

Fig. 4. Output XCES files for the example in Figure 3

Figure 4 shows excerpts of the two new files that were added in this project: (a) the simplification operations and (b) the alignment between natural and strong simplified sentences.

In Figure 4-a, one simplification operation is performed in the sentence identified as p2s3: the operation split. Figure 4-b shows that there is an alignment between p2s3 in natural-s.xml (the XCES file with the natural simplified sentences) and p2s3 and p2s4 in strong-s.xml file (the XCES file with the strong simplified sentences).

In order to align the sentences from the original and simplified versions of the text, we define a cardinality property for each operation, that is, how many sentences should be produced by such operation. The operation of joining sentences has cardinality -1; dropping one sentence has cardinality 0; sentence splitting requires asking the annotator for such cardinality, since different numbers of new sentences may be produced; for all other operations, the cardinality is 1. The cardinality information is used to generate links among original and simplified sentences.

3.3 The Parallel Corpus of Original and Simplified Versions

The first corpus simplified in the PorSimples project is composed of 104 texts from the *Zero Hora* newspaper. These texts were selected because they had a corresponding simplified version, also published in that newspaper, meant to be read by children. Therefore, this parallel corpus can also be useful to evaluate the proposed simplification operations for automatically generating newspaper versions for children. The corpus was simplified by a linguist, expert in text simplification, with the help of the Simplification Annotation Editor, which has been considered user-friendly by the annotator.

Table 2 shows the total number of sentences and words and the average sentence length (in words) of the original, natural and strong simplified texts. The last column shows the percentage of change in the numbers from original texts to strong simplifications. A considerable reduction happened with respect to individual sentence lengths. The overall text length is longer than the original, which was expected, as simplification usually yields the repetition of information in different sentences, particularly when splitting operations are performed. In the PorSimples project, we also provide summarization tools to shorten the texts, as part of the simplification process.

Table 2. Statistics on the original, natural and strong corpora

	Original	Natural	Strong	Change from original to strong
Number of sentences	2,116	3,104	3,537	+ 67.15%
Number of words	41,897	43,013	43,676	+ 4.24%
Average sentence length	19.8	13.85	12.35	- 37.63%

Tables 3 and 4 show the number of sentences, the percentage of sentences with respect to the input texts (original and natural, respectively), and the average sentence length (in words) after the simplifications from *original to natural*, and from *natural to strong*, focusing on two aspects: the types of operations applied and the syntactic phenomena addressed. The total number of sentences in the original corpus was 2,116, with an average sentence length of 19.8 words. The natural simplified corpus resulted in 3,104 sentences, with an average sentence length of 13.86 words. As mentioned before, the number of sentences increases with simplification, but these sentences are usually shorter.

Table 3. Statistics on the simplification operations

Syntactic and Lexical Simplification Operations	Number of sentences / (%) / Average sentence length					
	Original to Natural			Natural to Strong		
Non-simplification	418	19.75%	13.1	2,220	71.52%	11.86
Strong rewriting	7	0.33%	19.85	4	0.13%	14.5
Simple rewriting	509	24.05%	21.91	313	10.0%	16.95
Subject-verb-object ordering	31	1.46%	25.06	13	0.42%	14.15
Transformation to active voice	89	4.21%	22.12	65	2.09%	18.95
Inversion of clause ordering	191	9.03%	22.36	74	2.38%	18.89
Splitting sentences	723	34.17%	26.80	380	12.24%	23.58
Joining sentences	5	0.24%	10.83	6	0.19%	18.33
Dropping one sentence	6	0.28%	11	3	0.09%	5.3
Dropping sentence parts	241	11.39%	26.20	49	1.58%	22.20
Lexical Substitution	980	46.31%	23.46	196	6.34%	18.01

In Table 3, only the "Non-simplification" and "Dropping one sentence" operations are exclusive. The other operations can be combined in one sentence. In the natural simplification process, the most common operation is lexical simplification, followed by splitting sentences, dropping parts of the text, and changing discourse markers by simpler and/or more frequent ones. Strong simplifications (from natural simplifications) prioritize splitting sentences and lexical substitution. The higher number of non-simplification operations in the strong simplification process is due to the fact that most of the sentences had already been simplified in the natural simplification process.

Table 4. Statistics on the syntactic phenomena

Syntactic Phenomena	Number of sentences / (%) /Average sentence length					
	Original to Natural			Natural to Strong		
Apposition	196	9.26%	28.48	54	1.74%	22.20
Coordinate Clauses	806	38.09%	25.31	801	25.80%	18.9
Passive Voice	198	9.35%	26.06	146	4.70%	18.4
Relative Clauses	521	24.62%	25.43	412	13.27%	20.22
Subordinate Clauses	452	21.36%	25.5	524	16.88%	20.03

As shown in Table 4, certain syntactic phenomena are more frequent than others, and therefore many more simplification operations on sentences containing those types of phenomena were performed. The most frequent ones are coordinate, relative and subordinate clauses. These are in general the most difficult cases to simplify, according to studies performed in our project, and we consider this as an additional motivation for the construction of tools to support the simplification process.

4 Conclusions and Future Work

In this paper we have presented a Simplification Annotation Editor and the first corpus resulting from the use of this tool in the context of the PorSimples project. The Editor was developed to help building a parallel corpus of original texts and two simplified versions: natural and strong. Although our focus was on building and analyzing a corpus of newspaper texts, the Editor and the Portal of Parallel Corpora of Simplified Texts can be used to build and query, respectively, other parallel corpora of original and simplified texts from different text genres. For different languages, the language-dependent resources have to be provided and integrated (i) a parser, (ii) a list of simple words, and (iii) dictionaries of complex/ambiguous to simpler discourse markers.

The parallel corpus containing 104 pairs of original and simplified versions can be queried and/or downloaded through the Portal of Parallel Corpora of Simplified Texts to be used in studies of text simplification. Another contribution of this work is the XCES annotation standard for parallel corpora of original-simplified texts, which can also be accessed in the Portal. This corpus can serve as training data for statistical or machine learning methods of simplification; indeed, this work is underway in the PorSimples project.

To summarize, besides the Editor, the PorSimples project has produced the following main contributions: (i) the original-simplified parallel corpora, (ii) the XCES annotation standard developed to register the simplification information and (iii) the Portal of Parallel Corpora to store and query the original or simplified texts.

Our efforts consist of the first step towards the development of automatic text simplification systems for poor literacy readers and potentially people with other cognitive disabilities. The ultimate goal is to help changing the alarming scenario in Brazil, where the majority (68%) of the 30.6 million people between 15 and 64 years who have studied up to 4 years only reach the rudimentary level of literacy, and the majority (75%) of people who studied up to 8 years is only literate at the basic level.

As future work, we will use the resulting corpus to help in the development of rule-based and corpus-based simplifications systems, starting from deciding if a sentence should be simplified or not (non-simplification), and when it should be split, since these cases present a large number of examples.

References

1. Ribeiro, V. M.: Analfabetismo e alfabetismo funcional no Brasil. In: Boletim INAF. Instituto Paulo Montenegro, São Paulo (2006)

2. Max, A.: Writing for Language-impaired Readers. In: Proceedings of Seventh International Conference on Intelligent Text Processing and Computational Linguistics, Mexico City, Mexico. Springer-Verlag, Berlin Heidelberg New York (2006) 567-570
3. Petersen, S. E.: Natural Language Processing Tools for Reading Level Assessment and Text Simplification for Bilingual Education. PhD thesis, University of Washington (2007)
4. Siddharthan, A.: Syntactic Simplification and Text Cohesion. PhD thesis, University of Cambridge (2003)
5. Devlin, S., Unthank, G.: Helping aphasic people process online information. In: Proceedings of the ACM SIGACCESS 2006, Conference on Computers and Accessibility, Portland, Oregon, USA (2006) 225-226
6. Klebanov, B., Knight, K., Marcu, D.: Text Simplification for Information-Seeking Applications. In: On the Move to Meaningful Internet Systems. Volume 3290 of LNCS, Springer-Verlag, Berlin Heidelberg New York (2004) 735-747
7. Vickrey, D., Koller, D.: Sentence Simplification for Semantic Role Labeling. In: Proceedings of the ACL-HLT (2008) 344-352
8. Chandrasekar, R., Srinivas, B.: Automatic Induction of Rules for Text Simplification. Knowledge-Based Systems, 10 (1997) 183-190
9. Daelemans, W., Hothker, A., Sang, E. T. K.: Automatic Sentence Simplification for Subtitling in Dutch and English. In: Proceedings of the 4th International Conference on Language Resources and Evaluation, Lisbon, Portugal (2004) 1045-1048
10. Petersen, S. E., Ostendorf, M.: Text Simplification for Language Learners: A Corpus Analysis. In: Proceedings of the Speech and Language Technology for Education Workshop (SLaTE-2007), Pennsylvania, USA (2007) 69-72
11. Muller, C., Strube, M.: Multi-Level Annotation in MMAX. In: Proceedings of the 4th SIGdial Workshop on Discourse and Dialogue, Sapporo, Japan (2003)
12. Ide, N., Romary, L.: International standard for a linguistic annotation framework. Journal of Natural Language Engineering, 10 (3-4) (2004) 211-225
13. Suderman, K., Ide, N.: Layering and Merging Linguistic Annotations. In: Proceedings of EACL Workshop "Multi-dimensional markup in NLP", Trento, Italy (2006) 89-92
14. Megyesi, B. B., Dahlqvist, B.: The Swedish-Turkish Parallel Corpus and Tools for its Creation. In: Proceedings of NoDaLida 2007, Tartu, Estonia (2007)
15. Specia, L., Aluisio, S. M., Pardo, T. A. S.: Manual de Simplificação Sintática para o Português. Technical Report NILC-TR-08-06. São Carlos-SP (2008) (In Portuguese)
16. Aluísio, S., Specia, L., Pardo, T., Maziero, E., Caseli, H. M., Fortes, R. "A Corpus Analysis of Simple Account Texts and the Proposal of Simplification Strategies: First Steps towards Text Simplification Systems " In the proceedings of The 26th ACM Symposium on Design of Communication (SIGDOC 2008), pp. 15-22.
17. Williams S., Reiter E.: Generating Readable Texts for Readers with Low Basic Skills. In: Proceedings of ENLG-2005 (2005) 140-147
18. Biderman, M. T. C.: Dicionário Iustrado de Português. Editora Ática, São Paulo (2005)
19. Janczura, G. A., Castilho, G. M., Rocha, N. O.: Normas de concretude para 909 palavras da língua portuguesa. Psic.: Teor. e Pesq. 23 (2007)195-204
20. Pardo, T. A. S., Nunes, M. G. V.: Review and Evaluation of DiZer - An Automatic Discourse Analyzer for Brazilian Portuguese. In: Proceedings of PROPOR 2006. Volume 3960 of LNCS, Springer-Verlag, Berlin Heidelberg New York (2006) 180-189
21. Bick, E.: The Parsing System "Palavras": Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework. PhD thesis, Aarhus University (2000)

Synchronized Morphological and Syntactic Disambiguation for Arabic

Daoud Daoud

Pricess Sumaya University for Technology
Daoud@batelco.jo

Abstract. In this paper, we present a unique approach to disambiguation Arabic using a synchronized rule-based model. This approach helps in highly accurate analysis of sentences. The analysis produces a semantic net like structure expressed by means of Universal Networking Language (UNL)- a recently proposed interlingua. Extremely varied and complex phenomena of Arabic language have been addressed.

Keywords: Arabic Language, Synchronized Model, Disambiguation, UNL

1 Introduction

Compared to French or English, Arabic as an agglutinative and highly inflected language shows its proper types of difficulties in morphological disambiguation, since a large number of its ambiguities come from both the stemming and the categorization of a morpheme while most of ambiguities in French or English are related to the categorization of a morpheme only.

Phrases and sentences in Arabic have a relatively free word. The same grammatical relations can have different syntactic structures. Thus, morphological information is crucial in providing signs for structural dependencies.

Arabic sentences are characterized by a strong tendency for agreement between its constituents, between verb and noun, noun and objective, in matters of numbers, gender, definitiveness, case, person etc. These properties are expressed by a comprehensive system of affixation.

Arabic uses a diverse system of prefixes, suffixes, and pronouns that are attached to the words, creating compound forms that further complicate text manipulation. Simultaneously, Arabic exhibits a large-scale ambiguity already at the word level, which means that there are multiple ways in which a word can be categorized or broken down to its constituent morphemes. This is further complicated by the fact that most vocalization marks (diacritics) are omitted in Arabic texts.

However, the morphological analysis of a word-form, and in particular its morphological segmentation, cannot be disambiguated without reference to context,

Received 10/11/08
Accepted 15/12/08
Final version 30/01/09

and various morphological features of syntactically related forms provide useful hints for morphological disambiguation.

Specifically, Arabic reveals strong interaction between morphological and syntactic processing, which challenges the validity of NLP models that are based on different phases (layers).

The available Arabic rule-based systems use the pipeline model (where morphology is performed first and syntactic processing follows) for processing and disambiguation. It is obvious that this approach is not adequate for Arabic. One the other hand, one would not expect statistical techniques to perform well on infixing languages like Arabic.

We suggest performing morphological and syntactic processing of Arabic text in a single and joint framework; thereby facilitating the disambiguation process. We will first discuss the sources of ambiguity in Arabic. Then, we discuss methods of disambiguation based on the dependency grammar and the necessity of having a synchronized model. Finally, we present the architecture and implementation of our system.

2 Sources of Ambiguities in Arabic

Ambiguities are mainly caused by the dropping of the short vowels. Thus, a word can have different meanings. In Arabic there are three categories of words: noun, verbs and particles. The dropping of short vowels can cause ambiguities within the same category or across different categories:

For example: the word قبل points out to many concepts (table 1).

Table 1: example of different meanings of a word

before	particle
accept	verb
Kiss	verb
kisses	Noun (broken plural)
to be accepted	Verb
to be kissed	verb

One needs to select the right meaning by looking at the context. Given the highly inflection nature of Arabic, resolving ambiguities is syntactically possible among different categories but harder within the same one.

Other source of ambiguity is caused by the compound forms that can be generated. Arabic uses a diverse system of prefixes, suffixes, and pronouns that are attached to the words, creating compound forms that further complicate text manipulation. Identifying such particles is crucial for analyzing syntactic structures as they reveal structural dependencies such as subordinate clauses, adjuncts, and prepositional phrase attachments. This means that there are multiple ways in which a word can be categorized or broken down to its constituent morphemes.

For instance, the word كوارث can be segmented as presented in table 2:

Table 2: Ambiguity caused by compound forms

catastrophes/tragedies	Noun (broken plural)
like/such as + inheritor	ka/PREP+wAriv/NOUN

On other cases, correct morphological analysis is required to resolve structural ambiguities among Arabic sentence.

For example, consider the first sentence in table 3, the "ين" suffix attached to "ولد" provides information about number (dual) and case ending (accusative). The accusative sign determines the syntactic roles of each constituents of the first sentence although it is in the basic order VSO. In the second sentence, the same suffix disambiguate the syntactic roles despite that the object precedes the subject. In the third sentence, the verb *hit* "ضربا" follows the two boys "الولدان" and there is a number agreement between both of them. Additionally, the two boys "الولدان" takes the nominative sign and *hani* "هانيا" takes the accusative sign suggesting that: *Hani* is the object and the *two boys* are the subject.

Table 3: Examples of structural ambiguities

sentence			Word order	Syntactic roles
الولدين the two boys	هاني Hani	ضرب hit	VSO	*Hani* is the subject *The two* boys are the object
هاني Hani	الولدين the two boys	ضرب hit	VOS	*Hani* is the subject *The two boys* are the object
هانيا Hani	ضربا hit	الولدان the two boys	SVO	*Hani* is the object The two boys are the subject

These examples show how difficult to disambiguate Arabic. The segmentation is driven by the context and the structural dependencies within the sentence. On the other hand, syntactic roles are disambiguated by morphology.

3 Methods of Disambiguation

Many of the ambiguities can be resolved by looking at the context. The linguistic contexture can resolve many of the ambiguities especially among different word classes.

From the development point of view, processing and disambiguation of Arabic depend in the following sources of information:

- The lexicon: provides basic and initial information about lexical items (grammatical attribute).
- Adjacency constraints: specify the compatibility or the incompatibility of two neighboring morphemes. For instance:

  - The Idafa construct[1] cannot be followed by a preposition.
  - A preposition cannot be followed by a preposition.
  - A noun cannot follow a noun unless it is an adjective or the second part of the idafa construct.
- Morphological dependencies [1]: describes the type and direction inflected from one constituent to another. As shown in Figure 1 a verb that follows the subject should agree in number and gender, thus the verb is morphologically dependent on the subject. On the other hand, the subject is morphologically dependent on the verb in case ending.
- Syntactic dependencies [1]: determine binary relations between the lexical items in the sentence. In Figure 1, the verb *hit* is the head of *two boys* (subject) and *hani* (object).

As shown figure 1, it is not necessarily that the syntactic dependent of a head is also morphologically dependent. *Hit* and *the two boys* are exhibiting mutual morphological dependencies.

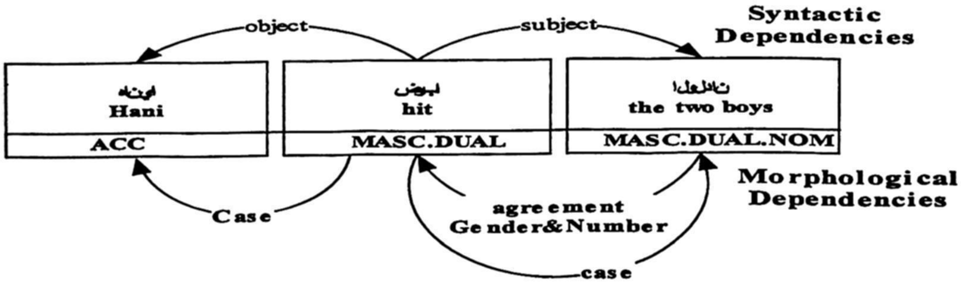


Figure 1: Example of morphological and syntactic dependencies

To demonstrate how the above information can be employed in disambiguation, consider the sentence shown in Figure 2. The ambiguity in the sentence is stemmed from the following two word forms:

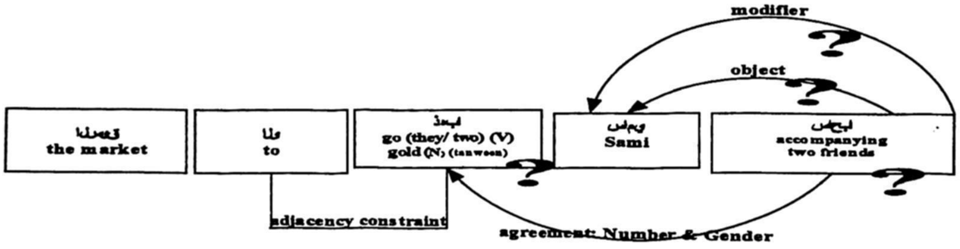


Figure 2: Example of ambiguity resolution

صاحب+ا ➔ (*accompanying or two friends*)

[1] The IDAFA construction is an important grammatical structure in Arabic. It is a genitive construction in which two nouns are linked in such a way that the second (second part of the construction) qualifies or specializes the first (first part of the construction).

ذهب+ا ➔ (*they went*) or *gold* (accusative)

The disambiguation process is started by using the adjacency condition that a noun cannot be followed by a preposition (الى **to**). Thus, ذهبا (*they went*) is a verb (go) [MASC, DUAL} not a noun. (*Sami*) سامي (a named entity) cannot be the subject of the verb as there are no morphological dependencies (agreement in number). On the other hand, a morphological dependencies exists between ذهبا and صاحبا suggesting that it is (*two friends*) and that it is the subject. This solution is verified by the existence of a morphological dependency between صاحبا (*two friends*) and سامي (*Sami*): the suffix that indicates duality ending is ان (NOM), but when the noun is the first part of the IDAFA construction the suffix should be ا which is the case in the above sentence. So, *Sami* is the second part of the IDAFA construction.

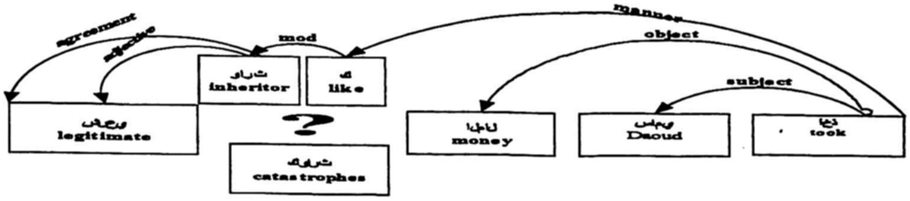


Figure 3: An example of syntactic dependencies disambiguation

In the sentence shown in figure 3, disambiguation is driven by syntactic dependencies. The verb (took) is the head of two dependents which are the subject and the object of (took). This is considered a NUCLEAR PROCESS that contains two participants in association with a 'process' element. Following [2], any additional constituent is either:

- Indirect participant in a process.
- Additional information about a condition or circumstances pertaining to a process.

In Modern Standard Arabic, both indirect participants and circumstances are realized by two basic types of grammatical structure:

- Accusative nominals.
- Prepositional phrases of various kinds.

This is left us with one solution to "كوارث"; it is a prepositional phrase, meaning "like/such as + inheritor". Thus, it should be segmented correctly by recognizing the first character as a preposition (ka) and the rest of the morpheme as the word "وارث inheritor". This solution is verified by the existence of both syntactic and morphological dependencies with the word following it "شرعي legitimate".

4 The Necessity for a Synchronized Model

In light of the above, it is clear that in some cases syntactic dependencies provide cues to perform segmentation and morphological analysis. On the other hand, morphological analysis and adjacency constraints are necessary to disambiguate syntactic structures. Thus, the pipeline model (where morphology is performed first and syntactic processing follows) will not suffice. In this model, a morphological analyzer provides all possible solutions to the syntactic parsing which leads to high magnitude of computational complexity of parsing. To demonstrate this, a word form in the Penn Arabic Treebank (ATB) has, on average, two morphological solutions [3]. The complexity of any parsing algorithm will have a term order of:

$$\prod_{i=1}^{N} a_i$$

where a_i is the number of alternative solutions of the *i*th word [4]. Therefore, the average complexity of parsing a 20 words Arabic sentence using the pipeline model can reach up to 1048576. Thus, linguistic information tend to be more effective at selecting between alternative solutions at the lower levels of the analysis and less effective at doing so at the higher levels [5].

Different systems that process Arabic with some degree of disambiguation are described in the literature [4, 6, 7]. All of them are rule-based systems adapting the pipeline model. Attia [6] tried to reduce ambiguity by putting restriction on the lexical items during the morphological analysis phase. He reported that his system took 141 minutes (CPU time) to parse a test suite of 229 sentences.

The system described in [4] took a more restricted approach by selecting one solution during the morphological phase without having any syntactic information.

On the other hand, statistical techniques have widely been applied to automatic morphological analysis for many languages including English, Turkish and Malay [8]. The main challenge for such systems is that in Arabic, any particular word will appear less often than in English for a given text length and type. Thus, an Arabic datasets will have a higher degree of sparseness than comparable English counterparts [9]. This is significant as it may affect the success of standard statistical techniques on Arabic data. However, Diab, Hacioglu, and Jurafsky [10] reported a remarkable performance for Arabic morphological Analysis using Support Vector Machines (SVMs). They claim above 99% accuracy on tokenization and 95.49 accuracy on POS tagging. Their tools are trained on a sample of 4519 sentence of ATB. For the same size of English dataset, they reported a 94.97 accuracy on POS tagging, a result that contradict the fact that the token to type ratio is smaller for Arabic texts than for comparably sized English texts [8, 9]. Habash and Rambow [3] also reported high accuracy rates in their system for tokenizing and morphologically tagging Arabic words. They used similar approach reported in [10], but by incorporating the Buckwalter morphological analyzer [11] into their system.

However, Larkly, Ballesteros and Conner [8] reported that their simple light stemmer outperformed Diab's morphological analyzer. One of their explanations to this result is: "Arabic text contains so many definite articles that one could obtain the claimed >99% tokenization accuracy simply by removing *AL* from the beginning of words."

Having this in mind, we will take a different approach from previous work. Our system is a rule-based one, which is conceptualized by using dependency grammar, in which linguistic structure is described in terms of dependency relations among the words of a sentence; it does so without resorting to units of analysis smaller or larger than the word. Although dependency grammar has its roots to the work of early

Arabic Grammarians (Kitab al-Usul of Ibn al- Sarraj,d. 928), all of the existing (rule-based) Arabic processing systems are built on phrase structure theory. Processing text using phrase structure framework may suit languages like English, but not a nearly free order language like Arabic [1, 12].

In the next section, we will describe our synchronized model, which is able to perform morphological and syntactic processing of Arabic in as single, integrated and synchronized framework, thus allowing shared information to support disambiguation in multiple levels.

5 The Synchronized Model

Our system is coded using EnCo [13] which we used previously in developing the first Arabic-UNL enconverter. EnCo is a rule-based programming language specialized for the writing of enconverters (translators from a NL into UNL), and provided by the UNL center.

5.1 The UNL

Universal networking language (UNL)[15-18] is a semantic, language independent representation of a sentence that mediates between the enconversion (analysis) and deconversion (generation). It is a computer language aiming at removing language barriers from the Internet. The pivot paradigm is used: the representation of an utterance in the UNL interlingua is a hypergraph where normal nodes bear UWs ("Universal Words", or interlingual acceptions) with semantic attributes, and arcs bear semantic relations [13].

The sentence "Khaled bought a new car" can be expressed in UNL as:

agt*(buy(icl>do(obj>thing),icl>purchase).@past.@entry, Khaled)*
obj*(buy(icl>do(obj>thing),icl>purchase).@past.@entry, car(icl>automobile))*
mod*(car(icl>automobile),new)*

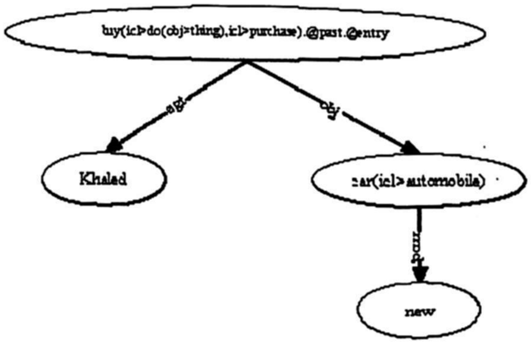


Figure 4: A UNL graph

Figure 4 shows the graph representation of this UNL expression. The node represents the Universal Word (UW). Arcs represent binary relations such as "agt", "obj" and "mod". Attributes are attached to UW to include information about time, aspect, number, modality, etc. In the previous sentence, the attribute "@past" was attached to the event "buy" to indicate that the event happened in the past. The "@entry" attribute is used to indicate the entry point or main node (head) for the whole expression.

5.2 The EnCo Rule-based Programming Language

EnCo [13] is a rule-based programming language specialized for the writing of enconverters[2] . EnCo works in the following way. An input string is scanned from left to right. During the scan, all matched morphemes with the same starting characters are retrieved from the dictionary and become candidate morphemes. The rules are applied to these candidate morphemes, according to the rule priority, in order to build a semantic network for the sentence. The character string not yet scanned is then scanned from the beginning according to the applied rule; the process continues in the same manner. The output of the whole process is a semantic network expressed in the UNL format. If the dictionary retrieval or the rule application fails, it backtracks.

The abstract model underlying EnCo is a computing device consisting of:

- an input tape (node-list), containing at the beginning the input text (in one node) and then the input morpheme or lexemes recognized so far, (each in one node), followed by the remaining text (in one node).
- 2 active heads on that tape (left analysis window (LAW) and right analysis window (RAW))
- a group of "context" heads (condition windows) surrounding the 2 active heads.
- an output "node-net" sharing some nodes with the node-list.

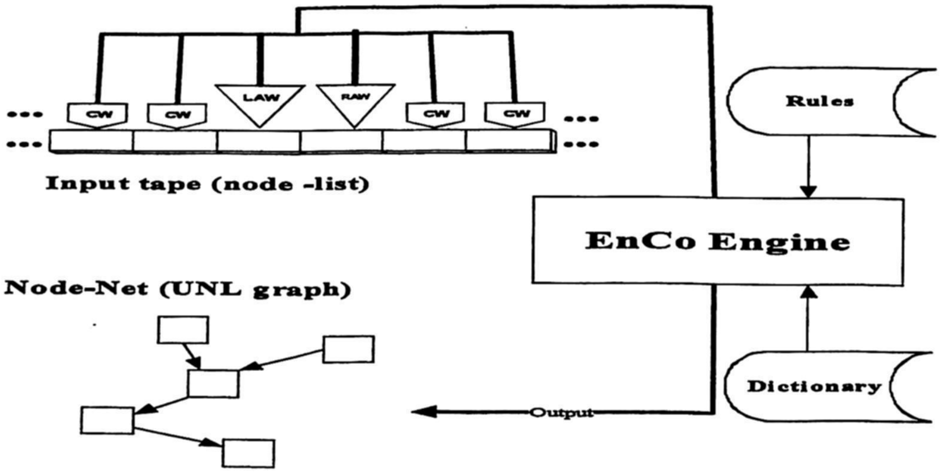


Figure 5: The Computing model of EnCo

[2] We use the term "enconverter", and not "parser", because the process involves a lexical transfer from the "lexical space" of the NL at hand (while many have several "levels" such are morphs, morphemes, word forms, lexemes, lemmas, derivational lexical families, and word senses) to the "lexical space" of UNL (the UWs, and their hierarchy).

The analysis rules have the following syntax (EnCo 1999):

```
<TYPE>...(<PRE2>)(<PRE1>){<LNODE>} {<RNODE>} (<SUF1>) (<SUF2>)... P<PRI>;
Where,
<LNODE>:="{" [<COND1>] ":" [<ACTION1>] ":" [<RELATION1>]":" [<ROLE1>] "}"
<LNODE>:="{" [<COND2>] ":" [<ACTION2>] ":" [<RELATION2>]":" [<ROLE2>] "}"
```

For example, the interpretation of the following rule is:

+{:+BLANK::}{BLK:::}P255;

Type of Operation = "+" which mean combination of right node to left node

Cond1 = nothing

Cond2 = BLK (white space)

Action1 = +BLANK (add the BLANK symbol to the existing list of grammatical attributes or symbols found in the left node)

Action2 = nothing

P255 = Priority 255 (High)

5.3 Overall Analysis Strategy using EnCo

Developing EnCo rules requires a controlling mechanism that specifies which rule should be fired and which rules should not be fired. For that, we use tactical symbols written or removed from the input tape. Without using the KB (knowledge base), the only way to analyze Arabic is to depend on linguistic knowledge and on what exists in the sentence. Without having this controlling mechanism, this task would be impossible.

For example, suppose we have the following sentence:

ساق خالد السيارة الجديدة بسرعة كبيرة

Khalid drove the new car at a high speed.

To analyze this sentence correctly, we should discover the boundaries of the entities that exist in the sentence. Since "Khalid" is not followed by an adjective, it is allowed to be an agent of the verb "drive" and it is removed from the node-list (tape). On the other hand, since "car" is followed by an adjective which has the same gender, it is not allowed for "car" to be an object before handling the adjective first ("car" is a dependent of "drive", and "new" is a dependent of "car": it is not allowed to process the head before its dependents).

5.3.1 EnCo and Dependency Grammars

The formalism provided by EnCo rules embeds the language description despite the fact that this description it is not clear or understandable by humans. This is because this formalism is more oriented to the process of building a practical application more than to describing the language.

EnCo is oriented towards the production of dependency graphs. It analyses a sentence by establishing links between individual words and specifying the type of link in each case. Each link connects a word (the "head") with one of its "dependents" (an argument or modifier). A head can have many dependents, but each dependent can have only one head. Of course, the same word can be the head in one link and a dependent in another.

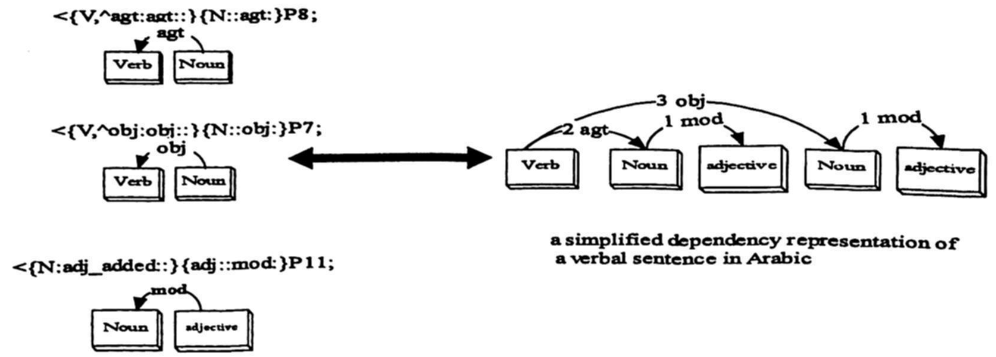

<{V,^agt:agt::}{N::agt:}P8;
agt
Verb
Noun
<{V,^obj:obj::}{N::obj:}P7;
obj
Verb
Noun
<{N:adj_added::}{adj::mod:}P11;
mod
Noun
adjective
3 obj
2 agt
1 mod
1 mod
Verb
Noun
adjective
Noun
adjective
a simplified dependency representation of
a verbal sentence in Arabic

Figure 6: The bidirectional mapping of EnCo rules and DG

Figure 6 shows also that the dependency representation of a sentence (arrows point from each word to its dependents: modifiers or arguments) is inferred from the EnCo rules.

Looking carefully at each rule, we find that it establishes a linking between two words, one is dependent on the other. Some links are shown clearly in the UNL-graph; others are implicit and are used within rules only. Each rule also indicates head-dependent order which is very important in specifying the word order typology. Dependent-Dependent order (the mutual order of two dependent of the same head) is specified by the priority strategy or by using symbols.

In the above example, the "agt" rule has a higher priority than "obj" rule, reflecting the fact that the subject of a verb is before its object.

In EnCo, this dependent-dependent order can be implemented alternatively by using symbols. As an example, consider the following two rules:

<{V,^agt:agt::}{N::agt:}P8;

<{V,^objt,agt:obj::}{N::obj:}P9;

The second rule executes after the first one independently of their priorities. This is because the "agt" symbol is added after the first rule and is a condition of the second rule. This shows how dependent-dependent relation can be implemented.

As we have seen, there is no intermediate representation between the text and the output graph. EnCo takes the input text and transforms it into the corresponding UNL graph directly. It is the responsibility of the rules to ensure the right sequence of execution as we have shown previously.

EnCo provides two mechanisms to ensure the right execution of the rules: rule prioritization and use of tactical symbols. The developers have to use them correctly as EnCo does not provide any other means to assist or to enforce this mechanism.

5.3.2 Disambiguation Mechanism

At any particular moment in time, EnCo is in a describable configuration. Between this moment and the next discrete time stamp, the machine reads its input from the tape, refers to rules controlling its behavior, and considering both the input and the current configuration, determines what behavior to exhibit (i.e. erase/write on tape, move left, move right, create a an arc in the UNL graph, etc.), which determine the next configuration.

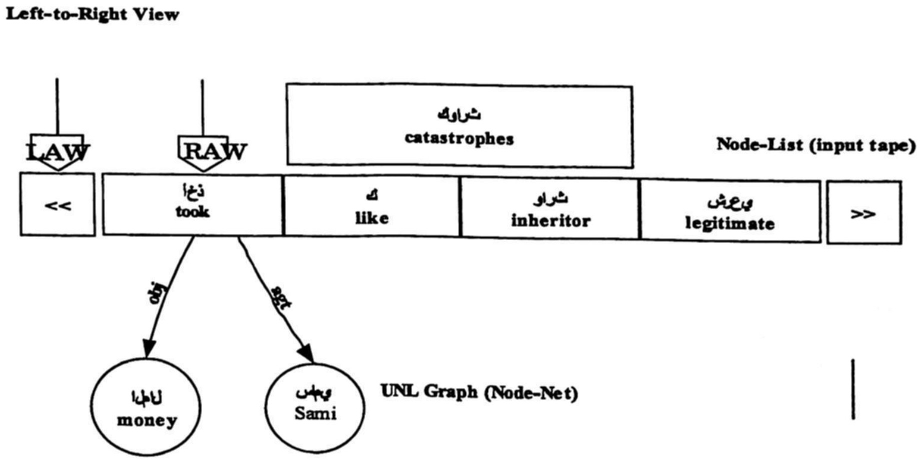


Figure 7: A describable configuration of EnCo

All information needed for disambiguation (adjacency, morphological dependencies, syntactic dependencies, in addition to basic lexical attributes retrieved from the dictionary) is accessible at any moment of processing. This information is expressed by the symbols attached to each node in the input tape. Figure 7, demonstrates the availability of syntactic dependencies needed to disambiguate "كوارث". The engagement of the verb *took* in "agt" and "obj" relationships, provides information to the enconverter to perform the correct segmentation and word selection. More to the point, the enconverter will backtrack if it had done wrong selection. For example, consider the following rule:

```
?R{V1,obj,agt:::}{NDE:::}P255;
```

This rule will force the enconverter to backtrack when it reaches the following configuration: the left node is a verb engaged into two syntactic relations (agt and obj) and the right node is an entity or a noun. The UNL expression of (Sami took the money as a legitimate(valid) inheritor) is shown below:

```
;======================= UNL =======================
;أخذ سامي المال كوارث شرعي;
[S]
agt(take(icl>event):00.@entry.@past,    Sami:04)
aoj:01(valid:0L,          inheritor:0G)
mod:01(like:0F.@entry,          inheritor:0G)
obj(take(icl>event):00.@entry.@past,    money:0B.@def)
man(take(icl>event):00.@entry.@past,   :01)
[/S]
;==================================================
;;Time 0.1     Sec
;;Done!
```

To implement this enconverter with disambiguation capabilities, 1500 rules were coded with the following functional classes:

- Backtracking rules. They are given the highest priority to prevent further execution when a wrong situation or assumption is recognized..
- Morphological analysis rules. They are important because when they are executed they provide information about morphological dependencies (by using symbols) that might be useful in executing other rules. For example, an accusative noun cannot be a subject. Morphological analysis is mainly done by combination type rules (+ or -).
- Information collectors rules. They determine structural dependencies and boundaries within the sentence by gathering information from the surface structure.
- Syntactic dependencies rules. They are responsible for producing the UNL graph by performing reduction and creating an arc in the UNL graph.
- The lowest priority is assigned to the "shift right" rules.

Longer sentences have been analyzed accurately with this system (.3 CPU time):

هزم الفريق السعودي هولندا على استاد فلسطين في مباراته الاخيرة في يوم الاحد وتمكن الفريق السعودي
من تحقيق النصر بثلاث اهداف جميلة بعد ان لعبو بطريقة جماعية وبذلك يصل الفريق السعودي الى النهائيات
محققا احلام الجمهور السعودي

The Saudi team defeated Holland on Palestine Stadium in its last match in Sunday and the Saudi team was able to achieve victory by three wonderful goals as a result of their collective play, so the Saudi team reaches the finals achieving the dreams of the Saudi audience.

6 Conclusion

During the development period of the Arabic enconverter, the number of lexical items added to UNL-Arabic dictionary reached 120,000 entries. This covers the UWs provided by UNL center and the most frequent Arabic lexicon. More sophisticated features are added to each entry to cover morphological, syntactic and semantics aspects. In designing those features, we took into consideration the analysis and generation processes. Functional words are also added to the dictionary along with all prefixes and suffixes needed for Arabic morphology.

Our system managed to handle the following situation and sentences:

- Agreement and Morphological generation
- All type of relations and attributes
- Embedded and relative sentences
- Nominal and verbal sentences

The synchronized computational model of EnCo along with conceptualization using Dependency Grammar provides us with the right mean to disambiguate a language such as Arabic. This approach outperform pipeline model in terms of computational time and accuracy. Our system disambiguate efficiently words that exhibit ambiguities across different categories (noun-verb ambiguity, particle- verb ambiguity), but less efficient in words that fall within same category (noun-noun, verb-verb). This is expected, as morphological and syntactic dependencies become less decisive in disambiguation in those situations. Our future work will focus in this issue.

References

1. I. Mel'tchuk, *Dependency Syntax: Theory and Practice*: State University of New York Press, 1988.
2. S. C. Dik, *The Theory of Functional Grammar*: Foris, 1989.
3. N. Habash and O. Rambow, "Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop," in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Ann Arbor, Michigan: Association for Computational Linguistics, 2005.
4. R. Allan and M. Hanady, "Towards including prosody in a text-to-speech system for modern standard Arabic," *Comput. Speech Lang.*, vol. 22, pp. 84-103, 2008.
5. M. C. Macdonald, N. J. Pearlmutter, and M. S. Seidenberg, "The lexical nature of syntactic ambiguity resolution," *Psychological view*, vol. 101, pp. 467-703, 1994.
6. M. A. Attia, "Handling Arabic Morphological and Syntactic Ambiguity within the LFG Framework with a View to Machine Translation " in *School of Languages, Linguistics and Cultures*, vol. Doctor of Philosophy: University of Manchester, 2008.
7. E. Othman, K. Shaalan, and A. Rafea, "Towards Resolving Ambiguity in Understanding Arabic Sentence," presented at the International Conference on Arabic Language Resources and Tools, NEMLAR, Egypt, 2004.
8. L. S. Larkey, L. Ballesteros, and M. E. Connell, "Light Stemming for Arabic Information Retrieval " in *Arabic Computational Morphology*, A. Soudi, A. v. d. Bosch, and G. Neumann, Eds.: Springer Netherlands, 2007.
9. A. Goweder and A. De Roeck, "Assessment of a significant Arabic corpus," presented at Arabic NLP Workshop at ACL/EACL 2001, Toulouse, France, 2001.
10. M. Diab, K. Hacioglu, and D. Jurafsky., "Automatic Tagging of Arabic Text: From raw text to Base Phrase Chunks," presented at HLT-NAACL, 2004.
11. T. Buckwalter, "Buckwalter Arabic Morphological Analyzer Version 1.0," *Linguistic Data Consortium (LDC)*, 2002.
12. M. A. Covington, "A dependency parser for variable–word–order languages," in *Computer assisted modeling on the IBM 3090: Papers from the 1989 IBM Supercomputing Competition*, vol. 2, K. R. Billingsley, H. U. Brown III, and E. Derohanes, Eds. Athens, Greece: Baldwin Press, 1992, pp. 799–845.
13. H. Uchida, "Enconverter Specifications," UNU/IAS UNL Center, 1999.
14. H. Uchida, "Deconverter Specifications," UNU/IAS UNL Center, 1999.
15. H. Uchida and M. Zhu, "The Universal Networking Language Beyond Machine Translation," 2001.
16. H. Uchida and M. Zhu, "The Universal Networking Language specification, version 3.0," 2003, Ed.: UNDL Foundation, 2003.
17. I. Boguslavskij, "UNL from the linguistic point of view," presented at MMA'01, 2001.
18. C. Boitet, "Gradable quality translations through mutualization of human translation and revision, and UNL-based MT and coedition," in *Universal Networking Language, advances in theory and applications*, vol. 12, *Research in Computing Science*, J. Carde˚osa, A. Gelbukh, and E. Tovar, Eds. Mexico, 2005, pp. 393—410.

Parsing with Polymorphic Categorial Grammars

Matteo Capelletti[1] and Fabio Tamburini[2]

[1] Lix, École Polytechnique - France
matteo.capelletti@elisanet.fi
[2] DSLO, University of Bologna - Italy
fabio.tamburini@unibo.it

Abstract. In this paper we investigate the use of polymorphic categorial grammars as a model for parsing natural language. We will show that, despite the undecidability of the general model, a subclass of polymorphic categorial grammars, which we call *linear*, is mildly context-sensitive and we propose a polynomial parsing algorithm for these grammars.

1 Introduction

The simplest model of a categorial grammar is the so called Ajdukiewicz–Bar-Hillel calculus of [2] and [4]. Syntactic categories are formed from a given set of atoms as *functions* a/b and $b\backslash a$, with b and a categories. The intuitive meaning of a syntactic category of the form a/b (resp. $b\backslash a$) is that it looks for an *argument* of category b to its right (resp. left) to give a category of type a. The resulting grammar system is known to be context-free.

Contemporary categorial grammars in the style of Ajdukiewicz–Bar-Hillel grammars are called *combinatory categorial grammars*, see [25]. Such systems adopt other forms of composition rules which enable them to generate non–context-free languages, see [29; 28]. The other main tradition of categorial grammar, the type-logical grammars of [20; 18], stemming from the work of [15], adopt special kinds of structural rules, that enable the system to generate non–context-free languages.

Both approaches increase the generative power of the basic system by adding special kinds of rules. In this paper, we adopt a different strategy which consists in keeping the elementary rule component of Ajdukiewicz–Bar-Hillel grammar and in introducing *polymorphic* categories, that is syntactic categories that contain variables ranging over categories. The inference process will be driven by unification, rather than by simple identity of formulas. We will see two kinds of polymorphic categorial grammars, one that is Turing complete and another, resulting from a restriction on the first, which is mildly context-sensitive. This second system, which is obviously the most interesting one for linguistics, has some important advantages with respect to the aforementioned ones. In respect to TLG, the polymorphic system we define is *polynomial*, as we will prove by providing a parsing algorithm. In respect to CCG (and most known TLG), our system is not affected by the so called *spurious ambiguity* problem, that is the problem of generating multiple, semantically equivalent, derivations.

© A. Gelbukh (Ed.)
Advances in Computational Linguistics.
Research in Computing Science 41, 2009, pp. 87-98

Received 07/11/08
Accepted 15/12/08
Final version 26/01/09

A system somewhat similar to our polymorphic categorial grammar was studied in the late 80s as a kind of feature theoretic categorial syntax by [26] and [30], although, to our knowledge, after these works it has been almost completely neglected. While our system, in its most general form, allows the encoding of feature structures into categorial grammar, our primary concern will be to discuss its generative power and its computational properties. We will present general polymorphic categorial grammars, and show that they are undecidable. Then we will define a restricted class of such grammars which is mildly context-sensitive. We will show this by giving examples of the languages that can and cannot be generated, and by providing a polynomial parser for such grammars.

In relation to other mildly context-sensitive grammar formalism, the categorial grammar framework is well known for its close correspondence between syntactic derivation and the construction of semantic representation, see [27]. For this reason the extensions of the basic calculi with more powerful devices allows for the construction of sophisticated parsing systems dealing in parallel with both the syntax and the semantics of languages.

2 Polymorphic Ajdukiewicz–Bar-Hillel Grammar

In this paper we will assume that categories are not only of the form a/b or $b\backslash a$, but also $a \otimes b$, this is not standard in the categorial literature, but such product categories are useful as we will see in the examples that follow. From now on, we use capital letters A, B and C and their variants as metavariables over formulas. As we said before, an expression of category A/B looks for an expression of category B to its right to give a compound expression of category A and dually for $B\backslash A$. Instead, an expression of category $A \otimes B$ is an expression whose syntactic information is given by a component of category A and another of category B. We assume that the slashes bind more tightly than the product. We define a *sequent* as a pair where the first element, the antecedent, is a list of category, while the second, the succedent, contains a single category, and we write it as $\Gamma \to A$.

The deductive system given in Figure 1, which we call $\mathrm{AB}^{\otimes}$, is a simple modification of the calculus of [14] to which it can easily be proved equivalent.

Identity Axioms:	$A \to A$	
Product Axioms:	$A,\ B \to A \otimes B$	
Shifting Rules:	$\dfrac{\Gamma \to C/A}{\Gamma, A \to C}\ (S_1)$	$\dfrac{\Gamma \to A\backslash C}{A, \Gamma \to C}\ (S_2)$
Cut Rules:	$\dfrac{\Gamma, A \to C \quad \Delta \to A}{\Gamma, \Delta \to C}\ (C_1)$	$\dfrac{\Gamma \to A \quad A, \Delta \to C}{\Gamma, \Delta \to C}\ (C_2)$

Fig. 1. Ajdukiewicz–Bar-Hillel calculus with product, $\mathrm{AB}^{\otimes}$.

There are two ways in literature for extending the basic models of categorial grammars to generate non context-free languages. The first is the use of structural rules or other types of composition schemes. These approaches are characteristic of type-logical grammars, see [20; 18; 19], and combinatory categorial grammars (CCG), see [25; 3], and have been widely explored in the past. The second is based on the introduction of *polymorphism*. In this paper, we study this second approach.

The formalism of polymorphic categorial grammar that we are going to present is inspired by the polymorphic theory of types, see [17; 9]. Types may contain type variables together with constants, and these variables may be (implicitly or explicitly) quantified over. The idea of polymorphism is very simple and natural. Rather than defining a class of id functions $\mathtt{id}_{Int} :: Int \rightarrow Int$, $\mathtt{id}_{Char} :: Char \rightarrow Char$ and so forth, the function id is defined for any type α, as $\mathtt{id} :: \forall\alpha.\alpha \rightarrow \alpha$ or $\mathtt{id} :: \alpha \rightarrow \alpha$ where α is implicitly universally quantified.

The same idea is very natural also in linguistics, where, for example, coordination particles such as 'and' and 'or' are typically polymorphic, as they coordinate expressions of *almost* any syntactic category. Thus one can find in the categorial grammar literature several examples of polymorphic assignments for these expressions [15; 24; 8; 7].

Another example of Ajdukiewicz–Bar-Hillel style categorial grammars adopting a form of polymorphism are the unification categorial grammars of [26; 30; 10], where polymorphism is used at the level of feature structures.

In this section, we are going to explore Ajdukiewicz–Bar-Hillel categorial grammars *with product* ($AB^{\otimes}$, for short) extended with different forms of polymorphism. In the first place, we extend the notion of category as to include variables and define a sort of categorial grammar with ML style polymorphism, see [17; 12]. Variables are assumed to be implicitly universally quantified, with quantifiers in prenex position.

Consider the formula $(\alpha\backslash\alpha)/\alpha$ which can be assigned to the word 'and' in a lexicon. Applied to 'John' $:: n$, it will give the expression 'and John' $:: n\backslash n$, while applied to 'walks' $:: n\backslash s$, it will give the expression 'and walks' $:: (n\backslash s)\backslash(n\backslash s)$, and so forth. Thus the right argument α in $(\alpha\backslash\alpha)/\alpha$ gets instantiated in the application process and the result of such instantiation, a *substitution*, is applied to the value $\alpha\backslash\alpha$. Hence the process of type inference for this kind of polymorphic categorial grammars requires nothing more than type unification and substitution. We call the resulting system Unification Ajdukiewicz–Bar-Hillel grammars with product, UAB$^{\otimes}$ for short.

2.1 Unification Ajdukiewicz–Bar-Hillel Grammars

Syntactic categories of UAB$^{\otimes}$ are defined as follows.

$$\begin{array}{lll}
\textbf{Atoms:} & \mathcal{A} & ::= \; a, b, c, n, s, i \ldots \\
\textbf{Variables:} & \mathcal{V} & ::= \; \alpha, \beta, \gamma \ldots \\
\textbf{Categories:} & \mathcal{F} & ::= \; \mathcal{A} \mid \mathcal{V} \mid \mathcal{F} \otimes \mathcal{F} \mid \mathcal{F}\backslash\mathcal{F} \mid \mathcal{F}/\mathcal{F}
\end{array}$$

Unification of categories is defined in (1). We use $\sharp$ as a variable over $\{/, \backslash, \otimes\}$ connectives. The *substitution* of a formula A for a variable α in a formula B, denoted $B[\alpha := A]$, is defined as follows:

$$\begin{array}{ll} \alpha[\alpha := C] = C & \alpha[\beta := C] = \alpha \text{ if } \alpha \neq \beta \\ A[\alpha := C] = A \text{ for } A \in \mathcal{A} & (A\sharp B)[\alpha := C] = (A[\alpha := C]\sharp B[\alpha := C]). \end{array}$$

We write $f \cdot g$ the composition of f and g, defined as $\lambda x.f\ (g\ x)$. Let $V(B)$ be the set of variables occurring in B, we define the *unification* of A and B, denoted $A \approx B$, by the following recursion, which is taken with minor modifications from [5].

$$\begin{array}{llll} \alpha \approx B & = & [\alpha := B] & \text{if } \alpha \notin V(B) \\ & = & \mathtt{Id} & \text{if } \alpha \equiv B \\ & = & \mathtt{fail} & \text{otherwise} \\ A\sharp B \approx A'\sharp B' & = & (\sigma A \approx \sigma A') \cdot \sigma & \text{where } \sigma = B \approx B' \\ A \approx \alpha & = & \alpha \approx A & \end{array} \tag{1}$$

The unification Ajdukiewicz–Bar-Hillel calculus, UAB$^{\otimes}$ is defined in Figure 2.[3]

Identity Axioms:	$A \to A$	
Product Axioms:	$A, B \to A \otimes B$	
Shifting Rules:	$\dfrac{\Gamma \to C/A}{\Gamma, A \to C}\ (S_1)$	$\dfrac{\Gamma \to A\backslash C}{A, \Gamma \to C}\ (S_2)$
Cut Rules:	$\dfrac{\Gamma, A \to C \quad \Delta \to B}{\Gamma, \Delta \to C(A \approx B)}\ (C_1')$	$\dfrac{\Gamma \to B \quad A, \Delta \to C}{\Gamma, \Delta \to C(A \approx B)}\ (C_2')$

Fig. 2. Unification Ajdukiewicz–Bar-Hillel calculus, UAB$^{\otimes}$

We give here some examples of non context-free languages generated by UAB$^{\otimes}$ grammars.

Example 1 *We define the UAB$^{\otimes}$ grammar for the language $a^n b^n c^n$, $n \geqslant 1$, a well-known non-context-free language. Let grammar G_1 consist of the following assignments:*

$$\begin{array}{ll} a :: s/(b \otimes c) & b :: b \\ a :: (s/\alpha)\backslash(s/(b \otimes (\alpha \otimes c))) & c :: c \end{array}$$

We derive the string 'aabbcc'. We write A for the formula $(s/\alpha)\backslash(s/(b \otimes (\alpha \otimes c)))$. For readability, boxes are drawn around the words that anchor the axioms to the lexicon.

[3] Obviously, the rules involving unification are only defined if unification is defined.

$$\dfrac{\dfrac{\overset{\boxed{a}}{s/(b\otimes c)\to s/(b\otimes c)} \quad \dfrac{\overset{\boxed{a}}{A\to A}}{s/\alpha, A\to s/(b\otimes(\alpha\otimes c))}}{s/(b\otimes c), A\to s/(b\otimes((b\otimes c)\otimes c))} \quad \dfrac{\overset{\boxed{b}}{b\to b} \quad \dfrac{\dfrac{\overset{\boxed{b}}{b\to b}\quad \overset{\boxed{c}}{c\to c}}{b,c\to b\otimes c}\quad \overset{\boxed{c}}{c\to c}}{b,c,c\to (b\otimes c)\otimes c}}{b,b,c,c\to b\otimes((b\otimes c)\otimes c)}}{s/(b\otimes c), A, b, b, c, c\to s}$$

To show that G_1 indeed generates the language $a^n b^n c^n$, $n \geqslant 1$, we proceed by induction on n. If $n = 1$, then ‘abc’ is generated by axioms $a/(b\otimes c) \to a/(b\otimes c)$, $b \to b$ and $c \to c$. Assume that G_1 generates $a^n b^n c^n$. Then G_1 assigns a^n the category s/A for some A and $b^n c^n$ the category A. We have $a :: (s/\alpha)\backslash(s/(b\otimes(\alpha\otimes c)))$. Hence, G_1 assigns a^{n+1} the category $s/(b\otimes(A\otimes c))$ and $b^{n+1}c^{n+1}$ the category $b\otimes(A\otimes c)$. We conclude that G_1 generates $a^{n+1}b^{n+1}c^{n+1}$.

Example 2 *We define a UAB$^{\otimes}$ grammar for ‘ww’, $w \in \{a, b\}^+$. Let grammar G_2 consist of the following assignments:*

$$\begin{array}{ll} a :: a & b :: b \\ a :: s/a & b :: s/b \\ a :: (s/\alpha)\backslash(s/(\alpha\otimes a)) \qquad & b :: (s/\alpha)\backslash(s/(\alpha\otimes b)) \end{array}$$

It is easy to see that grammar G_2 generates exactly the language ‘ww’ with $w \in \{a, b\}^+$. As in the case of G_1, type variables are used as accumulators for long-distance dependencies. Here we give an example deduction for the string ‘aabaab‘, using A as for $(s/\alpha)\backslash(s/(\alpha\otimes a))$ and B for $(s/\alpha)\backslash(s/(\alpha\otimes b))$.

$$\dfrac{\dfrac{\dfrac{\overset{\boxed{a}}{s/a}\quad \dfrac{\overset{\boxed{a}}{A\to A}}{s/\alpha, A\to s/(\alpha\otimes a)}}{s/a, A\to s/(a\otimes a)}\quad \dfrac{\overset{\boxed{b}}{B\to B}}{s/\alpha, B\to s/(\alpha\otimes b)}}{s/a, A, B\to s/((a\otimes a)\otimes b)}\quad \dfrac{\dfrac{\overset{\boxed{a}}{a\to a}\quad \overset{\boxed{a}}{a\to a}}{a, a\to a\otimes a}\quad \overset{\boxed{b}}{b\to b}}{a, a, b\to (a\otimes a)\otimes b}}{s/a, A, B, a, a, b\to s}$$

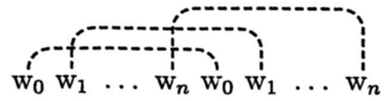
A typical example of non context-freeness of natural language are the so called cross serial dependencies, which can be found, for instance, in Dutch subordinate clauses.

Example 3 *We define a UAB$^{\otimes}$ grammar for Dutch cross-serial dependencies. An example is the following subordinate clause, from [25]:*

Ik	*Cecilia*	*Henk*	*de*	*nijlpaarden*	*zag*	*helpen*	*voeren.*
I	Cecilia	Henk	the	hippopotamuses	saw	help	feed.

I saw Cecilia help Henk feed the hippopotamuses.

These constructs exhibit dependencies of the form ‘ww’, where the ith words in the two halves are matched.



A sample lexicon generating the sentence in this example is the following:

Ik, Cecilia, Henk, de nijlpaarden	$:: n$
zag	$:: ((n \otimes (n \otimes \alpha))\backslash c)/(\alpha\backslash i)$
helpen	$:: ((n \otimes \alpha)\backslash i)/(\alpha\backslash i)$
voeren	$:: n\backslash i$

With such a lexicon we obtain the following deduction for the subordinate clause (we write Z for the type of 'zag', H for that of 'helpen' and N for the string 'Ik Cecilia Henk de nijlpaarden').

$$\dfrac{\begin{array}{c}\boxed{N}\\ n \otimes (n \otimes (n \otimes n))\end{array} \qquad \dfrac{\dfrac{\dfrac{\boxed{zag}}{Z \to Z}}{Z, \alpha\backslash i \to (n \otimes (n \otimes \alpha))\backslash c} \qquad \dfrac{\dfrac{\dfrac{\boxed{helpen}}{H \to H}}{H, \alpha\backslash i \to (n \otimes \alpha)\backslash i} \quad \dfrac{\boxed{voeren}}{n\backslash i \to n\backslash i}}{H, n\backslash i \to (n \otimes n)\backslash i}}{Z, (H, n\backslash i) \to (n \otimes (n \otimes (n \otimes n)))\backslash c}}{n \otimes (n \otimes (n \otimes n)), (Z, (H, n\backslash i)) \to c}$$

These examples show that the languages generated by UAB$^\otimes$ grammars properly include the context-free languages (since AB$^\otimes$ grammars are instances of UAB$^\otimes$ grammars).

With regard to the generative power of UAB$^\otimes$ grammars, it can be proven that if we allow null assignments, that is assignments of the form $\epsilon :: A$, where ϵ is the empty string, the UAB$^\otimes$ formalism becomes undecidable. We can show this by translating in our categorial setting the argument of [13] to prove the Turing completeness of unification based attribute-value grammars. It is possible to encode a Turing machine M in a UAB$^\otimes$ grammar $G(M)$ such that $G(M)$ generates the string 'halt' if and only if M halts with a blank input tape; this is enough to conclude the undecidability of UAB$^\otimes$ grammars. Polymorphic null assignments, in fact, correspond quite neatly to lexical rules, as proposed in [6], leading to an undecidable formalism.

2.2 Constraining UAB⊗ Grammars

A constrain that we can impose on UAB$^\otimes$ grammars to avoid undecidability is *linearity*. Roughly, we impose the restriction that any lexical type may contain at most one variable, occurring once in an argument position and once in value position. Thus, α/α, $(s/\alpha)\backslash(s/(\alpha \otimes a))$ are licit types, while $(\alpha\backslash\alpha)/\alpha$, $(s/(\alpha \otimes \beta))\backslash(s/((\alpha\otimes\beta)\otimes a))$ and $(s/(\alpha\otimes\alpha))\backslash(s/((\alpha\otimes\alpha)\otimes a))$ are not. More precisely we define *linear* categories as the types F_2 generated by the following context-free grammar.

$$\begin{array}{rcl} \sharp & ::= & \otimes \mid / \mid \backslash \\ F_0 & ::= & A \mid F_0 \sharp F_0 \\ F_2 & ::= & F_1 \sharp' F_1 \mid F_0 \mid F_2 \sharp' F_0 \mid F_0 \sharp' F_2 \end{array} \qquad \begin{array}{rcl} \sharp' & ::= & / \mid \backslash \\ F_1 & ::= & F_1 \sharp F_0 \mid F_0 \sharp F_1 \mid \alpha \end{array} \tag{2}$$

This definition of categories may deserve some comments. The interesting cases are the F_2 formulas of the form A/B or $B\backslash A$, with A and B in F_1 (the other cases are meant essentially to put these in context). Consider the case of A/B, then α occurs exactly once in A and in B, since a F_1 category contains the variable α by construction. By analogy with lambda terms, we can think of the occurrence of α in B as a *binder* (possibly a pattern-binder), and of the occurrence in A as the *bound* variable.

An UAB$^\otimes$ grammar is linear if all its lexical assignments are linear. Furthermore, in linear UAB$^\otimes$ grammar, we work by simple *variable instantiation*, rather than by a full-fledged unification algorithm. More precisely let us denote A^B a formula A with a distinguished occurrence of a subformula B. A^C is the formula obtained from A^B by replacing the occurrence of the subformula B with the formula C. The linear UAB$^\otimes$ calculus results from the UAB$^\otimes$ calculus in Figure 2 by replacing the Cut rules with the following *instantiation* rules.

$$\frac{\Delta \rightarrow A^B \quad \Gamma, A^\alpha \rightarrow C}{\Gamma, \Delta \rightarrow C[\alpha := B]} \qquad \frac{\Gamma \rightarrow A^B \quad A^\alpha, \Delta \rightarrow C}{\Gamma, \Delta \rightarrow C[\alpha := B]} \tag{3}$$

Observe that given a linear UAB$^\otimes$ grammar adopting the rules in 3, only linear types can occur in any of its deductions.

Observe also that the UAB$^\otimes$ grammars for $\mathrm{a}^n\mathrm{b}^n\mathrm{c}^n$ and ww languages as well as that for the Dutch cross serial patterns, are all linear. On the other hand, no linear UAB$^\otimes$ grammar can be given for the so called MIX or Bach language that is the language of the strings containing an equal number of a's, b's and c's[4].

As we have the proper inclusion of context-free languages and the realization of limited cross-serial dependencies, in order to have a *mildly context-sensitive* grammar formalism we shall prove that linear UAB$^\otimes$ grammars can be parsed in polynomial time. We do this in the next section by providing a parsing algorithm for linear UAB$^\otimes$ grammars.

3 Polynomial Parsing with Linear UAB⊗ Grammars

In this section we define a polynomial parsing algorithm for linear UAB$^\otimes$ grammars. It is based on a standard agenda-driven chart-parsing method and makes use of an external table, which we call *instantiation table*, for storing the 'partial' instantiations of variables Let n be the length of the input string and *Lex*

[4] To see this, we observe that the context-free language of the strings containing an equal number of a's and b's is not *linear*, in the sense of [11], see [16]. Hence for the MIX language, a UAB$^\otimes$ grammar needs to bind two distinct variables for each symbol, what violates linearity.

the input lexicon. Cells of the instantiation table are denoted $\mathcal{I}_{(i,k,j)}$, where $0 \leqslant i < j \leqslant n$ and $0 \leqslant k \leqslant |Lex|$. We extend the construction of formulas with two kinds of variables, α_k and $\alpha_{(i,k,j)}$ where i, k and j are as before. The difference between the two kinds of variables is that α_k is an uninstantiated variable while $\alpha_{(i,k,j)}$ is a variable α_k which has been instantiated when an item spanning between i and j was generated. The algorithm assumes that different lexical entries contain different variables, that is for no k the variable α_k occurs in two distinct lexical assignments[5].

The parser operates on two kinds of items which we represent as

$$(i, \Delta \rhd \Gamma \to A, j) \quad \text{and} \quad (i, \Gamma \lhd \Delta \to A, j)$$

where i and j are integers and $\Delta \rhd \Gamma \to A$ and $\Gamma \lhd \Delta \to A$ are *sequents* in which exactly one occurrence of an auxiliary symbols (either $\rhd$ or $\lhd$) appear. These symbols play a role similar to the dot in Earley parsing systems.

An item of the form $(i, \Delta \rhd \Gamma \to A, j)$ asserts that $\Delta\Gamma \to A$ is derivable in linear UAB$^{\otimes}$ and that $w_{i+1} \dots w_j \Rightarrow^* \Delta$. Furthermore, the items have a predictive component.

- In case $A \equiv A' \otimes A''$ and $\Delta\Gamma \equiv A' A''$, it asserts that for some context Λ, $w_{l+1} \dots w_i A \Lambda \Rightarrow^* C$ with $0 \leqslant l < i$. This means that the item has been *predicted* from another item $(l, \Xi \rhd A \Lambda \to C, i)$.
- In case $A \equiv \alpha_{(i,k,j)}$ and $\Delta\Gamma \equiv B$ for some formula $B \in \mathcal{I}_{(i,k,j)}$, it asserts that for some context Λ, $w_{l+1} \dots w_i A^{\alpha_{(i,k,j)}} \Lambda \Rightarrow^* C$ with $0 \leqslant l < i$. This means that the item has been *dereferenced* from another item $(l, \Xi \rhd A^{\alpha_{(i,k,j)}} \Lambda \to C, i)$.

The dual conditions hold for $(i, \Gamma \lhd \Delta \to A, j)$. For simplicity, we write items of the form $(i, \lhd \Delta \to A, j)$ and $(i, \Delta \rhd \to A, j)$ as $(i, \Delta \to A, j)$.

The proposed parsing algorithm for linear UAB$^{\otimes}$ grammars is showed in Figure 3.

The correctness of the algorithm can be proven by adapting the methods of [1; 23] for the CYK and Earley parsers and by observing that the triple (i, k, j) resulting from the instantiation of a variable is determined by the unique name k of the variable α_k and by the span (i, j) over which the instantiation has been determined. Observe that the Completion rules apply to a non-instantiated variable and therefore that instantiated variables behave like constants in the parsing process, whose value is determined by the Dereference rules.

To show that the resulting algorithm is polynomial, we follow the usual argument for agenda-driven chart-based parser evaluation, see for instance [22; 21]. The number of sequents that can occur in a cell of the chart for a given grammar with lexicon Lex is $O(n^2|Lex||\Sigma|)$ where Σ is the set of subformulas of the lexicon[6] and $|Lex|$ is the number of word-category pairs contained in the lexi-

[5] Clearly all these modifications do not affect linearity, and are motivated by correctness and efficiency reasons.

[6] That is for each lexical entry w :: A, Σ contains all the subformulas of A. For example the formula $(a\backslash b)/c$ generates the subformulas $\{(a\backslash b)/c, a\backslash b, a, b, c\}$

Input: a string $w = w_1 \dots w_n$ and an $UAB^{\otimes}$ grammar G.
Output: Accept/reject.
Data Structures: an $n+1 \times n+1$ matrix $\mathcal{T}$, the *chart*, whose cells are sets of sequents, a set of items $\mathcal{N}$, the *agenda*, containing all the items to be processed and the instantiation table $\mathcal{I}$ as defined before.

```
Initialization:
Let $\mathcal{N} = \varnothing$ and $\mathcal{T}_{(i,j)} = \varnothing \ \ \forall i, j$.
For $i = 1$ to $n$ do
    $\mathcal{N} = \mathcal{N} \cup \{ (i-1, A \to A, i) \mid w_i :: A \in Lex\}$
Main cycle:
While $\mathcal{N} \neq \varnothing$ do
  remove one item $\nu = (i, \Gamma \to A, j)$ from $\mathcal{N}$.
  If $\Gamma \to A \notin \mathcal{T}_{(i,j)}$, then
    add $\Gamma \to A$ to chart $\mathcal{T}_{(i,j)}$
    Shifting:
    If $\nu = (i, \Gamma \to C/A, j)$, then add $(i, \Gamma \triangleright A \to C, j)$ to $\mathcal{N}$.
    If $\nu = (i, \Gamma \to A\backslash C, j)$, then add $(i, A \triangleleft \Gamma \to C, j)$ to $\mathcal{N}$.
    Prediction:
    If $\nu = (i, \Gamma \triangleright A \otimes B\Delta \to C, j)$, then add $(j, \triangleright A, B \to A \otimes B, j)$ to $\mathcal{N}$.
    If $\nu = (i, \Gamma A \otimes B \triangleleft \Delta \to C, j)$, then add $(i, A, B\triangleleft \to A \otimes B, i)$ to $\mathcal{N}$.
    $\epsilon$-Scanning:
    If $\nu = (i, \Gamma \triangleright A\Delta \to C, j)$ and $\epsilon \Rightarrow^{+} A$, then add $(i, \Gamma A \triangleright \Delta \to C, j)$ to $\mathcal{N}$.
    If $\nu = (i, \Gamma A \triangleleft \Delta \to C, j)$ and $\epsilon \Rightarrow^{+} A$, then add $(i, \Gamma \triangleleft A\Delta \to C, j)$ to $\mathcal{N}$.
    Completion:
    If $\nu = (i, \Gamma \triangleright A\Delta \to C, j)$, then for all $\Lambda \to A \in \mathcal{T}_{(j,k)}$, add $(i, \Gamma A \triangleright \Delta \to C, k)$ to $\mathcal{N}$.
    If $\nu = (i, \Gamma A \triangleleft \Delta \to C, j)$, then for all $\Lambda \to A \in \mathcal{T}_{(k,i)}$, add $(k, \Gamma \triangleleft A\Delta \to C, j)$ to $\mathcal{N}$.
    If $\nu = (i, \Lambda \to A, j)$, then $\begin{cases} \text{for all } \Gamma \triangleright A\Delta \to C \in \mathcal{T}_{(k,i)}, \text{ add } (k, \Gamma A \triangleright \Delta \to C, j) \text{ to } \mathcal{N}, \\ \text{for all } \Gamma A \triangleleft \Delta \to C \in \mathcal{T}_{(j,k)}, \text{ add } (i, \Gamma \triangleleft A\Delta \to C, k) \text{ to } \mathcal{N}. \end{cases}$
    If $\nu = (i, \Gamma \triangleright A^{\alpha_l} \Delta \to C, j)$, then
      for all $\Lambda \to A^{B} \in \mathcal{T}_{(j,k)}$
        add $(i, \Gamma A^{\alpha_l} \triangleright \Delta \to C[\alpha_l := \alpha_{(i,l,k)}], k)$ to $\mathcal{N}$ and update $\mathcal{I}_{(i,l,k)} = \mathcal{I}_{(i,l,k)} \cup \{B\}$.
    If $\nu = (i, \Gamma A^{\alpha_l} \triangleleft \Delta \to C, j)$, then
      for all $\Lambda \to A^{B} \in \mathcal{T}_{(k,i)}$
        add $(k, \Gamma \triangleleft A^{\alpha_l}\Delta \to C[\alpha_l := \alpha_{(k,l,j)}], j)$ to $\mathcal{N}$ and update $\mathcal{I}_{(k,l,j)} = \mathcal{I}_{(k,l,j)} \cup \{B\}$.
    If $\nu = (i, \Lambda \to A^{B}, j)$, then
      for all $\Gamma \triangleright A^{\alpha_l}\Delta \to C \in \mathcal{T}_{(k,i)}$
        add $(k, \Gamma A^{\alpha_l} \triangleright \Delta \to C[\alpha_l := \alpha_{(k,l,j)}], j)$ to $\mathcal{N}$ and update $\mathcal{I}_{(k,l,j)} = \mathcal{I}_{(k,l,j)} \cup \{B\}$.
      for all $\Gamma A^{\alpha_l} \triangleleft \Delta \to C \in \mathcal{T}_{(j,k)}$
        add $(i, \Gamma \triangleleft A^{\alpha_l}\Delta \to C[\alpha_l := \alpha_{(i,l,k)}], k)$ to $\mathcal{N}$ and update $\mathcal{I}_{(i,l,k)} = \mathcal{I}_{(i,l,k)} \cup \{B\}$.
    Dereference:
    If $\nu = (i, \Gamma \triangleright \alpha_{(k,l,m)} \Delta \to C, j)$ and $A \in \mathcal{I}_{(k,l,m)}$ then add $(j, \triangleright A \to \alpha_{(k,l,m)}, j)$ to $\mathcal{N}$.
    If $\nu = (i, \Gamma \alpha_{(k,l,m)} \triangleleft \Delta \to C, j)$ and $A \in \mathcal{I}_{(k,l,m)}$ then add $(j, A\triangleleft \to \alpha_{(k,l,m)}, j)$ to $\mathcal{N}$.
Termination: If, when $\mathcal{N} = \varnothing$, $\Gamma \to s \in \mathcal{T}_{(0,n)}$, for some $\Gamma$, then accept, else reject.
```

Fig. 3. Recognition algorithm for linear UAB$^{\otimes}$grammars.

con. Then, the time is dominated by the Completion rules, that gives a global asymptotic complexity of $O(n^5|Lex||\Sigma|)$.

4 Conclusion

In this paper we have investigated some linguistic and computational properties of unification based categorial grammars. We have seen that, like other unification based grammar formalisms, unrestricted UAB$^{\otimes}$ grammars are Turing complete. However, we have also seen that the constraint of *linearity* defines a linguistically interesting class of categorial grammars. Most notably, it locates the system among the mildly context-sensitive formalisms. Another pleasant aspect of the resulting system, at least with respect to any other CG-based mildly context-sensitive categorial formalism (be it CCG or type-logical grammar) is the absence of spurious ambiguity.

The work in [8] presents a decision procedure for a kind of polymorphic categorial grammar allowing at most two instances of the same variable in argument position and one in value (e.g. $(X\backslash X)/X$). However, with such kind of polymorphism we can generate languages that are beyond the mildly context-sensitives (for instance, we can easily generate indexed languages such as $\{www|w \in \{a, b\}^+\}$). This extension, and its implications on recognition complexity, are currently being investigated.

Despite the fact that in this paper we have been concerned only with a recognition algorithms, a proper *parsing* algorithm that provides all the parses of the input, handling also semantic information, can be easily provided. In that case, the correspondence between syntax and semantics typical of categorial grammars and the absence of spurious ambiguity of our systems become important ingredients of the resulting natural language parser.

We wish to conclude this section by observing that the linearity constraint can also be relaxed. For instance, while preserving the condition that only one variable occurs in a formula, we can allow more than two occurrences of this variable. Then bounded languages such as w^i or $\mathrm{a}_1^i\mathrm{a}_2^i \ldots \mathrm{a}_n^i$, which are generalizations of 'ww' and $\mathrm{a}^i\mathrm{b}^i\mathrm{c}^i$, can easily be generated and recognized in polynomial time.

References

1. A. Aho and J. Ullman. *The Theory of Parsing, Translation and Compiling*, volume 1: Parsing. Prentice-Hall, INC., 1972.
2. K. Ajdukiewicz. Die syntaktische Konnexität. *Studia Philosophica*, 1:1–27, 1935.
3. J. Baldridge. *Lexically Specified Derivational Control in Combinatory Categorial Grammar*. PhD thesis, University of Edinburgh, 2002.
4. Y. Bar-Hillel. A quasi-arithmetical notation for syntactic description. *Language*, 29:47–58, 1953.
5. H. Barendregt. Lambda calculus with types. In S. Abramsky, Dov M. Gabbay, and T. S. E. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 2, pages 117–309. Oxford University Press, 1992.
6. B. Carpenter. The generative power of categorial grammars and head-driven phrase structure grammars with lexical rules. *Computational Linguistics*, 17(3):301–313, 1991.
7. S. Clark and J. R. Curran. Wide-coverage efficient statistical parsing with ccg and log-linear models. *Computational Linguistic*, 33(4):493–552, 2007.
8. M. Emms. Parsing with polymorphism. In *EACL*, pages 120–129, 1993.
9. J.-Y. Girard, P. Taylor, and Y. Lafont. *Proofs and Types*. Cambridge University Press, 1989.
10. D. Heylen. *Types and Sorts. Resource logic for feature checking*. PhD thesis, UiL-OTS, Utrecht, 1999.
11. J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, 1979.
12. G. Huet. A uniform approach to type theory. In *Logical foundations of functional programming*, pages 337–397. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1990.
13. M. Johnson. *Attribute-Value Logic and the Theory of Grammar*, volume 16 of *CSLI Lecture Notes*. CSLI, Stanford, California, 1988.
14. M. Kandulski. The equivalence of nonassociative Lambek categorial grammars and context-free grammars. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 34:41–52, 1988.
15. J. Lambek. The mathematic of sentence structure. *American Mathematical Monthly*, 65(3):154–170, 1958.
16. P. Linz. *An introduction to formal languages and automata*. D. C. Heath and Company, Lexington, MA, USA, 1990.
17. R. Milner. A theory of type polymorphism in programming. *Journal of Computer and System Sciences*, 17:348–375, 1978.
18. M. Moortgat. Categorial type logics. In J. van Benthem and A. ter Meulen, editors, *Handbook of Logic and Language*, pages 93–177. Elsevier, Amsterdam, 1997.
19. R. Moot. *Proof Nets for Linguistic Analysis*. PhD thesis, UiL-OTS, Utrecht, 2002.

20. G. Morrill. *Type Logical Grammar: Categorial Logic of Signs.* Kluwer, Dordrecht, 1994.
21. M.-J. Nederhof and G. Satta. Tabular parsing. In C. Martin-Vide, V. Mitrana, and G. Paun, editors, *Formal Languages and Applications, Studies in Fuzziness and Soft Computing 148*, pages 529–549. Springer, 2004.
22. G. Satta and O. Stock. Bidirectional context-free grammar parsing for natural language processing. *AIJ: Artificial Intelligence*, 69, 1994.
23. K. Sikkel. Parsing schemata and correctness of parsing algorithms. *Theoretical Computer Science*, 199, 1998.
24. M. Steedman. Dependency and coordination in the grammar of dutch and english. *Language*, 61(3):523–568, 1985.
25. M. Steedman. *The Syntactic Process.* The MIT Press, 2000.
26. Hans Uszkoreit. Categorial unification grammars. In *COLING*, pages 187–194, 1986.
27. J. van Benthem. *Essays in Logical Semantics.* Reidel, Dordrecht, 1986.
28. K. Vijay-Shanker and D. J. Weir. The equivalence of four extensions of context-free grammars. *Mathematical Systems Theory*, 27, 1994.
29. D. J. Weir and A. K. Joshi. Combinatory categorial grammars: Generative power and relationship to linear context-free rewriting systems. In *Meeting of the Association for Computational Linguistics*, pages 278–285, 1988.
30. H. Zeevat. Combining categorial grammar and unification. In U. Reyle and C. Rohrer, editors, *Natural Language Parsing and Linguistic Theories*, pages 202–229. D. Reidel, Dordrecht, 1988.

A CCG-based System for Valence Shifting for Sentiment Analysis

František Simančík and Mark Lee

School of Computer Science, University of Birmingham, Birmingham, UK
frantisek.simancik@worc.ox.ac.uk, m.g.lee@cs.bham.ac.uk

Abstract. The automatic classification of sentiment in text is becoming an important area of research. In this work, we present a linguistic system for sentence-level valence annotation. Our system uses the formalism of Combinatory Categorial Grammar to represent words as functions acting on their syntactic arguments, which provides a unified way of implementing various classes of valence shifters. We propose two simple semi-automatic methods for estimating the valence of individual terms based on the lexical relations of WordNet. We evaluate the system on the data generated for the Affective Text task of SemEval 2007 and show that it compares favourably with the systems participating in the task.

Keywords: sentiment analysis, valence annotation, valence shifters, headlines, combinatory categorial grammar.

1 Introduction

The number of opinion-rich resources such as discussions, blogs and review sites has been growing rapidly in recent years. As a result of this, there is a demand for tools capable of classifying texts not only by the topic but also the attitude and opinion they convey; giving rise to new areas in Natural Language Processing called Opinion Mining and Sentiment Analysis.

One of the most prominent tasks in the field is the classification of valence (positive/negative orientation). Researchers (Pang et al. [7], Kennedy and Inkpen [6] and others) have successfully applied supervised machine learning methods[1] to determine the valence of longer texts. These approaches rely on the availability of a large amount of human-tagged training data and, compared to linguistic methods, reveal very little about the nature of the connection between a text and the opinion it expresses.

[1] Naive Bayes Classifiers, Support Vector Machines, etc.

© *A. Gelbukh (Ed.)*
Advances in Computational Linguistics.
Research in Computing Science 41, 2009, pp. 99-108

Received 10/11/08
Accepted 15/12/08
Final version 04/02/09

The Affective Text task (Strapparava and Mihalcea [11]), conducted as a part of SemEval 2007, focuses on unsupervised sentence-level classification of emotions and valence in newspaper headlines. The reason behind this, the authors said, was to emphasise the study of emotion lexical semantics, and avoid biasing participants toward simple "text categorization" approaches ([11]). Indeed, the average length of a headline in the SemEval data is only seven words, which is too short to be susceptible to statistical analysis without adequate training data.

The task consisted of two independent parts: emotion labelling (using a fixed set of predefined labels) and ternary valence annotation. We present a system for the second subtask. The reasons for choosing the setting of SemEval were twofold - it sets a well-defined problem and gives us direct means of comparing our results with other existing systems.

1.1 The SemEval Data Sets and Evaluation

The data sets gathered for the Affective Text task were formed of newspaper headlines, which are believed to have a high load of emotional content and are therefore suitable for sentence-level sentiment analysis ([11]). The headlines were collected from major online newspaper portals such as New York Times, CNN and BBC News.

The participants were presented with a smaller development data set consisting of 250 headlines, while the final submissions were evaluated on a larger test set with 1000 headlines. The valence of each headline was labelled independently by six human annotators in the interval $[-100, 100]$. For the coarse-grained evaluation it was subsequently mapped to three classes: negative $[-100, -50]$, neutral $(-50, 50)$ and positive $[50, 100]$.

1.2 Brief Outline of Our Method

Similarly to other existing systems for this task (Andreevskaia and Bergler [1], Chaumartin [2]), we use a pre-built dictionary of sentiment-bearing unigrams, which provides a mapping from terms to their valence. To construct the dictionary, we manually compiled a list of seed words with strong valence and then extended it through WordNet's lexical links.

Using the valence dictionary alone is, however, not enough. Consider, for example, the following sentence from the development data set: *"Nigeria hostage feared dead is freed."* which has a positive meaning even though it contains three negative (*hostage, fear, dead*) and only one positive word (*free*).

In order to improve the performance, we enhanced the simple bag-of-words approach by employing sentence-level valence shifters: words which influence the sentiment expressed by other words in the sentence (Polanyi and Zaenen [8]). In the example above, it is the role of the phrase *"is freed"* to shift the valence of its subject *"Nigeria hostage feared dead"* from negative to positive.

We believe that most words (and verbs in particular) may exhibit valence-shifting behaviour. In our model, each term potentially affects the valence of all its syntactic arguments (subjects/objects). This effect is always in the form of multiplication by an appropriate factor, which may be different for different arguments. Thus, for example, a transitive word (e.g. *reject*) may flip the valence of its object while preserving its subject unchanged.

We extend this analysis one step further. The effect of a term on its arguments is not only applied to their valence but also to the effects they have themselves. Thus, for example, *not* flips the effect of *very* in *"not very"* from intensifying to diminishing. Similarly, under this model, the negating effect of *reject* on its object will be inverted in *"don't reject"*, producing a phrase which is neutral to both its subject and object.

We applied Combinatory Categorial Grammar (Steedman [10]) to determine the structural dependencies between individual terms in a sentence. The main reason for this decision was the fact that the syntax of CCG gives rise to a semantic interpretation whose structure (e.g. treating adjectives as functions from nouns to nouns) maps easily to the functionality of valence shifters. We used Clark and Curran's CCG parser ([3]) which seems to work reasonably well even on fragmented sentences.

2 Resources

In this section we will introduce the resources, theories and formalisms which form the basis of our system.

- WordNet 3.0
- Contextual Valence Shifters
- Combinatory Categorial Grammar

2.1 WordNet 3.0

WordNet ([4]), one of the best-known NLP resources, is a lexical database of English developed and maintained at Princeton University. It organises nouns, verbs, adjectives and adverbs into groups of synonyms (synsets), with each synset representing a distinct meaning (word sense). The synsets are interconnected by various semantic links, of which the following are the most relevant to our purpose: *hyponym* (links to a more specific concept), *hypernym* (links to a more general concept), *similar to* and *see also*. The last two are, however, only present amongst adjectives.

2.2 Contextual Valence Shifters

The presence of certain words and phrases in a sentence can modify (intensify, diminish or even flip) the valence expressed by other terms. For instance, in the sentence *"He is not bright."* the valence of *bright* is shifted by *not* from positive to negative. In the following subsections we describe the valence shifters we used in our system. Polanyi and Zaenen ([8]) investigate this phenomenon to a great depth; we implement only a fraction of their suggestions.

Negatives, Intensifiers and Diminishers

Negatives (*not, none, never...*), intensifiers (*very, rather...*) and diminishers (*slightly, a bit...*) are the most obvious valence shifters. In our model, the effect of such a term is to multiply the valence of its argument, which can be a single word or a longer constituent, by a predefined factor (-1 in case of negatives) and also modify its effect appropriately (e.g. negating an intensifier results into a diminisher, intensifying a diminisher produces a stronger diminisher, affecting a neutral term leaves it unchanged).

It has to be recognised, however, that this is an oversimplification. There are occasions on which the above approach is insufficient, often when two or more of these terms compose. For example, consider the phrase *"not very good"*, whose meaning depends strongly on the context and may range from negative to slightly positive. Under our model it always evaluates to the same as *"quite good"*.

Connectors

Certain conjunctions (*but, while, although, however...*) are often used to set up a deliberate contrast in the discussion by firstly introducing a new piece of information and contradicting it immediately. In such cases, it is only the main clause of the sentence which expresses the attitude of the speaker, the effect of the first clause is neutralised by the connector. For example:

> *The plot sounds promising but the audience is likely to leave unimpressed.*

Verbs

Even though not directly mentioned in [8], we believe that verbs have the strongest impact on the overall sentiment of a sentence. For very short sentences, as in the case of headlines, these are often the only valence shifters present at all and their role must not be overlooked. Consider these examples:

> *EU criticises the war in Georgia.*
> *Threat against airlines has been eliminated.*

Although both the sentences are composed of negative and neutral words only, the verbs *criticise* and *eliminate* flip the valence of their objects from negative to positive, and the resulting messages are thus positive. There are many other verbs with this functionality: *attack, stop, forbid, prevent, dislike, reject...*

Other less radical verbs may act on their objects by weakening or intensifying their valence (*emphasise, support, increase...*).

2.3 Combinatory Categorial Grammar

Combinatory Categorial Grammar (Steedman [10]) is a grammatical theory based on categorial calculus and combinatory logic. It provides a completely straightforward mapping from the syntactic properties of its terms to their semantic functionalities, yet it is still efficiently parseable.

In a categorial grammar, all constituents (including individual words) are assigned specific categories describing their syntactic behaviour. Combinatory rules[2] (functional application, composition, etc.) then specify how phrases can combine into larger constituents according to their categories.

The class of syntactic categories can be defined recursively as the set including the atomic categories N (noun), NP (noun phrase), PP (prepositional phrase), S (sentence), and others, and complex categories (compound of the atomic categories) of the form X/Y and $X\backslash Y$, where X and Y are categories.

Using Steedman's notation, complex categories X/Y and $X\backslash Y$ are functors taking argument of category Y and returning a result of category X. The type of the slash specifies the directionality of the argument: / indicates that the argument appears to the right of the functor, whereas \ means that the argument comes to the left.

For example, the category of an English adjective may be written as N/N, indicating that it is a function from nouns (which it takes to its right) to nouns. Similarly, a typical transitive verb has the category $(S\backslash NP)/NP$, making the verb a curried function of two noun phrases (its object and subject) producing a sentence.

To demonstrate the correspondence between syntax and semantics, consider the parse of the following sentence:

$$\begin{array}{ccccc}
\text{John} & \text{has} & \text{very} & \text{little} & \text{money}\\
\cline{1-1}\cline{2-2}\cline{3-3}\cline{4-4}\cline{5-5}
N & (S\backslash NP)/NP & (N/N)/(N/N) & (N/N) & N\\
\cline{1-1}\cline{3-4}
NP & & \multicolumn{2}{c}{N/N} & \\
\cline{3-5}
 & & \multicolumn{3}{c}{N}\\
\cline{3-5}
 & & \multicolumn{3}{c}{NP}\\
\cline{2-5}
 & \multicolumn{4}{c}{S\backslash NP}\\
\cline{1-5}
\multicolumn{5}{c}{\mathrm{S}}
\end{array}$$

Giving rise (by means of functional application) to the following semantic structure:

$$\begin{array}{ccccc}
\text{John} & \text{has} & \text{very} & \text{little} & \text{money}\\
\cline{1-1}\cline{2-2}\cline{3-3}\cline{4-4}\cline{5-5}
John & \lambda x.\lambda y.has(x)(y) & \lambda f.\lambda x.very(f)(x) & \lambda x.little(x) & money\\
\cline{3-4}
 & & \multicolumn{2}{c}{\lambda x.very(little)(x)} & \\
\cline{3-5}
 & & \multicolumn{3}{c}{very(little)(money)}\\
\cline{2-5}
 & \multicolumn{4}{c}{\lambda y.has(very(little)(money))(y)}\\
\cline{1-5}
\multicolumn{5}{c}{has(very(little)(money))(John)}
\end{array}$$

[2] See Steedman [10] or Hockenmaier [5] for full treatment of combinatory rules.

3 The System

Our system consists of four basic components: a dictionary of sentiment-bearing and valence-shifting words, a simple preprocessor, the C&C CCG parser (Clark and Curran [3]), and a classifier, which links the other components together.

3.1 Construction of the Dictionary

Starting with a small set of manually selected seed words, we follow the links in WordNet to derive a larger collection of words with similar meaning. Because the structure of WordNet (in terms of which links are present) differs significantly across the word classes, we propose two different methods for this task: one treats adjectives and adverbs, the other one nouns and verbs.

It is necessary to address the problem of homonymy at this point. The context provided by newspaper headlines is too short to perform sense disambiguation, so we decided to ignore this issue completely and to each word form we simply assign the valence of its most common synset (based on WordNet's relative frequency counts).

The dictionary also contains a short list of common negatives, intensifiers, diminishers and conjunctions (as mentioned in 2.2).

Adjectives and Adverbs

We compiled small sets of positively (*good, beautiful, happy, pleasant, clean, friendly, healthy, correct, lucky, alive, clever*) and negatively (*bad, hideous, sad, unpleasant, dirty, hostile, sick, wrong, unfortunate, dead, stupid*) oriented adjectives, capturing a variety of distinct concepts from the class of sentiment-bearing terms.

Taking one of the seed words at a time, we perform a breadth-first search starting from its three most frequent synsets and proceeding along the *similar to* and *see also* links. These restrict the search space to the class of adjective synsets and, in our experience, reliably preserve valence while still provide satisfactory expansion. To avoid exploring the vast space of irrelevant terms, we terminate the search at depth 10 and treat all the unexplored synsets as being at depth 11.

To calculate the valence of an adjective synset, we sum its distances from the negative seeds and subtract its distances from the positive seeds, employing the intuition that positive synsets are closer to the positive seeds than to the negative ones. Finally, we scale the valence to the interval $[-100, 100]$.

We applied the same method to derive a list of valence-shifting adjectives, using seed sets with increasing (*extreme, large, huge, enormous, immense*) and decreasing (*mild, small, minute, micro, slight*) semantics. This time we restricted our search to the *similar to* links only, which preserve the meaning more accurately than the *see also* ones. The resulting value was mapped exponentially to the range $[\frac{1}{2}, 2]$.

This way we obtained about 4200 sentiment-bearing and 100 quantity-affecting adjectives.

Our treatment of adverbs is morphological. Those derived from adjectives (by appending a morpheme such as *-y, -ly, -ily*) are given the same properties as the corresponding adjectives. This is also applied to common noun-generating morphemes (*-ity, -ness*).

Nouns and Verbs

There are no similarity links amongst nouns and verbs in WordNet, so the above approach (and its adaptations) cannot be used. Instead, we turn our attention to hyponymy, which is often the only semantic link present at all. We compiled a list of general concepts whose valence and effect (as described in 2.2) were estimated manually. All their hyponyms were then assigned the same values as the original concepts.

Our selection was based on the trial data set, but a lot remained open to our intuition only. We tried to include concepts which are likely to appear in a newspaper setting, such as *catastrophe, misfortune, disrespect, immorality, pain, fear, mistreat* and *celebrate, pleasure, protect, cure, wonder, success.*

The entire list contains a hundred concepts. These were inflated through hyponymy into about 5700 synsets giving us 3900 different word forms.

3.2 Parsing and Preprocessing

The C&C parser expects all tokens (even quotation and punctuation marks) to be separated by spaces; we use a simple preprocessor to achieve this. While inserting an extra space is enough in most cases, certain grammatical constructions require special care. For example, we detect negated auxiliaries and expand them to their long forms (e.g. *don't* to *do not*).

We then present the processed text to the parser, which annotates each token with its grammatical category and produces a structure of combinatory rules to be used at each level. Conveniently, the parser also labels words with their morphological base forms, which simplifies their look-up in the dictionary.

3.3 Classification

The final component combines information from the dictionary and the parser. Firstly, all the words are converted into functions of zero or more parameters according to their syntactic categories. The atomic categories are mapped to a basic type, whose instances are completely described by their valence alone. The complex categories give rise to functional types and we treat them as functions modifying the valence and effect of their arguments. Their action is represented by their own valence and by one scaling factor for each of their parameters, which describes their effect (intensification, diminution, negation or just neutral propagation) on that particular argument. Scaling a function by a constant results into multiplying its valence by that constant and modifying its own effects appropriately (e.g. negating an intensifier yields a diminisher). All these numbers are looked up in the dictionary and neutral values are used (0 for valence and 1 for scaling factor) if no matching is found.

A collection of handwritten rules (one for each type) defines how a function processes its arguments. Most rules fall under this scheme: a function takes several arguments of the same type X, scales each of them by the corresponding factor, sums their valence and combines them into one object, and adds its own valence to the result, which is again of type X.

The above scheme applies to, among others, the following type classes:

X/X , the category of adjectives ($X = N$), certain adverbs ($X = N/N$) and common negatives.

(X\X)/X , the category of conjunctions.

(S\NP)[/NP... /NP] , the category of common verbs. The innermost NP refers to the subject, the others to the objects of the verb.

Once all the functions are defined, they are combined according to the combinatory rules suggested by the parser. In an ideal case, the final category of a headline would be S, giving us directly a result of the basic type. Quite often, however, a headline is only a fragment of a sentence and its category is complex. In this case, we return the valence of the resulting function, which corresponds to evaluating it on neutral arguments.

4 Results

Table 1 shows the results of our system on the test data. The full system achieves accuracy of 63.20% and F-score (Rijsbergen [9]) of 51.81 and compares favourably with the systems participating in the SemeEval task[3], where the best results were 55.10% for accuracy and 42.43 for F-measure and, as shown in Table 2, even these were obtained by two different systems.

Table 1 also shows how the performance changes when we restrict the dictionary to certain word classes. It transpires that the effect of adjectives and adverbs is only marginal and the system draws its strength from its treatment of nouns and verbs. We attribute this to the fact that newspaper headlines are often too short to contain any sentiment-bearing adjectives, in which case their valence has to be determined from nouns and verbs only.

5 Conclusions

The results of the Affective Text task indicate that valence annotation is not easy. Our system performs relatively well (for a ternary classifier) in both

[3] See [11] for the full table of results.

Table 1: System results on the test data.

Dictionary in use	Accuracy	Precision	Recall	F_1
full	63.20	53.21	50.48	51.81
adjectives and adverbs only	58.80	43.48	2.44	4.62
nouns and verbs only	62.30	52.00	50.73	51.36

Table 2: The best systems (achieving highest accuracy and F-measure) participating in the SemEval task.

System	Accuracy	Precision	Recall	F_1
CLaC	55.10	61.42	9.20	16.00
CLaC-NB	31.20	31.18	66.38	42.43

precision and recall and improves upon the results obtained by the participating programs.

We adopted the formalism of Combinatory Categorial Grammar to represent words as functions acting on their arguments, which provides a unified and transparent way of implementing some common classes of valence shifters. Our work also emphasises the role of nouns and verbs in short sentence sentiment tagging. We argue that if WordNet is to be used to estimate their valence, the absence of the similarity-like links forces us to abandon the methods commonly used for adjectives. Instead, we proposed a crude semi-automatic approach based on hyponymy.

Acknowledgments. This project was carried out under the Summer Research Scholarship scheme at the School of Computer Science, the University of Birmingham. František Simančík was supported by a grant from the SPP Foundation.

References

1. Andreevskaia, A & Bergler, S 2007, 'CLaC and CLaC-NB: Knowledge-based and corpus-based approaches to sentiment tagging', *Proceedings of SemEval-2007: 4th International Workshop on Semantic Evaluations at ACL 2007*, Prague.
2. Chaumartin, FR 2007, 'UPAR7: A knowledge-based system for headline sentiment tagging', *Proceedings of SemEval-2007: 4th International Workshop on Semantic Evaluations at ACL 2007*, Prague.
3. Clark, S & Curran, JR 2007, 'Wide-Coverage Efficient Statistical Parsing with CCG and Log-Linear Models', *Computational Linguistics*, vol. 33, no. 4, pp. 493-552.

4. Fellbaum, C 1998 *WordNet, An Electronic Lexical Database*, MIT Press, Cambridge, MA (1998).
5. Hockenmaier, J 2003, 'Data and Models for Statistical Parsing with Combinatory Categorial Grammar', PhD thesis, University of Edinburgh.
6. Kennedy, A & Inkpen, D 2005. 'Sentiment Classification of Movie and Product Reviews Using Contextual Valence Shifters', *Proceedings of FINEXIN-05, Workshop on the Analysis of Informal and Formal Information Exchange during Negotiations.*
7. Pang, B, Lee, L & Vaithyanathan, S 2002, 'Thumbs up? Sentiment Classification using Machine Learning Techniques', *Proceedings of EMNLP-02, the Conference on Empirical Methods in Natural Language Processing*, Philadelphia, US, pp. 79-86.
8. Polanyi, L & Zaenen, A 2005 'Contextual valence shifters', *Computing Attitude and Affect in Text: Theory and Applications*, pp. 1-10.
9. Rijsbergen, CJV 1979, *Information Retrieval*, Butterworth-Heinemann.
10. Steedman, M 2000, *The Syntactic Process*, Spech and Communication Series, MIT Press Language.
11. Strapparava, C & Mihalcea, R 2007, 'SemEval-2007 Taks 14: Affective Text', *Proceedings of SemEval-2007: 4th International Workshop on Semantic Evaluations at ACL 2007*, Prague.

Classification of "Inheritance" Relations: a Semi-automatic Approach

Ekaterina Lapshinova-Koltunski

IMS, Universität Stuttgart
Azenbergstr.12
70174 Stuttgart
katerina@ims.uni-stuttgart.de

Abstract. This study describes a semi-automatic approach to the classification of "inheritance" relations between morphologically related predicates.

Predicates, such as verbs and nouns subcategorizing for a subclause, are automatically extracted from text corpora and are classified accroding to their subcategorisation properties. For this purpose, we elaborate a semi-automatic knowledge-rich extraction and classification architecture. Our aim is also to compare subcategorisation properties of morphologically related predicates, i.e. verbs and deverbal nouns.

In this work, we concentrate exclusively on the predicates with sentential complements, such as *dass, ob* and *w*-clauses (that, if and wh-clauses) in German, although our methods can be applied for other complement types as well.

1 Introduction

This paper describes a semi-automatic approach to the analysis of subcategorisation properties of morphologically related predicates, such as verbs and nouns. We classify predicates according to their subcategorisation properties by means of extracting them from German corpora along with their complements. In this work, we concentrate exclusively on sentential complements, such as *dass, ob* and *w*-clauses, although our methods can be also applied for other types of complements.

It is usually assumpted that subcategorisation properties of nominalisations are taken over from their underlying verbs. However, our preliminary tests show that there exist different types of relations between them. Thus, our aim is to review the properties of morphologically related words and to analyse the phenomenon of "inheritance" of subcategorisation properties.

For this purpose, we elaborate a set of semi-automatic procedures, with the help of which we not only classify extracted units according to their subcategorisation properties, but also compare the properties of verbs and their nominalisations. Our aim is to serve NLP, especially such large symbolic grammar for deep processing as HPSG or LFG, which need detailed subcategorisation data for their lexicons and grammars.

© A. Gelbukh (Ed.)
Advances in Computational Linguistics.
Research in Computing Science 41, 2009, pp. 109-120

Received 05/11/08
Accepted 09/12/08
Final version 05/02/09

2 Data and Existing Approaches

As mentioned above, our interest targets verbs and their nominalisations. In this study, we focus only on two types of predicates: verbs and nominalisations which occur freely in a sentence. The same methods can be applied for the analysis of nominalisations within a support verb constructions, which is a task for our future work. Subcategorisation properties of verbs and nouns have been described in many linguistic and NLP studies. There exist various works on verb valency in NLP approaches (e.g. [1], [2], [3], [4], [5] and [6]). Most of them concentrate on English verbal predicates, but there exist studies for other languages as well, e.g. [7], [8], and [9] for German or [10] and [11] for Italian.

Nominalisations are also described in many studies, for instance, [12], [13], etc. for English, [14] and [15] for German nominalisations.

3 The Phenomenon of "Inheritance" in Subcategorisation

The phenomenon of "inheritance" of subcategorisation was mostly studied within the relationships of verbs and their nominalisations, deverbal nouns which are morphologically derived from verbs by affixation, and which often share much of their meaning with the base verbs. Many authors who analyze nominalsiations, e.g. [12], [14], [15], [13], mention correspondences between arguments of nominalisations and those of their underlying verbs, depending on the type of complements and the classes of verbs under analysis.

However, only a few lexical resources provide systematic correspondences between verbs and their nominalisations. For instance, [16] describes a computational lexicon of nominalisations NOMLEX which maps noun roles into the predicate-argument structure of their associated verbs. Another example is the analysis described in [17], where the authors use the PARC's text processing system for the process of mapping the predicate-argument structure of nominalisations and that of their base verbs.

In NOMLEX, we find two types of nominalisations depending on the ability to absorb the arguments of the base verb: VERB-NOM for those that appear with many or all verbal complements, and NOM-TYPE for those nominalsiations that can "inherit" only one of the arguments of the base verb. That shows that some deverbals only partially take over verbal valency patterns,. thus, there are also non-correspondences in the predicate-argument structures of a nominalisation and its base verb.

Our preliminary extraction tests also show that there are both correspondences ("inheritance") and differences ("non-inheritance") in the subcategorisation of morphologically related predicates.

In many cases subcategorisation properties of deverbal nominal predicates are "inherited" from their base verbs (example (1)).

(1) – *begründen, dass/w-...* ("to justify that/wh-...")
vs. *Begründung, dass/w-...* ("justification that/wh-...")

- *befürchten, dass...* (" to fear that...")
  vs. *Befürchtung, dass...* ("fear that...")
- *erklären, dass/w-...* ("to explain that/wh-...")
  vs. *Erklärung, dass/w-...* ("explanation that/wh-...")

But there are also cases where subcategorisation of a nominalisation differs from that of its base verb (cf. (2))

(2)
- *vorstellen, dass/w-...* ("to think that/wh-...")
  vs. *die Vorstellung, dass/*w-...* ("idea that/*wh-...")
- *vermuten, dass/w-...* ("to suppose that/wh-...")
  vs. *die Vermutung, dass/*w-...* ("supposition that/*wh-...")

All the above mentioned cases should be analysed and considered in the mapping rules for predicate-argument structure. Linking the predicate-argument structure of such deverbals like in (2), with the predicate-argument structure of their base elements, we should take into account that subcategorisation properties of verbs underlying deverbals in these cases can not be just transferred and reapplied.

4 Methods and Tools

4.1 Input and Context

For this study, we use a corpus of German texts consisting of newspaper texts from Germany which include extracts (1992–2000) from *die tageszeitung* ('taz', 111M), *Frankfurter Rundschau* ('FR', 40M), *Frankfurter Allgemeine Zeitung* ('FAZ', 71M).

All corpora are pre-processed: sentence-tokenised, tagged for part-of-speech, lemmatised and partially chunked[1]. Extraction queries in the form of regular expressions rely on the Stuttgart CorpusWorkBench (CWB, [22]). As extraction context for verbal predicates, we chose German verb-final clauses (VL) (in this case, the subcategorised subclause usually follows the verb, cf. Table 1) and passive sentences (where we have a regular sequence of elements, and the subclause follows the 2nd part of the verb, cf. Table 2).

Table 1. *Dass*-clause after a verb in VL

main clause		**subclause**
	verb	
DE: *Wenn sie*	*erfahren,*	*dass John Miller große Mengen Alkohol kauft...*
EN: "If they"	"found out"	"that John Miller buys much alcohol..."

[1] For annotations we used the Tokeniser of [18], Tree-Tagger described in [19] and [20] and YAC-Chunker [21]

Table 2. *Dass*-clause after a verb in passive

main clause				subclause
	verb: 1st part		verb: 2nd part	
DE: *Es*	*muss*	*heute*	*gesagt werden,*	dass der Nikolaus ein Türke ist.
EN: "It"	"should be"	"today"	"told"	"that Santa Claus is Turk."

Nominalisations are extracted in Vorfeld construction (VF), a clause initial position before the finite verb in German declaratives. If a noun in VF is followed by a subclause, this subclause can only be subcategorised by the noun (see Table 3).

Table 3. *W*-clause after a noun in VF

main clause: 1st part noun phrase	subclause	main clause: 2nd part the rest
DE: *Die Erklärungsversuche,*	*warum der Teufel sich an X heranmacht*	*sind auf der Glatze gedrehte Locken.*
EN: "The explanation attempts",	"why the devil chats up X"	"are as futile as giving a bald man a comb."

4.2 Extraction and Classification Architecture

Extraction and Classification of Nominalisations. We automatically extract predicates from text corpora classifying them according to their subcategorisation properties. The extraction steps proceed from the general to the specific.

For the extraction and classification of "inheritance" relations, we start with the analysis of nominalsations, extracted in VF. They are classified into the three groups shown in Table 4.

Table 4. Classification of nominalisations extracted in VF

type	subcategorisation properties
N1	nominalisations that subcategorise only for a *dass*-clause
N2	nominalisations that can take all the three sentential complements
N3	nominalisations with which a *dass*-clause was not found

Extraction and Classification of Base Verbs. With the help of morphological tools, e.g. SMOR, [23], we get a list of base verbs underlying the nominalisations extracted in Vorfeld constructions.

Table 5. Nominalisation-verb pairs after SMOR analysis

nouns vs. verbs		**translation**	
Ankündigung	– *ankündigen*	"announcement"	– "to announce"
Bedingung	– *bedingen*	"condition"	– "to condition"
Befürchtung	– *befürchten*	"fear"	– "to fear"
Erwartung	– *erwarten*	"expectation"	– "to expect"
Entscheidung	– *entscheiden*	"decision"	– "to decide"
Erklärung	– *erklären*	"explanation"	– "to explain"
Darstellung	– *darstellen*	"presentation"	– "to present"
Vermutung	– *vermuten*	"assumption"	– "to assumpt"
Vorstellung	– *vorstellen*	"idea"	– "to think"

The generated list of base verbs is integrated into the query for verb extraction. We lexically specify the constraints for the verbal predicate extraction (line 3 in Fig. 1) adding the generated base verbs list $base_verbs (line 3b.).

Query building blocks	**comments**	**matching sentence**	**translation**
1. [pos="KOU.*|PREL.*|PW.*"]	conj., relat. or inter. pronoun	*weil*	"because"
2. [pos!="V.*FIN"&word!=",|-"]*	optional, no fin. verbs or punctuation	*nicht mehr die Parlamentarier selbst künftig darüber*	"in the future not even the parlament members themselves"
3a. <vc>...	verb		
3b. [lemma=RE($base_verbs)]	complex	*entscheiden*	"deside"
3c. ...</vc>		*sollen*	"must"
4. ","	comma	,	,
5. [(pos="PW.*") | (word="ob") | (word="dass")]	rel. pronoun or conj. "ob" or conj. "daß"	*wieviel*	"how much"
6. [pos!="V.*FIN"]*	optional, no fin. verbs	*Geld sie*	"money they"
7. [pos="V.FIN*"]	fin. verb	*bekommen*	"become"
8. [pos="$."]	sent. end	.	.
9. within s;	within a sent.	sentence context	

Fig. 1. Query for base verbs in VL subcategorizing for a *dass/ob/w*-clause

The system searches for base verbs subcategorising for all three complement types (*dass, ob* and *w*-clauses). The list of extracted verbs (with frequency data) is used for the subsequent comparison of subcategorisation properties of the extracted verbs and those of their nominalisations. Base verbs are also classified into three groups according to their subcategorisation properties, as seen in Table 6.

Table 6. Classification of base verbs

type	subcategorisation properties
V1	verbs that subcategorise only for a *dass*-clause
V2	verbs that can take all the three sentential complements
V3	verbs with which a *dass*-clause was not found

Classification and description of Subcategorisation Relations. We analyse the relations between the subcategorisation properties of verbs and those of their nominalisations as it is shown in Table 7.

Table 7. relations between verbs and their nominalisatons

relations	description of subcategorisation relations
V1N1	nominalisation and its underlying verb subcategorise only for a *dass*-clause.
V2N1	the base verb has all three (or two) complement types but the nominalisation has only a *dass*-clause (the loss of *ob, w*-clauses).
V3N1	the base verb has no *dass*-clause but its nominalisation has a subcategorised *dass*-clause.
V1N2	the base verb has only a *dass*-clause (found in corpora), but its nominalisation has all three (or two) complement types.
V2N2	the base verb has all three (or two) complement types, so does its nominalisation (V1N1 and V2N2 – similar relations).
V3N2	the base verb has no *dass*-clause, but its nominalisation has all three (or two) complement types.
V1N3	the base verb has only a *dass*-clause, but its nominalisation doesn't have any *dass*-clause.
V2N3	the base verb has all three (or two) complement types (including the *dass*-clause), but the nominalisation has no *dass*-clause.
V3N3	the base verb does not have a *dass*-clause, neither does its nominalisation (V1N1 and V3N2 – similar relations).

Classification of "Inheritance" Relations. We classify the relations between the subcategorisation properties of nominalisations and those of their base verbs described above into the three following groups:

R1 subcategorisation properties are "inherited" from the verb
(V1N1, V2N2, V3N3):
entscheiden, dass/ob/w- ("to decide that/if/wh-")
vs. *Entscheidung, dass/ob/w-* ("decision that/if/wh-")

R2 subcategorisation properties are "inherited" with the loss of clauses by the nominalisation:
- loss of *ob/w*-clauses (V2N1):
  ankündigen, dass/w- ("to announce that/wh-")
  vs. *Ankündigung, dass* ("announcement that")
- loss of *dass*-clauses (V2N3, V1N3):
  ermitteln, dass/ob/w- ("to investigate that/if/wh-")
  vs. *Ermittlung (darüber), ob* ("investigation (about) if")

R3 subcategorisation properties are "inherited" from the verb, but the nominalisation has additional subcategorisation properties of its own
(V3N1, V1N2, V3N2):
darstellen, w- ("to present wh-")
vs. *Darstellung, dass/w-* ("the presentation that/wh-")

5 Results

5.1 Extraction Results and their Interpretation

Subcategorisation of deverbal nouns is "inherited" from their base verbs in most cases. Table 8 contains examples of R1 relation type. Subcategorisation properties of nominalisations *Bedingung* and *Befürchtung* which occur only with a *dass*-clause, as well as subcategorisation properties of the nominalisations *Entscheidung* and *Erklärung* which occur with all three complement types, correspond with subcategorisation properties of their base verbs *bedingen, befürchten, entscheiden* and *erklären*. Hence, subcategorisation of nominalisations is "inherited" from their base verbs.

Table 8. Examples of type R1 relations

predicate	**translation**	**subclause**		
		dass	*w-*	*ob*
bedingen	”to condition”	+	-	-
Bedingung	”condition”	+	-	-
befürchten	”to fear”	+	-	-
Befürchtung	”fear”	+	-	-
entscheiden	”to decide”	+	+	+
Entscheidung	”decision”	+	+	+
erklären	”to explain”	+	+	+
Erklärung	”explanation”	+	+	+

Table 9 shows cases when a nominalisation takes over only a part of the base verb's subcategorisation (R2 relation type). For instance, the verbs *ankündigen, erfahren* and *fordern* subcategorise for two or three sentential complements, whereas their deverbals *Ankündigung, Erfahrung* and *Forderung* occur only with a *dass*-clause.

Table 9. Examples of type R2 relations

predicate	**translation**	**subclause** *dass w- ob*
ankündigen	"to announce"	+ + -
Ankündigung	"announcement"	+ - -
erfahren	"to find out"	+ + +
Erfahrung	"experience"	+ - -
fordern	"to claim"	+ + -
Forderung	"claim"	+ - -

The R3 relations cases, when nominalisations get some additional properties are very seldom and sometimes difficult to detect.

In Table 10, we otline frequency data for some cases extracted in 'FR', 'FAZ' and 'taz'. The occurrence of nominalisations in VF subcategorising for *dass, ob* or *w*-clauses is compared with the occurrence of their base verbs in VL (see Sect. 4.1).

Table 10. Predicates extracted from German corpora (ca. 220M)

relations	**predicates**	**translation**	**TOTAL**	*dass*	*w-*	*ob*
			abs.	**in%**	**in%**	**in%**
R1	*bedingen*	"to condition"	**100**	100,00	0	0
	Bedingung	"condition"	**85**	98,82	1,18	0
	fragen	"to ask"	**786**	0	97,33	2,67
	Frage	"question"	**1631**	0	26,98	73,02
R2	*erfahren*	"to find out	**4826**	80,90	14,67	4,43
	Erfahrung	"experience"	**124**	96,77	1,61	1,61
	vorstellen	"to think"	**100**	32,00	68,00	0
	Vorstellung	"idea"	**81**	100,00	0	0
	vermuten	"to assumpt"	**20**	70,00	30,00	0
	Vermutung	"assumption"	**76**	100,00	0	0
	regeln	"to settle"	**14**	42,86	57,14	0
	Regelung	"settlement"	**19**	100,00	0	0
R3	*beweisen*	"to evidence"	**65**	36,92	63,08	0
	Beweis	"evidence"	**65**	95,38	0	4,62
	darstellen	"to present"	**14**	0	100,00	0
	Darstellung	"presentation"	**9**	77,78	22,22	0

Table 10 reveals that the verb *bedingen* never occur with *w*- or *ob*-clauses. Neither does its nominalisation *Bedingung.* Both the deverbal noun *Frage* and its base verb *fragen* subcategorise for *w*- and *ob*-clauses, and never take a *dass*-clause as a complement.

The verb *erfahren* and its nominalisation *Erfahrung* show preferences for a *dass*-clause (in ca. 81% and ca. 97% of cases) as well. However, ca. 15% of the verb occurrences are found with a *w*-clause, whereas only ca. 2% of its nominalisations occur with this complement type. The nomilnalisation *Erfahrung* seems to "inherit" only a *dass*-clause from the base verb. Further examples of "non-inheritance" are nominalisations *Vorstellung, Vermutung* and *Regelung*, which subcategorise only for a *dass*-clause, whereas their base verbs occur also with other complement types.

Subcategorisation of deverbals *Darstellung* and *Beweis* also differs from that of their base verbs *darstellen* and *beweisen.* The verb *darstellen* occurs only with a *w*-clause (100%) in our corpora, whereas its deverbal can subcategorise both for a *w*- and a *dass*-clause (22% and ca.78% respectively).

Ca. 95% of the occurrences of *Beweis* and only ca. 37% of occurrences of *beweisen* are found with a *dass*-clause. The verb *beweisen* shows preference for *w*-clauses (with 63%), whereas *Beweis* occurs with *ob*- and never with *w*-clauses in the analysed corpora.

5.2 Reasons for "non-inheritance"

One of the reasons for "non-inheritance" among nominalisations lies in their semantics. Most *ung*-nominalisations (e.g. *Erfahrung, Forderung, Vorstellung, Vermutung* ("experience, idea")) express a proposition, a fact, and the subcategorised *dass*-clause is their "content" (e.g. *Bedingung, Erfahrung, Vorstellung*). *W*- amd *ob*-clauses presuppose an open set of answers which doesn't correspond to the semantics of "fact"-nominalisations.

The meaning of "fact"-nominalisations can be introspectively tested with the help of deletion tests. A nominalisation in Vorfeld is deleted in front of its subcategorised subclause. If the complement clause can be used without the nominalisation, this nominalisation has a "fact"-reading (cf. (3a) and (3b)). Otherwise it has a "non-fact"-reading (cf. (4a) and (4b)).

(3a) *Für die Vermutung,* ***dass die Krawalle von rechts inszeniert worden seien,*** *spreche auch...*
("In the favour of the assumption **that the riots were organized by right-wingers** militates also...")
vs.

(3b) *Dafür,* ***dass die Krawalle von rechts inszeniert worden seien,*** *spreche auch...*
("In the favour of **that the riots were organized by right-wingers** militates also...")

(4a) *Die Überlegung,* ***ob Mullvorfahren von Afrika nach Lateinamerika über das Meer getrieben worden sein könnten,*** *ist hypothetisch.*
("The consideration **if the ancestors of moles floated from Africa to Latin America by sea** is hypothetic.")
vs.

(4b) ****Ob Mullvorfahren von Afrika nach Lateinamerika über das Meer getrieben worden sein könnten,*** *ist hypothetisch.*
("**If the ancestors of moles floated from Africa to Latin America by sea** is hypothetic.")

6 Treatment in NLP Lexicon Building

The phenomena described above should receive a specific treatment in NLP lexicon building. Classification of "inheritance" relations described in 4.2 limits the need for spelling out all subcategorisation properties of nominalisations.

Subcategorisation indications for nominalisations of all three relation types (from R1 to R3) should contain references to subcategorisation of the base verbs. A special note about the loss of certain properties should be included into the entry for R2 nominalisations, whereas entries for R3 nominalisations should contain a note about additional properties that the verb does not have.

7 Conclusion

Our experiments showed that although "inheritance" of subcategorisation properties from verbs to nominalisations is widespread, some morphologically derived predicates can have their own subccategorisation properties, which are not "inherited" from the verbs. These phenomena should receive a specific treatment in NLP lexicon building.

The system described above, allows us to extract and classify such cases semi-automatically according to their subcategorisation relations. It is possible to identify such cases automatically by means of extracting them from tokenised, pos-tagged and lemmatised text corpora.

Our future work will include extraction procedures on a larger corpora to achieve substantial coverage, and a deeper semantic analysis of nominalisations and possible reasons for the "non-inheritance" cases. We also intend to study contextual properties of predicates (e.g. polarity or modality) which can influence the subcategorisation properties of nominalisations. The future tests should include not only nominalisations that appear freely in a sentence but also support verb constructions which contain nominalisations, e.g. *unter Beweis stellen, in Erfahrung bringen*, etc.

References

1. Brent, M.: From grammar to lexicon: Unsupervised learning of lexical syntax. Computational Linguistics **19(2)** (1993) 243–262
2. Ushioda, A., Evans, D., Gibson, T., Waibel, A.: The automatic acquisition of frequencies of verb subcategorization frames from tagged corpora. In: Proceedings of the Workshop on the Acquisition of Lexical Knowledge from Text, Columbus, OH (1993) 95–106
3. Manning, C.: Automatic acquisition of a large subcategorization dictionary from corpora. In: Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics, Columbus, OH (1993) 235–242
4. Briscoe, T., Carroll, J.: Automatic extraction of subcategorization from corpora. In: Proceedings of the 5th ACL Conference on Applied Natural Language Processing, Washington, DC. (1997) 356–363

5. Carroll, G., Fang, A.: The automatic acquisition of verb subcategorisations and their impact on the performance of an hpsg parser. In: Proceedings of the 1st International Joint Conference on Natural Language Processing, Sanya City, China (2004) 107–114
6. O'Donovan, R., Burke, M., Cahill, A., van Genabith, J., Way, A.: Large-scale induction and evaluation of lexical resources from the penn-ii and penn-iii treebanks. Computational Linguistics **31(3)** (2005) 329–365
7. Schulte im Walde, S., Brew, C.: Inducing german semantic verb classes from purely syntactic subcategorisation information. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, PA (2002) 223–230
8. Wauschkuhn, O.: Automatische Extraktion von Verbvalenzen aus deutschen Textkorpora. PhD thesis, Universität Stuttgart: Institut für Informatik (1999)
9. Eckle-Kohler, J.: Linguistic Knowledge for Automatic Lexicon Acquisition from German Text Corpora. PhD thesis, Universitat Stuttgart: IMS (1999)
10. Ienco, D., Villata, S., Bosco, C.: Automatic extraction of subactegorization frames for italian. In: Proceedinga of LREC-2008, Marrakech, Marrokko (2008)
11. Lenci, A., McGillivray, B., Montemagni, S., Pirrelli, V.: Unsupervised acquisition of verb subcategorization frames from shallow-parsed corpora. In: Proceedinga of LREC-2008, Marrakech, Marrokko (2008)
12. Nunes, M.: Argument linking in english derived nominals. In Valin, R.V., ed.: Advances in Role and Reference Grammar. John Benjamins (1993) 375–432
13. Meinschaefer, J.: The syntax and argument structure of deverbal nouns from the point of view of a theory of argument linking. In Dal, G., Miller, P., L. Tovena, L., de Velde, D.V., eds.: Deverbal nouns. John Benjamins, Amsterdam forthcoming.
14. Ehrich, V., Rapp, I.: Sortale bedeutung und argumentstruktur: ungnominalisierungen im deutschen. Zeitschrift für Sprachwissenschaft **19** (2000) 245–303
15. Schierholz, S.: Präpositionalattribute. Syntaktische und semantische Analysen, Tübingen (2001) Linguistische Arbeiten 447.
16. Macleod, C., Grishman, R., Meyers, A., Barrett, L., Reeves, R.: Nomlex: A lexicon of nominalizations. In: Proceedings of EURALEX-98, Liege, Belgium (1998) http://nlp.cs.nyu.edu/nomlex/index.html.
17. Gurevich, O., Crouch, R., King, T., de Paiva, V.: Deverbal nouns in knowledge representation. Journal of Logic and Computation Advance Access (December 20 2007)
18. Schmid, H.: Unsupervised learning of period disambiguation for tokenisation. Internal Report (2000)
19. Schmid, H.: Probabilistic part-of-speech tagging using decision trees. In: International Conference on New Methods in Language Processing, Manchester, UK (1994) 44–49
20. Schmid, H.: Improvements in part-of-speech tagging with an application to german. In Armstrong, S., Church, K., Isabelle, P., Manzi, S., Tzoukermann, E., Yarowsky, D., eds.: Natural Language Processing Using Very Large Corpora. Volume 11 of Text, Speech and Language Processing. Kluwer Academic Publishers (1999) 13–26
21. Kermes, H.: Offline (and Online) Text Analysis for Computational Lexicography. PhD thesis, Universität Stuttgart: IMS (2003) AIMS.
22. Evert, S.: The CQP Query Language Tutorial. Universität Stuttgart: IMS. (2005) http://www.ims.uni-stuttgart.de/projekte/CorpusWorkbench/CQPTutorial/html/.
23. Schmid, H., Fitschen, A., Heid, U.: Smor: A german computational morphology covering derivation, composition, and inflection. In: Proceedings of LREC-2004, Lisboa, LREC (2004)

Discovering Discourse Motifs in Instructional Dialog

Juan M. Huerta

IBM T.J. Watson Research Center
1101 Kitchawan Road, Yorktown Heights, NY 10598, USA
huerta@us.ibm.com

Abstract. We propose a method to analyze conversational interaction using discourse motifs (sequence of labels). We focus specifically on instructional transactive discourse. We first describe the characteristics of transactive discourse, its relationship to other frameworks of instructional discourse, and introduce a refined taxonomy of transactive discourse. Based on this new taxonomy, we construct a set of classifiers to automatically label instructional dialog segments. After labeling, we search for salient patterns of discourse common to these chains of labels using Multiple EM for Motif Elicitation and Gapped Local Analysis of Motifs (which are two techniques available for DNA and protein motif discovery). From our analysis of a corpus of classroom data, a set of Transactive-Participatory-Coherent motifs emerge. This approach to interaction-motif discovery and analysis can find application in dialog and discourse analysis, pedagogical domains (e.g., assessment and professional development), automatic tutoring systems, meeting analysis, problem solving, etc.

1 Introduction

We focus on the analysis of classroom discourse particularly when the focus is on solving mathematical problems. While the analysis of classroom discourse and mathematical problem solving is useful in providing pedagogical insight into teaching practices (see for example Huerta (2008), Blanton (2008)), its analysis can also shed light into interaction mechanisms used in more general collaborative problem solving.

Research in human dialog has been approached from various viewpoints using frameworks and methodologies of analysis that have been tailored to address the specific requirements of these viewpoints (examples of relatively recent perspectives to dialog analysis include Stolcke (2000), Stent (2000) among others, and a good summary can be found in Moore (2003)).

More problem-solving specific frameworks have also been proposed to analyze planning-oriented and instructional dialog in the classroom (Linden (1995)). Additionally, there have also been other efforts in the manual analysis of classroom interaction from purely pedagogical and sociological perspectives (Blanton (2008), Mehan (1985), Stark (2002), Haussman (2003)). There has been also work focusing on specific theoretical frameworks of interaction and the correlation of their elements to individual learning (e.g., Haussman (2006) and elaborative discourse, Meyer (2002) and scaffolding and self-regulation) as well as development of discourse frameworks

© A. Gelbukh (Ed.)
Advances in Computational Linguistics.
Research in Computing Science 41, 2009, pp. 123-134

Received 11/11/08
Accepted 11/12/08
Final version 07/02/09

for the analysis of tutoring speech and implementation of tutoring systems (e.g., Marineau (2000)) .

The specific focus of this paper is around discourse that occurs inside a classroom when the teacher guides and regulates problem-solving activities with the students. We look into Mathematics classes when the classroom is collaboratively solving a problem under the guidance of the instructor. We propose a taxonomy of instructional discourse acts that is specific to this domain and focus on transactive and coherence elements and use this taxonomy to label discourse. The result of this labeling is a set of strings, or linear sequences of labels. We then apply techniques for the discovery of motifs (strong patterns) in these strings. The goal is to extract motifs that can be of interest and help us identify strong or salient patterns. Because our taxonomy is based specifically on transactive and coherent discourse, the motifs that emerge during our data analysis strongly highlight these characteristics. Motifs discovered in this fashion can be used as features of further stages of discourse analysis in support of applications in the areas of problem solving, tutoring systems, meeting analysis, as well as purely pedagogical ones.

While the techniques developed for essay analysis (e.g., Burstein (2003) and Burstein (2003b)) address a different series of issues (due to the differences between essay discourse and classroom interaction), some basic ideas (like the relevance of coherence discourse) can be utilized for the analysis of classroom interaction.

This paper is organized as follows, in Section 2 we present a general overview of the main existing approaches that are relevant to this paper; specifically, we describe the framework of transactive discourse based on Blanton (2008) and Huerta (2008). In section 3 we describe in detail the particular taxonomy labels that we use in later sections of this paper and describe the classification techniques we used in order to label our data. In section 4 we describe the methods we use to discover motif sequences in the labels of classroom discourse. In section 5 we describe the results of the analysis of data and the most salient motifs of this discourse and illustrate how these motifs can be utilized in dialog analysis applications. And finally in section 6 we conclude our paper with a summary of the contributions of this paper, a discussion of results observed and a discussion of future directions.

2 Relevant Approaches

In this section we briefly describe some of the existing theories and abstractions that are most related to this paper, specifically RST, elaborative-collaborative dialog, and transactive dialog.

In the area of theories of discourse analysis, Rhetorical Structure Theory is quite relevant to the type of discourse we focus on; specifically, Stent (2000) proposed the application of RST for content-planning of mixed-initiative task-oriented dialogs (TRIPS dialogs). RST is a descriptive theory of hierarchical structure in discourse that identifies functional relationships between discourse parts based on the intentions behind their production (Mann (1987)). While the discourse activity that occurs in the classroom in the context of mathematic problem solving to have many commonalities with the sort of mixed-initiative, task oriented, content-planning characteristics of a domain like TRIPS, a much simpler taxonomy to the classroom data suffices.

Hausmann (2006), focuses on measuring the effect that elaborative and collaborative dialogs have on learning and understanding. In his literature review, he says that previous research has found that *only certain* collaborative dialogs have been found to have strong gains in understanding. He says that while elaborative dialog has been shown to impact individual learning, for collaborative learning the results have shown no correlation with deeper measure techniques can be trained and lead to deep learning outcomes.

Transactive reasoning is defined as discourse in which the participant continues the reasoning, analysis or interpretation of the discussion and which possibly leads into or motivates further transactive discourse (Blanton 2008). Berkowitz (1983) describe transactive dialogs as, "reasoning that operates on the reasoning of another". Co-construction qualifies as a transactive dialog because the listener takes the speaker's message as input, manipulates it, and produces an output based on, yet separate from, the original input (Salomon, 1993).

We can see then, that elaborative dialog is a subset of the transactive discourse and that frameworks focusing on transactive discourse are adequate for analyzing mathematical problem solving in the classroom.

In terms of abstractions for analysis, Truxaw (2004) and DeFranco (2007) describes recursive discourse cycles as components of a cyclical process. The authors describe that in their observed data this cyclical process serves an inductive purpose (to move from the particular to the general hypothesis and rules). For this purpose they rely on the concept of a sequence map, which is a machine that produces the observed sequence. In Truxaw and DeFranco (2002) a sociolinguistic framework is used to analyze classroom speech using sequence maps.

We have defined as an interaction motif as sequence of labels that describe the discourse given a taxonomy and a coherent portion of discourse. A sequence map is the finite state machine that accepts such motif. In this paper we will focus on the mechanisms of discovery of motifs, and such motifs can be abstracted into sequence maps.

3 Taxonomy of Transactive Discourse

In this section we describe the basic taxonomy of problem-solving oriented classroom discourse that we use in this paper. It is based in (Blanton (2008) and Huerta (2008)).

3.1 Taxonomy

The basic components of the taxonomy are described in terms of mutually exclusive characteristics or labels. The basic characteristics/labels are:

- **Transactive Teacher Prompt:** Question or prompt in which the instructor *elicits* continued reasoning, analysis or interpretation of the discussion and which response possibly leads into or motivates further transactive discourse

- **Transactive and Non-transactive Student Response:** A student participates in a non-trivial way. It can be both transactive response, as in providing further elaboration to the thinking and discussion process, or it can be non-transactive, like a direct yes-no response to a question.
- **Student-Coherence Teacher Discourse:** The instructor implicitly validates or emphasizes the student utterance by repeating verbatim or paraphrasing part, or the whole, of what the student has said. This is related to coherence in essays and in text to work by Barzilay (2006), Higgins (2004), Grosz (1995), and Higgings (2006).
- **Explicit Teacher Validation:** The Instructor explicitly validates a student response by using yes-no utterances or equivalent expressions (e.g., ‘sure”, “of course”, etc).
- **Other (Instructive+Directive):** This is a catch-all category absorbs all the teacher’s utterances that fall mostly in the instructive and directive categories. Instructive utterances are those that the teacher uses to lecture, or teach. Directive utterances are those that the teacher uses to provide overall direction of activities.

The labels above are not meant to be exhaustive; hence, the *other* category. In the discourse there are other possible labels, but for now we focus on these. A single utterance can combine more than one of the characteristics above: e.g., a teacher might say in a single utterance: “That’s quite a good guess, anybody else has a different answer?” which would simultaneously correspond to Explicit Validation and Transactive Teacher Question categories. We will explain further below how we code this.

Thus, we map sentences with the characteristics above to sequences of 5 characters (or labels). The character order is very important for motif analysis. Table 1 shows the maps from characteristics or labels to characters for each utterance.

3.2 Classification and Labeling Approaches

Here we describe how the labels are generated. Due to the relative simplicity of the classroom speech, most of these classifiers are quite simple.

- Transactive teacher prompt: We could have relied on bag-of-words Maximum Entropy approaches to utterance classification (like Wu et al., 2003), but we noticed that most of the time, in classroom speech, simple rule-based approaches suffice (key-words, key-phrases, and lexical patterns), i.e., we look for words like “what is” (Spoerleder (2005) analogously looks into lexical cues for rhetorical relations)
- Student: Trivial mapping generated from the speaker identification.
- Student-Coherence Teacher discourse: Substantial work has been done in this area for document/essay coherence (Higgins 2004, Higgins 2006, Barzilay 2008). Bag of word comparisons based on frequencies or on word rank orders (Huerta 2008) are possible.
- Explicit teacher validation: Similarly, simple keyword suffices, but also other classification approaches (like Maximum Entropy) could be used.

- Other (Instructive+Directive): This is not performed using an actual classifier, but rather is the remainder of the teacher's utterances that are not transactive, coherent-student, or validation. An utterance will fall in this category if none of the features used to identify the other teacher labels are found.

Table 1. Label-to-string mapping

	Transactive Teacher Prompt	Student Response	Student Coherence Teacher	Explicit Reaffirmation	Directive+ Instructive Teacher
S		1			
T	1				
C			1		
YC			1	1	
YCT	1		1	1	
YT	1			1	
CT	1		1		
Y				1	
X					1

4 Methods for Motif Discovery

We have defined a motif as a strong recurring pattern in the sequence of characters. As our taxonomy defines labels in terms of transactive roles in the discourse, we expect that the patterns that emerge reflect transactive motifs that provide insight into the interaction. Depending on the approach, a Motif can be permitted to have gaps and insertions and deletions, as well as to make various assumptions regarding the minimum and maximum times a motif occurs as well as the location of the motif (context). We used specifically the MEME and GLAM approaches which we detail below.

4.1 Multiple EM for Motif Elicitation

The MEM method (Bailey 1994, Bailey 2006) [1] is a Maximum Likelihood based approach to motif discovery. It works with the assumption that motifs occur zero or more times in the data. It uses a two component finite mixture model. One component models the probability that each position in a segment of length n in the sequence was generated *independently* by a position-specific multinomial random trial variable. The background model has a similar multinomial random variable but it is not position specific. The dataset over which the models are trained consists of all possible over-

[1] http://meme.nbcr.net/meme4/cgi-bin/meme.cgi

lapping segments of length *n* in the data. There are constraints in place to ensure that the model does not predict that two overlapping sequences were predicted by the same motif, as well as to reduce its bias to sequences of one or two letters.

4.2 Gapped Local Analysis of Motifs

GLAM (Frith 2008) [2] searches for key positions in the input sequences optimizing this number of key positions. Each sequence string contributes only zero or one substrings to the alignment. GLAM maximizes the alignment score which includes penalizations for insertions and deletions, penalizing less if these are clustered together. The model has position specific residue (character generation) probabilities as well as position-specific insertion and deletion probabilities. A Beta distribution for priors is assumed. Search is performed using stochastic annealing.

5 Experiments

In this section we describe the experiments we perform on motif discovery. We used as a corpus data from a college course on Discrete Mathematics at freshman level that was recorded and manually transcribed (Blanton 2008). Four segments were identified for analysis. These segments originated in four different lectures. These segments comprised a total of 1000 turns (utterances), more than 18,000 words (tokens) and around 100 minutes of classroom interaction.

5.1 Discourse Labeling and Sequence Generation

Each segment was classified as described in section 3.2. Labels were converted into sequences of characters (one utterance was allowed to generate more than one character). Figure 1 below shows a level plot corresponding to the instructional discourse labels found in one of the for segments, which consisted of 331 events (or utterances). In this figure, each dot at level 0.8 represents a Transactive and non-transactive student response, a dot at level 1.0 represents a Transactive-Teacher prompt, a dot at level 1.1 represent a student-coherence teacher response, a dot at level 1.2 represents an explicit teacher validation, and a dot at level zero represents *other*. In this example, it is very clear from the figure that there *are* patterns of interaction present in the classroom data. Through motif analysis, we will show how to identify those motifs. Motifs can be used as features in analysis that address questions like: what is the effect or correlation of a certain motif in the future student response? What are the characteristic discourse patterns that emerge for a specific teacher? Are teachers A and B using similar interaction strategies in the classroom?

[2] http://meme.sdsc.edu/meme4/cgi-bin/glam2.cgi

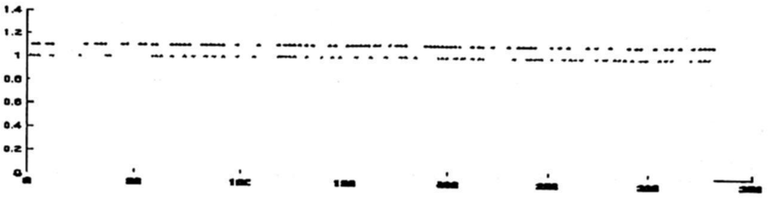
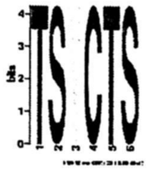


Fig. 1. Level plot of instructional discourse labels for a lecture segment.

5.2 Motif Discovery

We now analyze the four segments using MEM and GLAM. MEM allows for any number of repetitions to be present in the data. We first specified a minimum motif length of 4, a maximum of 6. The main pattern found is TCYCTS, which means a Transactive teacher prompt, followed by student participation, followed by explicit affirmation, then coherent teacher discourse (and then a fresh Transactive and Coherent labels).



The relative entropy of the motif relative to a uniform background frequency model is 25.9 bits. It was found 20 times in the data. When we limited our search exclusively to motifs of length 4, the result is CTSY, which is essentially included in the 6 character pattern originally found. One could argue that the core of this pattern is TSYC, or even more simply TSC. The block diagram, showing the occurrences of exactly the TSYCTS motif in the data is shown below:

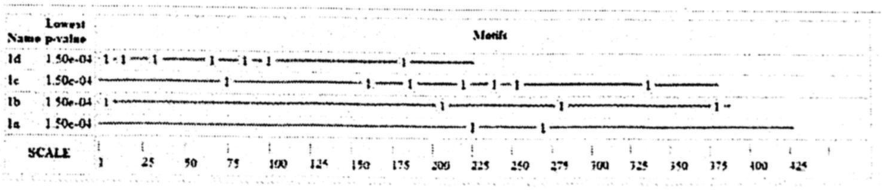
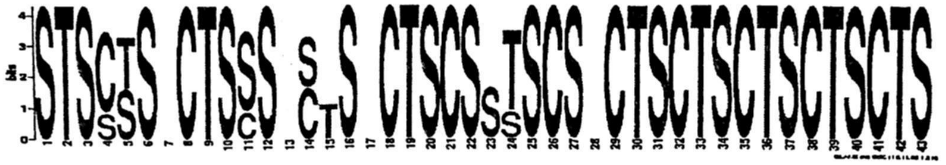
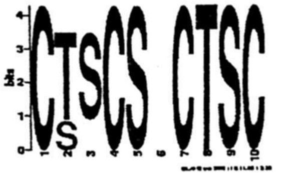


Fig. 2. Block diagram displaying the location of the occurrences of the TSYCTS motif in the data.

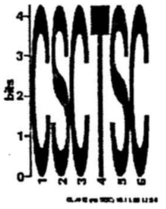
GLAM allows for insertions and deletions and thus is able to provide longer runs. The results of the GLAM analysis are very different from the MEM results and can be used to supplement each other. The parameters we used initially for GLAM are: Minimum number of sequences in the alignment=2, Min. Num. of aligned columns=50, Initial num of aligned cols =4, num alignment runs=40. The Result is shown below.



When constrained to find a shorter motif the result is:



Reducing further the length of the found motif:



GLAM provides, in addition to the best motif, the top alignments in the data allowing for deletions and insertion. The motif found by MEME is alignment #40 in GLAM with score 58.49. In other words CSCTSC is the most generalizable pattern under insertions and deletions.

Considering this alignment that exist in both MEME and GLAM both approaches produced very consistent results.

So far, we have just applied two techniques to extraction and discovery of motifs. Now we are interested in applying these newly discovered motifs to further analyze the data.

5.3 Motifs as Discourse Analysis Features

We have shown how to discover and extract interaction motifs. These motifs can be then used as features in the analysis of the discourse. In this section we present a simple example of such analysis. For this purpose, we define two parameters we are interested in analyzing: the smoothed participation index and the smoothed transactive coherent (TSC) pattern. The smoothing of the TSC pattern is defined simply as a sort of asymmetrically *charging* and discharging a virtual capacitor in which if either motifs TSYC and YSC are found in the dialog the value of function increases (charges) at a certain rate and if not it decreases (discharges) at another rate, i.e. if the smoothed TSC at time i is denoted by s_i,

$$s_i = \begin{cases} 0.8 + 0.2s_{i-1} & \text{if TSC patterns occur at time i} \\ 0.9s_{i-1} & \text{if TSC pattern does not occur at time i} \end{cases}$$

The participation index is defined similarly as the smoothed TSC pattern, except that it will charge at time *i* if a student event occurs then and it will discharge otherwise. Ideally, for balanced participation, this index should have value 0.5. Below we show the

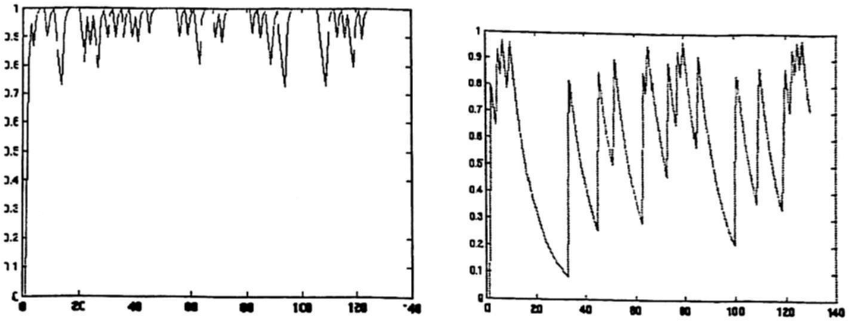


Fig. 3. Sample smoothed TSC (top) and participation functions for a segment of classroom discourse.

We now show the scatter plot between the log values of the two variables observed above except that we provide a time lag of 10 events. This scatter plot shows us the extent of the predictability of the logarithm of the balanced interaction coefficient and the logarithm of the smoothed occurrences of the TSC motif. As we can see, there is a region of correlation in which there seems to be strong.

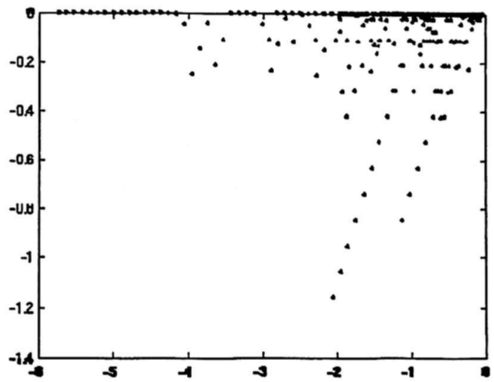


Fig. 4. Scatter plot of the log values of the delayed (time lagged) smoothed student participation index smoothed vs. smoothed TSC values.

6 Conclusions

In this paper we have looked at instructional mathematical discourse, and we have introduced a simple taxonomy to label classroom discourse events based on transactive and coherence discourse. We have discussed how classroom discourse events (utterances) can be classified into these categories using simple lexical feature classifiers, which can be easily extended to Logistic Regression/Maximum Entropy classifiers. We discussed to approaches to Motif discovery in biological sequences (MEM and GLAM) and introduced the utilization of these approaches to the sequences created by the interaction discourse labelers. Analysis of classroom data using both the Maximum Likelihood computation of finite mixtures and the Gapped local analysis using stochastic annealing revealed a common basic pattern: the TSC (which also generates TSYC and TSYCT). We demonstrated how based on motifs discovered using MEM and GLAM it is possible to use these motifs for other purposes, like prediction of interaction coefficient balance, or other predictive or analytic applications.

The main contribution of this paper is the introduction of a motif-based analysis of sequence of discourse labels and the application of motif discovering approaches used for DNA and protein motif discovery.

Future work should integrate motif discovery with other discourse analysis approaches including, for example, the modeling of discourse using dynamical systems, etc. Applications of the motif discovery approach includes: feature discovery for Tutoring systems, problem solving systems, meeting summarization as well as pedagogy-specific applications like teacher assessment, student assessment, professional development, portfolio creation and analysis, et cetera.

References

1. Hausmann, R. G. M. (2006). *Why do elaborative dialogs lead to effective problem solving and deep learning?* Poster presented at the 28th Annual Meeting of the Cognitive Science Conference, Vancouver, Canada.
2. Hausmann, R.G.M., Chi, M.T.H., & Roy, M. (2004) *Learning from collaborative problem solving: An analysis of three hypothesized mechanisms.* In K. D. Forbus, D. Gentner & T. Regier (Eds.), 26nd annual conference of the cognitive science society. Mahwah, NJ: Lawrence Erlbaum.
3. Stark, R., Mandl, H., Gruber, H., & Renkl, A. (2002). *Conditions and effects of example elaboration. Learning and Instruction, 12*(1), 39-60
4. Berkowitz, M. W., & Gibbs, J. C. (1983). *Measuring the developmental features of moral discussion*. Merrill-Palmer Quarterly, 29(4), 399-410.
5. Salomon, G. (1993). *No distribution without individuals' cognition: A dynamic interactional view.* In G. Salomon (Ed.), Distributed cognitions: Psychological and educational considerations. Cambridge: Cambridge University Press.
6. Hausmann, R.G.M., & Chi, M. T.H. (2003). *Co-construction: Mechanisms for peer-group learning.* Annual Meeting of the Midwestern Psychological Assn., 2003

7. Huerta, J. M., Stylianou, D. (2008) *The Teaching Buddy: Speech and Language Technologies for Assisting and Assessing Instructional Practice*, 7th European Conference on e-Learning, Cyprus 2008
8. Huerta J. M., (2008b) *Relative Rank Statistics for Dialog Analysis*, Conference on Empirical Methods in Natural Language Processing, 2008, Hawaii.
9. Blanton, M., Stylianou, D., & David, M. (2008), *Understanding Instructional Scaffolding in Classroom Discourse on Proof*, in The Learning and Teaching of Proof Across the Grades, eds. Stylianou, D., Blanton, M., & Knuth, E., Taylor Francis-Routledge
10. Marineau J., Wiemer-Hastings, P., Harter, D., Olde, B., Chipman, P., Karnavat, A., Pomeroy, V., Graesser, A. & TRG (2000), *Classification of speech acts in tutorial dialog*, Workshop on modeling human teaching tactics and strategies at the Intelligent Tutoring Systems
11. Mehan, H. (1985), *The Structure of Classroom Discourse: Handbook of Discourse Analysis*, Academic Press.
12. Truxaw, M. P. & DeFranco, T. C. (2007), *Mathematics in the making: Mapping verbal discourse in Pólya's "Let us teach guessing" lesson*, J. of Math. Behavior, 26(2)
13. Meyer D., Turner J. (2002) *Using Instructional Discourse Analysis to Study the Scaffolding of Student Self Regulation*, Educational Psychologist, 37(1), 17-25
14. Truxaw, M. P. and DeFranco, T. C. , (2004) *A Model for Examining the Nature and Role of Discourse in Middle Grades Mathematics Classes* Presented at the annual meeting of the North American Chapter of the International Group for the Psychology of Mathematics Education, Toronto, Ontario, Canada
15. Wu C., Lubensky D., Huerta J., Li X., and Kuo H-K (2003) *A Framework for Large Scalable Natural Language Call Routing Systems*, IEEE international conference on Natural Language Processing and Knowledge Engineering, Beijing, China
16. Higgins, D., Burstein, J., Marcu, D., & Gentile, C. (2004). *Evaluating multiple aspects of coherence in student essays*. Proc. of the annual meeting of HLT/NAACL
17. Higgins, D., & Burstein, J. (2006). *Sentence similarity measures for essay coherence*. Proceedings of the seventh international workshop on computational semantics (IWCS-7), Tilburg, The Netherlands.
18. Burstein, J., Marcu, D., & Knight, K. (2003). *Finding the WRITE stuff: Automatic identification of discourse structure in student essays*. In S. Harabagiu & F. Ciravegna (Eds.), IEEE Intelligent Systems: Special Issue on Advances in Natural Language Processing, 18(1).
19. Burstein, J., Chodorow, M., & Leacock, C. (2003). *Criterion: Online essay evaluation: An application for automated evaluation of student essays*. Proc. of the 15th annual conf. on innovative applications of artificial intelligence, Mexico.
20. Barzilay R., Lapata M. (2008) *Modeling Local Coherence: An Entity-based Approach*, Computational Linguistics, Vol.34 No. 1.
21. Stent A., *Rhetorical structure in dialog, in* Proceedings of the 2nd International Natural Language Generation Conference (INLG'2000).
22. Sporleder, Caroline and Alex Lascarides (2005) *Exploiting linguistic cues to classify rhetorical relations*, Proceedings of Recent Advances in Natural Language Processing (RANLP-05). Borovets, Bulgaria
23. Linden, K. V. and Martin, J. H. (1995). *Expressing rhetorical relations in instructional text: a case study of the purpose relation*. Comput. Linguist. 21, 1.
24. Taboada M., (2006) *Discourse markers as signals (or not) of rhetorical relations*, Journal of Pragmatics,Volume 38, Issue 4.
25. Miltsakaki E., Robaldo L., Lee A. and Joshi A., (2008) *Sense Annotation in the Penn Discourse Treebank*. Lecture Notes in Computer Science Publisher Springer. Computational Linguistics and Intelligent Text Processing

26. Moore J. D. and Wiemer-Hastings P., (2003) *Discourse in Computational Linguistics and Artificial Intelligence*. In A. G. Graesser, M. A. Gernbacher and S. R. Goldman, editors, Handbook of Discourse Processes, Lawrence Erlbaum.
27. Stolcke A., Coccaro N., Bates R., Taylor P., Van Ess-Dykema C., Ries K., Shriberg E., Jurafsky D., Martin R. and Meteer M. (2000) *Dialogue Act Modeling for Automatic Tagging and Recognition of Conversational Speech*. Comput. Ling., 3(26).
28. Grosz B., Joshi A., and Weinstein S., (1995) *Centering: A Framework for Modeling the Local Coherence of Discourse*." Comput. Ling., 2(21).
29. Grosz B. J., Pollack M., and Sidner C., (1989) *Computational Models of Discourse*' Foundations of Cognitive Science, Michael Posner, ed. MIT Press, Bradford Books, 1989.
30. Frith M.C., Saunders N.F., Kobe B., Bailey T., (2008) *Discovering sequence motifs with arbitrary insertions and deletions*, PLoS Comput. Biology, 4(5)
31. Bailey T. L., and Elkan C., (1994) *Fitting a mixture model by expectation maximization to discover motifs in biopolymers*, Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology, AAAI Press, Menlo Park, CA.
32. Bailey T. L., Williams N., Misleh C., and Li W. W., (2006) *MEME: discovering and analyzing DNA and protein sequence motifs*, Nucleic Acids Research, Vol. 34.
33. Mann W., and Thompson S., (1987). *Rhetorical structure theory: a theory of text organisation*. In L. Polanyi, editor, The Structure of Discourse. Ablex, Norwood, NJ.
34. Kruger, A.C. (1993). *Peer collaboration: Conflict, cooperation or both?* Social Development, 2(3), 165-182.

Ontology Oriented Computation of English Verbs Metaphorical Trait

Zili Chen, Jonathan J. Webster, Ian I. Chow and Tianyong Hao

Department of Chinese, Translation and Linguistics, City University of Hong Kong,
81 Tat Chee Avenue, Kowloon Tong, KLN, Hong Kong

Abstract. Research on metaphor has generally focused on exploring its context-dependent behavior and function. This current study aims to testify the postulate of English verb's innate trait of Metaphor Making potential. This paper intends to carry out an in-depth case study of a group of English core verbs using WordNet and SUMO ontology. In order to operationalize the assessment of an English verb's metaphor making potential, a refined algorithm has been developed, and a program made to realize the computation. At last, it is observed that higher frequency verbs generally possess greater metaphor making potential; while a verb's metaphor making potential on the other hand is also strongly influenced by its functional categories. As a preliminary context-free experiment with metaphor, this research foresees the possibility of providing an annotation schema for critical discourse analysis and a new parameter for scaling the difficulty level of reading comprehension of English texts.

Keywords: ontological computation, English verbs, MMP

1 Introduction and Previous Work

Metaphorical computation continues to remain a significant challenge to NLP. Recent researches of it mainly fall into two categories: rule-based approaches and statistical-based approaches. Up to now, some achievements have been attained, among which knowledge representation based methods are predominant [1]. These methods mainly employ knowledge representation based ontologies, such as The Suggested Upper Merged Ontology (SUMO), as their working mechanism. However, those researches are all limited to the study of metaphor's behavior and function in different contexts.

In line with Lakoff's view [2], "Metaphor allows us to understand one domain of experience in terms of another. This suggests that understanding takes place in terms of entire domains of experience and not in terms of isolated concepts", SUMO, an effort of the IEEE Standard Upper Ontology Working Group with the support of Teknowledge, contains terms chosen to cover all general domain concepts needed to represent world knowledge. Whereas Ahrens & Huang's research with SUMO and metaphor has focused on specific domain metaphors [3, 4], thus failing to make full use of SUMO's overall domain coverage.

Now that verb maintains the core for language processing, as believed by some

© A. Gelbukh (Ed.)
Advances in Computational Linguistics.
Research in Computing Science 41, 2009, pp. 135-142

Received 28/11/08
Accepted 11/12/08
Final version 04/02/09

linguists and philosophers, and previous work on metaphorical computation was focusing on noun metaphors, or verb's collocations, now the question is, would it be possible to look into the verb itself for its metaphorical property?

Lakoff also argues that verbs, as well as words of other classes, develop their new metaphorical meanings and usages from their root meanings through interaction with their surroundings [5]. But illustration and validation of this phenomenon depends on linguists' introspection and inference. We thus should expect the most efficient and objective way to investigate the Interactional Property and its underlying internal cross-domain alignment of prototypes is to examine how they are projected by the category-oriented SUMO hierarchy. Investigating this phenomenon using SUMO's hierarchy will provide a de facto computable ground for understanding verbs' self-contained metaphorical nature.

This paper conducts an in-depth case study of a selected group of English core verbs in the framework of WordNet and SUMO. In seeking ways to operationalize the assessment of English verbs' property of MMP, an algorithm is proposed based on the WordNet lexical representation and SUMO ontology. A pilot experiment is carried out with a small sample size of 50 most frequent English non-modal verbs of both imperfective and perfective obtained from BNC, TIME Magazine, CCAE (previously ANC) and Brown Corpus. A hypothesis based on Lakoff view [2] that metaphor is the result of "our constant interaction with our physical and cultural environments" has been set up as well to test whether higher frequency verbs show greater MMP. As a study both theory and application-oriented, this paper also shows that an ontology-based approach is more objective than an intuition-based approach in generating insights into verbs' metaphorical property. As a pilot context-free study with metaphor, this research foresees the possibility of providing an annotation schema for critical discourse analysis and a new parameter for scaling the difficulty level of reading comprehension of English texts.

1.1 Metaphor, Conceptual Metaphor and Metaphorical Computation

Metaphor study has gone through three major stages from Aristotle's Comparison and Substitute View, through Richard and Black's Interaction View to finally the current Conceptual View. Meanwhile, Chinese linguists have for the most part limited their investigation of metaphor to its rhetorical and psychological properties.

G. Lakoff and M. Johnson [2] set out to develop a new theory called Conceptual Metaphor (CM), in which they argue that human thought processes and conceptual system are metaphorically defined and structured; and "the essence of metaphor is understanding and experiencing one kind of thing in terms of another." Differing from the objectivist's view of inherent property, CM's conceptual system is the product of how we interact with our physical and cultural environments. Furthering the definition of a concept and changing its range of applicability is possible because metaphor-driven categorization and recategorization render the open-endedness of concept. Thus we should expect the most efficient way to investigate those Interactional Properties and their underlying internal cross-domain alignment of prototypes is to examine how they are projected by the category-oriented SUMO hierarchy.

Recent researches in metaphorical computation mainly fall into two categories: rule-based approaches and statistical-based approaches. The former stems from conventional theories of metaphor in linguistics, philosophy and psychology, including specifically metaphor semantics, possible worlds semantics, and knowledge representation. And the latter dwells on corpus linguistics and employs statistical based techniques. Those papers are all limited to the study of metaphor's behavior and function in different contexts [1].

2 Research Justification and Design

In terms of the above consideration, the intended experiment will look into a selected group of English core verb's self-contained metaphorical traits through mapping their senses in WordNet to SUMO's domain-aligned hierarchy.

Lakoff argues that verbs, as well as words of other classes, develop their new metaphorical meanings and usages from their root meanings through interaction with their surroundings [5]. But illustration and validation of this phenomenon depends on linguists' introspection and inference. Investigating this phenomenon using SUMO's hierarchy will provide a de facto computable ground for understanding verbs' self-contained metaphorical nature. Moreover, the centrality of verbs for language progression and processing has often been emphasized [6].

SUMO has more than 1000 terms, 4000 axioms and 750 rules. A verb in WordNet has various senses all of which are located in different levels of concepts under Entity in SUMO. Verbs differ from each other in that each verb's senses' depth to the root differs from that of other verbs [7, 8, 9]. Calculation of these differences resembles computation of words' semantic distance, semantic similarity and semantic relatedness. There are currently dozens of calculators to measure words' semantic distance/similarity/relatedness, most of which rest on WordNet. Representative measures are Hirst-St-Onge [10], Leacock-Chodorow [11], Wu and Palmer [12], Jiang-Conrath [13], Lin [12], and Gloss Vector (pairwise) [13]. They assign different weights on words' width, depth, information content, etc., thus output different semantic distances. All those measures calculate the semantic distance by computing the shortest edges or IC between two words. Our tentative measurement varies from the above in that instead of directly measuring the shortest paths between two words, this method determines a verb's metaphorical width by adding up its senses' overall relative distance, which by turns is calculated by tracing and measuring each closest concept pair's Lowest Common Consumer's location in SUMO hierarchy back to its root. Comparing with methods of information content, the major difference is that it measures the information content above the LCCs, not below the LCCs.

3 Research Methodology

3.1 Identification of the Selected List of English Core Verbs and Mapping Their WordNet Senses to SUMO Concepts

A simple method shown to be very useful to delimit a group of core verbs is frequency ranking (e.g. the normal practice is the 10, 20, 50, or 100 most frequent verbs) within a particular word class; frequency ranking of general purpose corpus will be considered for trimming the list of core verbs. Specifically, the British National Corpus (BNC), TIME Magazine, Corpus of Contemporary American English and the book "Frequency Analysis of English Usage" based on the earlier Brown Corpus are consulted for English verbs' general purpose frequency ranking. We filtered and finalized a list of 50 most frequent verbs for our pilot study.

Adam Peace et al have already mapped a word's WordNet senses to its SUMO corresponding concepts [16].

3.2 Algorithmic Consideration

Calculate a Verb's MMP Value. A verb's metaphor making potential (MMP) is measured in terms of the verb's WordNet Senses' locations in the SUMO ontology, which are mapped onto SUMO's hierarchical concepts. The verb's MMP in the SUMO hierarchy is further determined by its' senses' respective Depths and Overall Relative Width (ORWD). A verb's MMP is calculated and partly normalized by the formula below,

$$MMP(Verb) = \sum_{i=1}^{n} \frac{DP(S_i)}{Max_{DP(S)}} \cdot ORWD(Verb).$$

Where n is a verb's total number of WordNet senses mapped to SUMO's hierarchical concept, $DP(S_i)$ is the depth of i-th sense in SUMO hierarchy, $Max_{DP(S)}$ is the maximum depth of a sense in SUMO hierarchical concept, $ORWD(Verb)$ is the verb's WordNet senses' overall relative width in SUMO hierarchy.

Calculate the Depth of a Sense in SUMO Ontology. The depth of a verb's WordNet sense is defined as the minimum edge count of paths in SUMO from the root to the sense, i.e. from the Entity to the concept that the sense subsumes or equates, including the sense when subsuming or not including the sense when equating.

We define the depth of the sense i as $DP(S_i)$ in SUMO ontology,

$$DP(S_i) = Min(Len_e(Path_l) \mid 1 \le l < TotalPaths\ )$$

where *TotalPaths* is the total number of paths from this sense to the Entity, $Len_e(Path_l)$

is the edge count of $Path_l$ of this sense i, including the sense edge when it subsumes the SUMO concept or not including the sense edge when it equates the SUMO concept.

Calculate a Verb's Overall Relative Width in SUMO Ontology. The Overall Relative Width of a verb's senses is a new term coined in this paper to describe another inherent metaphorical property of a verb - Metaphorical Width, namely, the horizontal reciprocal distance of all concepts that a verb's senses subsume or equate. Unlike the more static and fixed methods for measuring semantic distance such as Hirst-St-Onge etc., this notion of metaphorical width is a dynamic and relative one. Following Lakoff [2], "Metaphor allows us to understand one domain of experience in terms of another. This suggests that understanding takes place in terms of entire domains of experience and not in terms of isolated concepts", this research postulates that a verb's metaphorical width must be assessed by viewing all concerned SUMO concepts simultaneously; any isolated treatment of concepts is theoretically and operationally partial and will fail to obtain the overall assessment. Moreover, since metaphorization is primarily about migration of a concept to any successive potential concept, the metaphorical width calculation shall consider the de facto displacement both between two interrelated concepts and among all interrelated concepts. In other words, this paper posits that it is the shifting between those interrelated concepts, instead of the static concepts themselves that works to delineate a word's metaphorical property. A shift from a concept to another generates a certain quantity of metaphorical potential. So what we do is to find a way to quantify how much metaphorical potential those shifts generate. The approach for counting a verb's metaphorical width sets off to compute all possible paths of the verb's all senses to spot the shortest one. Suppose a verb has a sense set S, which contains $\{S_1 \ldots S_n\}$. Each sense is mapped to corresponding SUMO concept in the verb's senses' SUMO concept set C, which contains $\{C_1, Ci, Cj \ldots C_k\}$ ($k \leq n$). A verb's metaphorical width is defined as the minimum overall relative distance in SUMO from C_1 through C_i, C_j to C_k. A verb's overall relative width (*ORWD(Verb)*) can be obtained by formulas below,

$$ORWD(Verb) = \sum_{i=1}^{k} RWD(C_i, C_j)$$

$$RWD(C_i, C_j) = \frac{1}{Min(len_n(Paths_{LCS(C_i, C_j)}))}$$

where *Cj* is the closest concept to any concept *Ci* of *C* in SUMO, RWD(*Ci, Cj*) is the relative width between *Ci* and *Cj*, *LCS*(*Ci, Cj*) is the Lowest Common Subsumer of *Ci* and *Cj*, and Len_n is the number of nodes count from *LCS*(*Ci, Cj*) to Entity. Note that we start from any concept *Ci*, to its closest concept *Cj*, then move on to *Cj*'s closest concept excluding *Ci*, and the like, till the last concept C_k; and since the whole metaphorical shifting process stops at C_k, C_k and its closest preceding concept thus forms the last interrelated pair which generates relative width.

4 Results and Discussion

Before the experiment, what has been anticipated is that the higher frequent verbs would possess the more metaphorical potential, which is based on the belief that a more utilized verb is involved in more interactions, thus tends to incur more metaphorical usages [5]. Result of this preliminary study shows that the hypothesis is generally true as shown by the trend line in Figure 1.

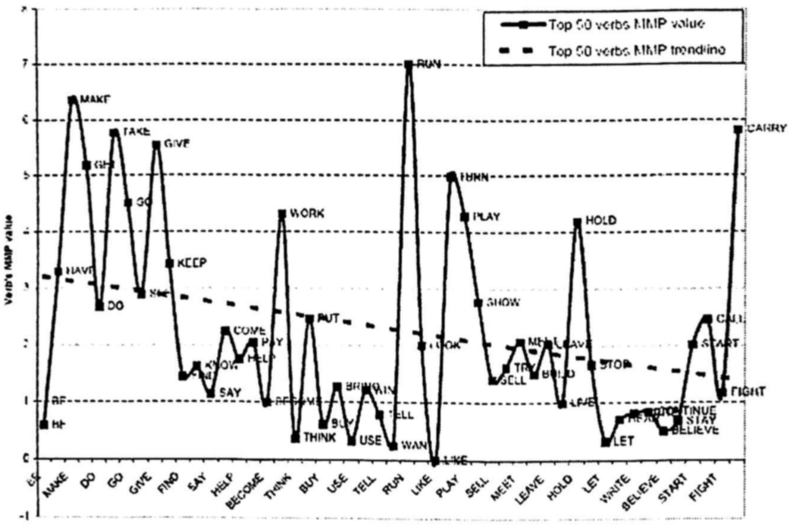


Fig. 1. Top most frequent verbs MMP distribution

Mann-Kendall method [17] is used to further test whether verbs' MMP has a significant downward trend in correlation with verbs' frequency ranking. Kendall test is a nonparametric test rule and insensitive to extreme value and thus fits the feature of the experimental data (MMP(Verb1), .., MMP(Verb50)) as a sample of independent and non-normally distributed random variables. Its null hypothesis H_0 is that there is no trend in the top 50 verbs' Metaphor Making Potential *MMP(Verb)*. The Kendall test rejected the H_0 by showing that there is a significant downward trend at the 0.05 level for the top 50 verb's MMP.

Moreover, we also observed some interesting phenomenon. Verbs like *give, take, make, get, run, turn, hold, carry, etc.*, which are positioned in the middle or bottom based on frequency ranking are at the top in terms of their MMP value; while verbs *be, do, say, think, want, etc.*, which are ranked at the top or middle based on their frequency ranking now at the bottom in terms of their MMP ranking. Further investigation reveals that those verbs ranking higher in terms of metaphorical potential fall into the verb categories of Possession, Production and Motion; while those ranking lower in metaphorical potential (with the exception of *say*) all fall into the verb category of General Dynamic and Cognition [18]. This finding suggests that verbs' MMP trait is closely linked to verbs' functional categories.

The small size of the 50 words samples analyzed however precludes the possibility of hastily drawing any generalizations. Instead, we anticipate that such should be possible after conducting a future study into verbs' metaphorical traits based on a large sample size analyzed using SUMO.

5 Summary and Future Work

The metaphor making potential in language is another breakthrough finding of a word's build-in universal trait in terms of metaphor. On the one hand, it depends on a word's ability to cross-domain attribute, while on the other hand it makes it feasible to understand and experience one kind of thing in terms of another. Expanding the definition of a concept and broadening its range of applicability is possible because every word has its metaphor making potential which renders the open-endedness of concept. Related to that, SUMO illustrates a full blown hierarchy of terms chosen to cover all general domain concepts needed to represent world knowledge. Thus SUMO ontology is used to project a verb's MMP.

This study is both theory- and application-oriented. A refined method is proposed to assess a word's intrinsic metaphorical property. And SUMO as an ontology benchmark is validated as well. We have observed that higher frequency verbs generally possess greater metaphor making potential; while the verb's MMP on the other hand is also strongly influenced by its functional category. As a preliminary context-free experiment with metaphor, this research foresees the possibility of providing an annotation schema for critical discourse analysis and a new parameter for scaling the difficulty level of reading comprehension of English texts.

One of the future tasks is to expand the sample size of core English verbs to produce a stronger validation; another is to apply this method to other classes of words to generate the contour of a word's trait of metaphor making potential. We also hope that its application to discourse analysis and textual annotation will also be explored.

References

1. Zhou, C.L., Yang Y., Huang X.X.: 2007. Computational Mechanisms for Metaphor in Languages: A Survey. Journal of Computer Science and Technology. 22(2), 308-319 (2007)
2. Lakoff, G., Johnson, M.: Metaphors We Live by. The University of Chicago Press, Chicago (1980)
3. Ahrens, k.: When Love Is Not Digested: Underlying Reasons for Source to Target Domain Pairing in the Contemporary Theory of Metaphor. In: Proc. 1st Cognitive Linguistics Conference, pp. 273-302. Taipei (2002)
4. Ahrens, K., Huang, C.R., Chung, S.F.: Conceptual Metaphors: Ontology-Based Representation and Corpora Driven Mapping Principles. In: Proc. ACL Workshop on Lexicon and Figurative Language, pp. 35-41. Sapporo, Japan (2003)
5. Lakoff, G.: The Contemporary Theory of Metaphor. In: Ortony A. (ed.) Metaphor and Thought, 2nd Edition, pp. 202--251. Cambridge University Press, Cambridge (1993)
6. Viberg, A.: Crosslinguistic Perspectives on Lexical Organization and Lexical Progression. In Hyltenstam, K., Viberg, A. (eds.) Progression & Regression in Language: Sociocultural,

Neuropsychological, & Linguistic Perspectives, pp. 340-385. Cambridge University Press, Cambridge (1993)

7. Niles, I., Pease, A.; Towards a Standard Upper Ontology. In: Welty C., Smith B. (eds.) Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001). October 17-19, Ogunquit, Maine (2001)
8. Pease, A., Niles, I., Li, J.: The Suggested Upper Merged Ontology: A Large Ontology for the Semantic Web and its Applications. In: Working Notes of the AAAI-2002 Workshop on Ontologies and the Semantic Web, July 28-August 1, Edmonton, Canada (2002)
9. Chow, I.C., Webster, J.J.: Mapping FrameNet and SUMO with WordNet Verb: Statistical Distribution of Lexical-Ontological Realization. In: Fifth Mexican International Conference on Artificial Intelligence (MICAI'06), pp. 262-268. (2006)
10. 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. MIT Press, Cambridge MA (1998)
11. Leacock, C., Chodorow, M.: Combining Local Context and WordNet Similarity for Word Sense Identification. In: Fellbaum, C. (ed.) WordNet: An Electronic Lexical Database, pp. 265–283. MIT Press, Cambridge MA (1998)
12. Wu, Z., Palmer, M.: Verb Semantics and Lexical Selection. In: 32nd Annual Meeting of the Association for Computational Linguistics, pp. 133–138. Las Cruces, New Mexico (1994)
13. Jiang, J., Conrath, D.: Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. In: Proceedings on International Conference on Research in Computational Linguistics, pp. 19–33. Taiwan (1997)
14. Lin, D.: An Information-Theoretic definition of Similarity. In: Proceedings of the International Conference on Machine Learning. Madison (1998)
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, pp. 241–257. February, Mexico City (2003)
16. WordNet & SUMO mapping. http://sigmakee.cvs.sourceforge.net/viewvc/sigmakee/KBs/WordNetMappings/
17. Kendall, M.G.: Rank Correlation Methods, 3rd edition. Hafner, New York (1962)
18. Levin, B.: English Verb Class and Alternations: A Preliminary Investigation. University of Chicago Press, Chicago (1993)

Automatic Formal Verification of Conceptual Model Documentation by Means of Self-organizing Map

Algirdas Laukaitis and Olegas Vasilecas

Vilnius Gediminas Technical University , Sauletekio al. 11,
LT-10223 Vilnius-40, Lithuania
{algirdas.laukaitis,olegas}@fm.vgtu.lt

Abstract. By using background knowledge of the general and specific domains and by processing new natural language corpus experts are able to produce a conceptual model for some specific domain. In this paper we present a model that tries to capture some aspects of this conceptual modeling process. This model is functionally organized into two information processing streams: one reflects the process of formal concept lattice generation from domain conceptual model, and the another one reflects the process of formal concept lattice generation from the domain documentation. It is expected that similarity between those concept lattices reflects similarity between documentation and conceptual model.In addition to this process of documentation formal verification the set of natural language processing artifacts are created. Those artifacts then can be used for the development of information systems natural language interfaces. To demonstrate it, an experiment for the concepts identification form natural language queries is provided at the end of this paper.

Key words: Information systems engineering, formal concept analysis, IS documents self-organization, natural language processing.

1 Introduction

Software engineers spend hours in defining information systems (IS) requirements and finding common ground of understanding. The overwhelming majority of IS requirements are written in natural language supplemented with conceptual model and other semi-formal UML diagrams. The bridge in the form of semantic indexes between documents and conceptual model can be useful for more effective communication and model management. Then, an integration of the natural language processing (NLP) into information system documentation process is an important factor in meeting challenges for methods of modern software engineering.

Reusing natural language IS requirement specifications and compiling them into formal statements has been an old challenge [1], [14]. Kevin Ryan claimed that NLP is not mature enough to be used in requirements engineering [13] and

© *A. Gelbukh (Ed.)*
Advances in Computational Linguistics.
Research in Computing Science 41, 2009, pp. 143-154

Received 10/11/08
Accepted 26/01/09
Final version 06/02/09

our research express that as well. Nevertheless, we hope that the current paper will suggest some promising findings towards this challenging task.

The idea that we want to investigate in this paper consist from comparison of two concept lattices received from different information processing streams: 1) the first one is information processed by the human expert in the process of conceptual modeling, 2) and the second one is NLP of the domain documents. We suggest the formal concept analysis (FCA) [3] as a framework to compare results from those two information processing streams. We assume that an expert can define the domain object-attribute matrix. From that matrix FCA produces concept lattice which can be interpreted as the domain's conceptual model. For the documents set the formal concept analysis, if applied directly to words-document matrix, produces lattice that is far too big for any comprehensible analysis. Then, we suggest the self-organizing map (SOM) [10] as a tool for preprocessing and the problem dimensionality reduction. Finally, all presented ideas and methodological inference is tested with the IBM Information FrameWork (IFW) [7], which is a comprehensive set of banking specific business models from IBM corporation. For our research we have chosen the set of models under the name *Banking Data Warehouse.*

Then, solution for the stated problem organize the rest of the paper as follows. First, we present the general framework of the automated model generation system from the documents set. Next, we present IBM's IFW solution and the model which we used in our experiments. At this section we present FCA as the formal technique to analyze the *object:attribute* sets. In the Section 4, we present the architectural solution of the natural language processing system that has been used in this paper. SOM of the conceptual model is introduced in chapter 5. Finally to prove the soundness of the proposed method we provide a numerical experiment in which the ability of the system to identify concepts from users utterance is tested. The IBM Voice Toolkit for WebSphere [8] (approach based on statistical machine learning) solution is compared with the system suggested in this paper.

2 General Framework of the Solution

Conceptual models offer an abstracted view on certain characteristics of the domain. They are used for different purposes, such as a communication instrument between users and developers, for managing and understanding the complexity within the application domain, etc. The presence of tools and methodology that supports integration of the documents and communication utterance into conceptual model development is crucial for the successful IS development. In this paper for such tool we suggest a framework that is presented in the Figure 1. First, the corpus is created from the model's concept descriptions and in the figure it is named as the domain descriptions. The goal of the framework is to derive two concept lattices shown in the right side of the Figure 1. One concept lattice is generated from conceptual model which was created by the domain analyst. In the next section the process is described in details. The second concept lattice is generated automatically from domain descriptions and we say that we have *"good"* descriptions if those lattices resembles each other.

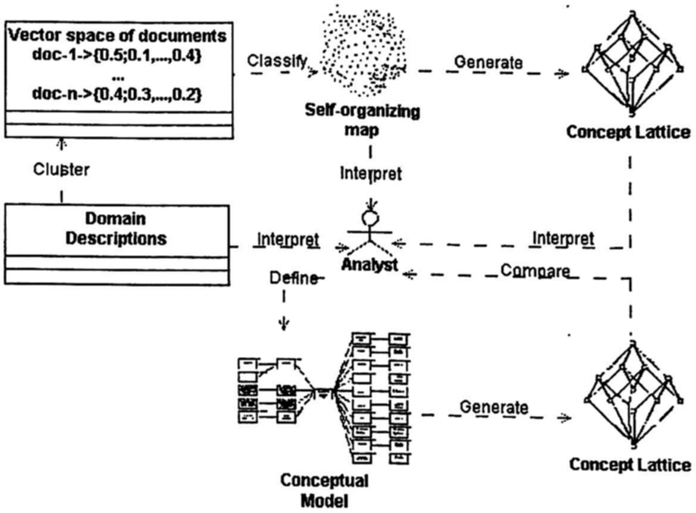


Fig. 1. Process of integration: Conceptual modeling, textual descriptions clusters detection and interpretation by use of FCA.

At the core of the second lattice generation is self-organizing map which is used for cluster analysis. In this paper we suggest the use of SOM to classify IS documentation and IS utterance on a supervised and an unsupervised basis. SOM has been extensively studied in the field of textual analysis. Such projects like WEBSOM [9], [11] have shown that the SOM algorithm can organize very large text collections and that it is suitable for visualization and intuitive exploration of the documents collection. The experiments with the Reuters corpus (a popular benchmark for text classification) have been investigated in the paper [6] and there was presented evidence that SOM can outperform other alternatives.

Because our goal is to derive formal lattice from the domain documents, which then can be compared with formal lattice from conceptual model, we suggest transformation schema that use FCA and transforms SOM to concepts lattice. It is important to notice that when directly applied to the big data set of textual information, FCA gives overwhelming lattice. This argument motivates integration of the FCA and other text clustering techniques. In that sense our work bears some resemblance with the work of Hotho et.al. [5]. They used BiSec-kk-Means algorithm for text clustering and then FCA was applied to explain relationships between clusters. Authors of the paper have shown the usability of such approach in explaining the relationships between clusters of the Reuters-21578 text collection.

3 Concept Lattice Representation of the Conceptual Model

The problem with data centric enterprise wide models is that it is difficult to achieve their use by all employees in the company. Their abstract and generic concepts are unfamiliar to both business users and IS professionals, and remote from their local organizational contexts. Natural language processing can be used to solve mentioned problems. But, before applying the NLP techniques for the model and its documentation management, we must have some formal method to deal with the set of {*classes, object and attributes*}. In this section we introduce the formal concept analysis as the method for automatically building the hierarchical structure of concepts (or classes) from the {*object:attribute*} set.

As an example, consider Figure 2 (left side) we can see an excerpt of the IBM IFW financial services data model (FSDM) [7]. The IBM financial services data model is shown to consist of a high level strategic classification of domain classes integrated with particular business solutions (e.g. Credit Risk Analysis) and logical and physical data entity-relationship (ER) models.

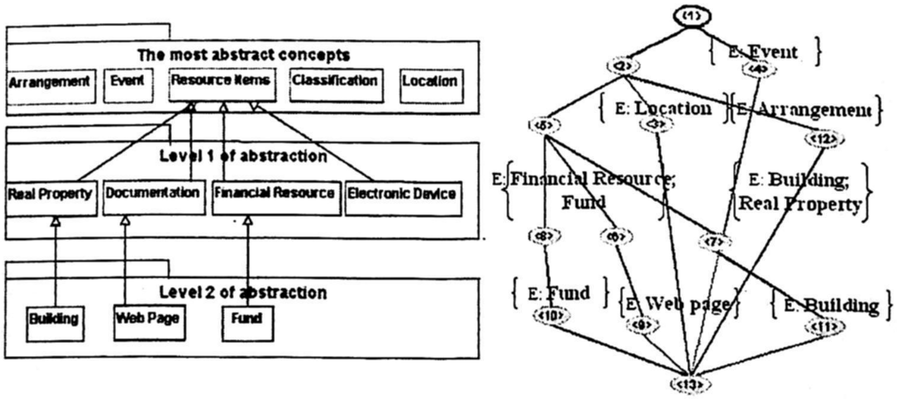


Fig. 2. Left side: A small extract from the financial services conceptual model. Right side: CL from this conceptual model. (We see that FCA depicts the structure from the conceptual model.)

Concept lattice of this model have been produced by FCA with Galicia software [16] and is shown in the right side of the Figure 2. As we can see it is consistent with the original model. It replicates underlying structure of conceptual model originally produced by an expert team and in addition suggests one formal concept that aggregates *Arrangement* and *Resource Item*: the two top concepts from the original model. As we can see from this simple example, FCA is used to represent underling data in the hierarchical form of the concepts. Due to its comprehensive form in visualising underlaying hierarchical structure of the data and rigorous mathematical formalism FCA grown up to mature theory for data analysis from its introduction in the 1980s [3].

In order to better understand the process of concept lattice generation from conceptual model, we can return to the Figure 2. As we can see from the figure, there are 12 concepts. As the first step, we instantiate the set of those concepts. The set of instantiated objects we name as G. Let M be the set of all attributes that characterise those objects i.e. an attribute is includes into the set M if it is an attribute for at least one object from the set G. In our example we have 137 attributes (the whole model has more than 1000 objects and more than 4000 attributes). We identify the index I as a binary relationship between two sets G and M i.e. $I \subseteq G \times M$. In our example the index I will mark that, eg., an attribute "interest rate" belongs to an object "Arrangement" and that it does not belong to an object "Event".

The FCA algorithms starts with the definition of the triple $\mathbb{K} := (G, M, I)$ which is called a formal context. Next, the subsets $A \subseteq G$ and $B \subseteq M$ are defined as follows:

$$A^{'} := \{m \in M | (g, m) \in I \texttt{ for all } g \in G\},$$

$$B^{'} := \{g \in G | (g, m) \in I \texttt{ for all } m \in B\}.$$

Then a formal concept of a formal context (G, M, I) is defined as a pair (A, B) with $A \subseteq G$, $B \subseteq M$, $A^{'} = B$ and $B^{'} = A$. The sets A and B are called extend and intend of the formal concept (A, B) . The set of all formal concepts $\mathfrak{B}(\mathbb{K})$ of a context (G, M, I) together with the partial order $(A_1, B_1) \leqslant (A_2, B_2) :\Leftrightarrow A_1 \subseteq A_2$ is called the concept lattice of context (G, M, I) .

In the Figure 2 the FCA algorithm *Incremental Lattice Builder* generated 11 formal concepts. In the lattice diagram, the name of an object g is attached to the circle and represents the smallest concept with g in its extent. The name of an attribute m is always attached to the circle representing the largest concept with m in its intent. In the lattice diagram an object g has an attribute m if and only if there is an ascending path from the circle labeled by g to the circle labeled by m. The extent of the formal concept includes all objects whose labels are below in the hierarchy, and the intent includes all attributes attached to the concepts above. For example the concept 7 has $\{$*Building; Real Property*$\}$ as extend (the label *E:* in the diagram), and $\{$*Postal Address; Environmental Problem Type;Owner;... etc.*$\}$ as intent (due to the huge number of attributes they are not shown in the figure).

4 Vector Space Representation of the Conceptual Model

The vector space model (VSM) for documents transformation to the vectors is a well-known representation approach that transforms a document to a weight vector. The most naive way to construct vector is based on the bag-of-words approach, which ignores the ordering of words within the sentence, ignores their semantic relationships and uses basic occurrence information [15]. On the other hand, if we take the bag-of-words approach, the dimension of the vector space is based on the total number of words in the data set and we are faced with problems of big dimensionality reduction.In this paper, the process of dimensionality reduction and noise filtering is depicted in Figure 4. All presented processes are described in details below.

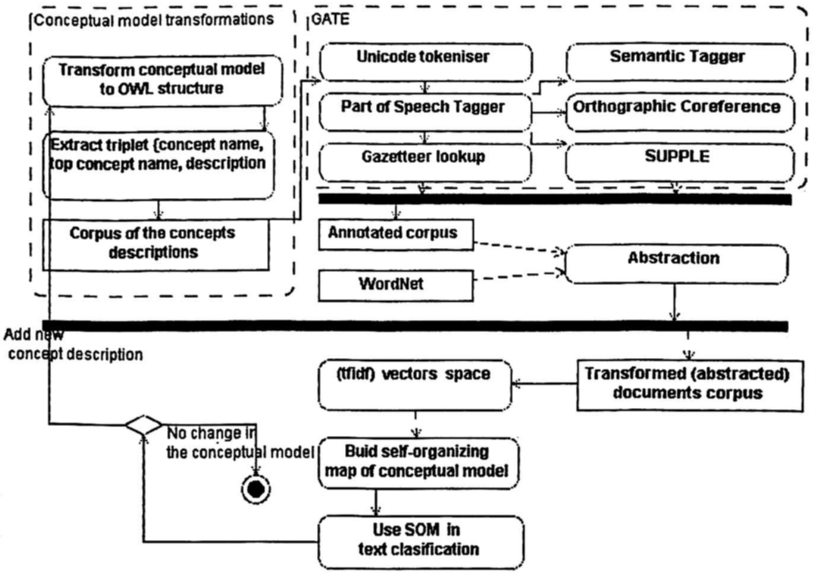

Conceptual model transformations
GATE
Transform conceptual model to OWL structure
Extract triplet (concept name, top concept name, description
Corpus of the concepts descriptions
Unicode tokeniser
Part of Speech Tagger
Gazetteer lookup
Semantic Tagger
Orthographic Coreference
SUPPLE
Annotated corpus
WordNet
Abstraction
Add new concept description
(tfidf) vectors space
Transformed (abstracted) documents corpus
No change in the conceptual model
Buid self-organizing map of conceptual model
Use SOM in text clasification

Fig. 3. The processes of dimensionality reduction and the conceptual model SOM design.

1. *Transform conceptual model.* As the first in this process, we transform conceptual model to the Web Ontology Language (OWL) structure. The motivation behind this step is that OWL is a standard for expressing ontologies in the Semantic Web (W3C recommendation).

2. *Extract triplet* is a process of data preparation. The triplet: concept name, the most abstract parent-concept name, and textual description of the concept are extracted. We received 1256 documents in the corpus where each document describes one concept. For example, the concept *Employee* has the following entry in the corpus: {***Concept***-*Employee;* ***Top parent concept*** - *Involved Party ;* ***Description*** - *An Employee is an Individual who is currently, potentially or previously employed by an Organization, commonly the Financial Institution itself...* }. It is important to mention that there was only 9 *top parent* concepts:

(*involved party, products, arrangement, event, location, resource items, condition, classification, business*).

3. *GATE - Natural Language Processing Engine* is a well-established infrastructure for customization and development of NLP components [2]. It is a robust and scalable infrastructure for NLP and allows users to use various modules of NLP as the plugging. We briefly describe the components used for the concepts vector space construction. The *Unicode tokeniser* splits the text into simple tokens. The *tagger* produces a part-of-speech tag as an annotation on each word or symbol. The *gazetteer* further reduces dimensionality of the document corpus. *Semantic tagger* - provides finite state transduction over annotations based on regular expressions. *Orthographic Coreference* produces identity relations between named entities found by the semantic tagger. *SUPPLE* is a bottom-up parser that constructs syntax trees and logical forms for English sentences.

4. *Abstraction.* The basic idea of the abstraction process is to replace the terms by more abstract concepts as defined in a given thesaurus, in order to capture similarities at various levels of generalization. For this purpose we used WordNet [12] and annotated GATE corpus as the background knowledge base. WordNet consists of so-called synsets, together with a hypernym/hyponym hierarchy [4]. To modify the word vector representations, all nouns have been replaced by WordNet corresponding concept('synset'). WordNet '*most common*' synset was used for a disambiguation.

5. *Vectors space.* In our experiments we used vector space of the term vectors weighted by *tfidf* (term frequency inverse document frequency)[15], which is defined as follows:

$$tfidf(c,t) = tf(c,t) \times \log \frac{|C|}{|C_t|},$$

where $tf(c,t)$ is the frequency of the term t in concept description c, and C is total number of terms and C_t is the number of concept descriptions. $tfidf(c,t)$ weighs the frequency of a term in a concept description with a factor that discounts its importance when it appears in almost all concept descriptions.

5 Self-organizing Map of the IS Conceptual Model

Neurally inspired systems also known as connectionist approach replace the use of symbols in problem solving by using simple arithmetic units through the process of adaptation. The winner-take-all algorithms also known as self-organizing network selects the single node in a layer of nodes that responds most strongly to the input pattern. In the past decade, SOM have been extensively studied in the area of text clustering. It consists of a regular grid of map units. Each output unit i is represented by prototype vector $m_i = [m_{i1}...m_{id}]$, where d is input vector dimension. Input units take the input in terms of a feature vector and propagate the input onto the output units. The number of neurons and topological structure of the grid determines the accuracy and generalization capabilities of the SOM.

During learning the unit with the highest activation, i.e. the best matching unit, with respect to a randomly selected input vector is adapted in a way that it will exhibit even higher activation with respect to this input in future. Additionally, the units in the neighborhood of the best matching unit are also adapted to exhibit higher activation with respect to the given input.

As a result of training with our financial conceptual model corpora we obtain a map which is shown in the Figure 4. This map has been trained for 100,000 learning iterations with learning rate set to 0.5 initially. The learning rate decreased gradually to 0 during the learning iterations.

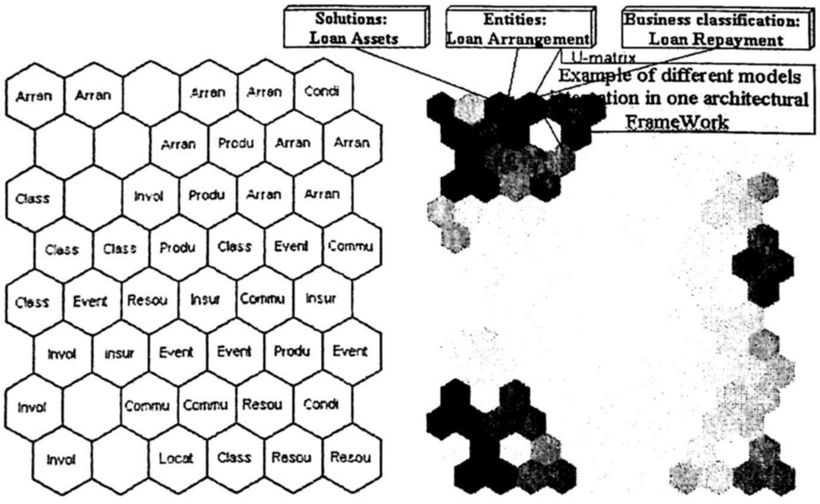


Fig. 4. SOM for the conceptual model. Labels: invol, accou, locat, arran, event, produ, resou, condi represents concepts: involved party, accounting, location, event, product, resource, condition.

We have expected that if the conceptual model vector space has some clusters that resembles conceptual model itself, then we can expect that the model will be easier understood compared with the model of more random structure. On a closer look at the map we can find regions containing semantically related concepts. For example, the right side top of the final map represents a cluster of concepts "Arrangement" and bottom right side "Resource items". Such map can be used as scarificator for any textual input. It always will assign a name of some concept from the conceptual model. Nevertheless, such map, as in the Figure 4, is difficult to use as an engineering tool. A hierarchical structure is more convenient for representation of the underlying structure of the concept vector space.

Figure 5 shows the concepts lattice computed from SOM shown in the figure 4. We obtain a list of 23 formal concepts. Each of them groups several neurons from SOM. We can find the grouping similarity of the neurons that are locate in the neighborhood of each other. On the other hand some concepts makes a group of neurons that are at some distance form each other. The basic idea of this step is that we received a closed loop in the business knowledge engineering by artificial intelligent agent. The agent classifies all textual information with the SOM technique and then using FCA it builds hierarchical knowledge bases. For the details on how to apply FCA to the cluster analysis (SOM in our case) we refer to the paper [5]. The paper describes an algorithm which has been used in our research.

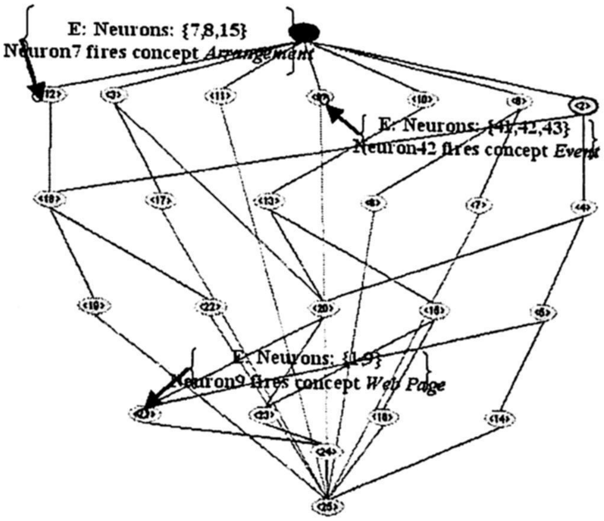


Fig. 5. Concepts lattice that has been received from the SOM presented in the Figure 4.

6 Experiment

Automatically generated concept lattice of document corpora is a useful tool for visual inspection of underlying vector space. Nevertheless, we would like to have a more rigorous evaluation of the lattice capability to depict conceptual model structure. For that purpose a classification accuracy (CA) measure has been used. CA simply counts the minority of concepts at any grid point and presents the count as classification error. For example, after the training, each map unit (and lattice node) has a label assigned by highest number of concepts mapped to this unit (Figure 4). As we can see in the Figure 4, the top left neuron mapped 4 concepts with the label *arrangement* and 2 with label *event*. Thus, classification accuracy for this neuron will be 66 %. For the whole concept map we received $CA = 39.27\%$. It is not very high classification accuracy, but it can be used as a benchmark to compare other methods.

The framework that we presented in the previous sections generates document corpora lattice which can be visually compared with lattice generated from conceptual model. But as mentioned in the introduction, one of the objectives in this research was to find the techniques and tools of modeling that, in addition to the visualization, generate some artifacts for natural language interfaces. In this paper we suggest to reuse SOM as classification component. Each time the sentence is presented to the SOM component we have one activated neuron which is associated with one concept from the conceptual model. Additionally, we have the set of formal concepts associated with the activated neuron. Both, the label from activated neuron and the set of formal concepts can be used by some formal language generation engine (i.e. structured query language (SQL) sentence generator for querying databases).

The following experiment has been conducted to test this approuch. IBM WebSphere Voice Server NLU toolbox [8], which is a part of the IBM WebSphere software platform have been chosen as the competitive solution to the one suggested in this paper. SOM of the conceptual model and CL have been used as an alternative to the IBM WebSphere Voice Server NLU solution. We have taken the black box approach for both solutions: put the training data, compile and test the system response for the new data set. The data set of 1058 pairs *textual description:concept name* mentioned above were constructed to train the IBM NLU model. The same set has been used to get SOM.

Then a group consisting of 9 students has been instructed about the database model. They have the task to present for the system 20 questions about information related to the concept "Involved`Party". For example one of the questions was: "*How many customers we have in our system*?" We scored the answers from the system as correct if it identified the correct concept "Involved Party".

Table 1. Concept identification comparison between IBM NLU toolbox and SOM of database conceptual model.

	CN=9	CN=50	CN=200	CN=400	CN=500
IBM NLU	36.82	17.26	14.82	11.15	8.22
SOM	46.73	30.70	27.11	20.53	18.83

At the beginning only 9 top concepts were considered i.e. all 1058 documents have been labeled with the most abstract concept names from the conceptual model. For example documents that described concepts "Loan" and "Deposit" are labeled with the concept name "Arrangement" because concepts "Loan" and "Deposit" are subtypes of the concept "Arrangement".

Next we increased the number of concept names that we put into the model up to 50. For example, documents that described concepts "Loan" and "Deposit" have been labeled with "Loan" and "Deposit" names. Then, number of concept names has been increased up to 200, 400 and finally 500. Table 1 shows the results of the experiment. Column names show the number of concepts. The row named *IBM NLU* represents results for the IBM WebSphere Voice Server NLU toolbox. The row named *SOM* represents results for the SOM of the conceptual model that has been constructed with the method described in this paper. For the classification error, the proportion of the correctly identified concepts has been used.

As we can see, the performance of the IBM system was similar to the SOM response. For all cases i.e. IBM and SOM the performance decreased when the number of concepts increased.

7 Conclusion

Conceptual models and other forms of knowledge bases can be viewed as the products emerged from human natural language processing. Self-organization is the key property of human mental activity and the present research investigated what self-organization properties can be found in the knowledge base documentation. It has been suggested to build conceptual model vector space and its SOM by comparing concept lattice received from manually constructed conceptual model and concept lattice received from SOM of the conceptual model. We argued that if both concept lattices resemble each other then we can say that IS documentation quality is acceptable.

Presented architectural solution for the software developers can be labor intensive. The payoff of such approach is an ability to generate formal language statements directly from IS documentation and IS user utterance. We have shown that with the SOM and FCA we can indicate inadequateness of the concept descriptions and improve the process of knowledge base development. Presented methodology can serve as the tool for maintaining and improving Enterprise-wide knowledge bases.

There were many research projects concerning questions of semantic parsing i.e. the automatic generation of the formal language from the natural language. But those projects were concerned only with semantic parsing as separate stage not integrated into the process of software development. Solution presented in this paper allows us to integrate IS design and analysis stages with the stage of semantic parsing. In this paper we demonstrated that we can label documents and user questions with the conceptual model concept name. In the future we hope to extend those results by generating SQL sentences and then querying databases. The present research has shown that if we want to build comprehensible model then, we must take more attention in describing concepts by the natural language.

Acknowledgment. The work is supported by Lithuanian State Science and Studies Foundation according to High Technology Development Program Project "Business Rules Solutions for Information Systems Development (VeTIS)" Reg. No. B-07042.

References

1. Burg, J.F.M., Riet, R.P.: Enhancing CASE Environments by Using Linguistics. International Journal of Software Engineering and Knowledge Engineering 8(4), (1998) 435–448.
2. Cunningham, H.: GATE, a General Architecture for Text Engineering. Computers and the Humanities, 36, (2002) 223–254.
3. Ganter B., Wille. R.: Formal Concept Analysis: Mathematical Foundations. Springer, Berlin-Heidelberg, (1999).
4. Hofmann, T.: Probabilistic latent semantic indexing. In Research and Development in Information Retrieval, (1999) 50–57.
5. Hotho, A., Staab, S., Stumme, G.: Explaining text clustering results using semantic structures. In Principles of Data Mining and Knowledge Discovery, 7th European Conference, PKDD 2003, Croatia. LNCS. Springer (2003) 22–26.
6. Hung, C., Wermter, S., Smith, P.: Hybrid Neural Document Clustering Using Guided Self-organisation and WordNet. Issue of IEEE Intelligent Systems, (2004) 68–77.
7. IBM. IBM Banking Data Warehouse General Information Manual. Available from on the IBM corporate site http://www.ibm.com (accessed July 2006).
8. IBM Voice Toolkit V5.1 for WebSphere Studio. http://www-306.ibm.com/software/ (accessed July 2006).
9. Kaski, S., Honkela, T., Lagus, K., Kohonen, T.: WEBSOM self-organizing maps of document collections. Neurocomputing, 21, (1998) 101–117.
10. Kohonen, T.: Self-Organizing Maps, Springer-Verlag, (2001).
11. Lagus, K., Honkela, T., Kaski, S., Kohonen, T.: WEBSOM for textual datamining. Articial Intelligence Review, 13 (5/6) (1999) 345-364.
12. Miller, G.A.: WordNet: A Dictionary Browser, Proc. 1st Int'l Conf. Information in Data, (1985) 25-28.
13. Ryan, K.: The role of natural language in requirements engineering. Proceedings of IEEE International Symposium on Requirements Engineering, IEEE Computer Society Press, (1993) 240—242.
14. Rolland, C., Proix, C.: A Natural Language Approach to Requirements Engineering. 4th International CAiSE Conference, Manchester UK, (1992) 257-277.
15. Salton. G.: Automatic Text Processing: The Transformation, Analysis and Retrieval of Information by Computer. Addison-Wesley, (1989).
16. Valtchev, P., Grosser, D., Roume, C., Rouane H. M.: GALICIA: an open platform for lattices. In A. de Moor B. Ganter, editor, Using Conceptual Structures: Contributions to 11th Intl. Conference on Conceptual Structures (2003) 241--254.

Coreference Resolution using Markov Logic Network

Shujian HUANG[1], Yabing ZHANG[1], Junsheng ZHOU[12] and Jiajun CHEN[1]

[1] State Key Laboratory for Novel Software Technology,
Nanjing University, Nanjing 210093, China
[2] Department of Computer Science, Nanjing Normal University,
Nanjing, Jiangsu, 210097, China
{huangsj, zhangyb, zhoujs, chenjj}@nlp.nju.edu.cn

Abstract. Most previous work treats the solution for pronouns and noun phrases either in two separate processes or in a single process. We argue that resolving them in two processes may result in the loss of potential useful information for each process. However, resolving them in a single process is also problematic. These two types of mentions have very different characteristics in some commonly used features. Current models cannot catch those differences and thus the two types may interfere with each other. In this paper, we propose a modeling strategy using Markov logic networks (MLNs) which can explicitly discriminate the two types in one single process. Experiments on ACE2005 Chinese dataset show that our modeling using MLNs, together with the correlation clustering technique, brings significant improvements to the task.

Key words: Coreference Resolution, Markov Logic Networks

1 Introduction

Coreference resolution (CR) has drawn a lot of attentions over the past decade, especially since McCarthy[1], Cardie and Wagstaff[2] introduced machine learning techniques into this field. It plays an important role in understanding complex texts and is widely used in a lot of applications such as question answering[3], summarization[4], etc. The strong relation with other popular topics such as entity resolution in database and citation analysis[5] makes it more attractive.

Pronoun and noun phrase are two major types of mentions in CR. There are two strategies for the resolution of them. One strategy tends to split the resolution of pronouns and noun phrases into two separate processes (*Separate Strategy*). Some works focus on just pronoun resolution, aiming to find the right antecedent for each pronoun[6–9]. Denis and Baldridge subdivide mentions into five categories such as third person pronouns, speech pronouns, etc. Then, specialized models are proposed for each individual type[10]. However, we argue that just considering pairwise relation between pronoun and each of its antecedent candidates does not make full use of the information among those candidate

© A. Gelbukh (Ed.)
Advances in Computational Linguistics.
Research in Computing Science 41, 2009, pp. 157-168

Received 10/11/08
Accepted 15/12/08
Final version 04/02/09

phrases. On the other hand, performing noun phrase resolution without considering pronouns may also lead to the loss of potential useful information.

In a more popular branch of researches, these two types of mentions are treated almost synchronously in a single process and only differs in some indicative features (*Uniform Strategy*)[1, 2, 11–15]. In this way, the interaction between noun and pronoun phrases can be captured by building up links between them. Recent works of Yang et al.[12] and Culotta et al.[16] proposed to solve this problem in a set-wise mode, which could capture more complex dependency relations.

However, some characteristic differences between the two types may bring conflicts to this kind of single process solution. We take two examples to informally explore these conflicts. String similarity is an important feature when judging the coreferential relation between noun phrases. Two noun phrases tend to refer to the same entity if their strings are similar to each other. For example, if the phrases "George W. Bush" and "President Bush" occur in the same text (as shown in Figure 1), they are very likely to refer to the same person. On the other hand, pronouns are not so sensitive to string similarity. Even two pronouns are identical, they can refer to different entities as well.

> *George W. Bush* is the 43rd President of the United States. ... Prior to his Presidency, *President Bush* served for 6 years as the 46th Governor of the State of Texas, where *he* earned a reputation for bipartisanship ...

Fig. 1. An example of coreference resolution. (Three phrases in italic refer to the same person.)

Similar conflict can be found in distance features. As we know, pronouns seldom refer to entities far away from them. Thus, long distance may have a strong negative impact on pronoun anaphora resolution. However, noun phrases have a much free characteristic of distance. Two noun phrases that are far apart, for example, occurring at the beginning and the end of an article, respectively, may both refer to the same entity. If pronouns and noun phrases share the same distance feature, the negative impact of long distance for pronoun anaphora will be interfered with by noun phrases. Thus, some pronouns may be linked to phrases that are far away from them, which is against our intuition. On the other hand, long distanced noun phrases co-refer will also be limited.

In this paper, we propose the modeling of coreference resolution using Markov logic networks, which can handle pronouns and noun phrases together while discriminate their differences. Specifically, we model the characteristics of pronouns and noun phrases using different formulas in Markov networks while still doing training and inference of them in the same process. A correlation clustering technique is also employed to get the final clustering results from pair-wise coreferential probabilities. Experiments show that our system achieves better results than several baseline systems that use *Separate* or *Uniform* strategies.

The rest of this paper is organized as follows: Section 2 reviews Markov logic networks. Section 3 presents our solution with MLNs. Section 4 reviews correlation clustering technique and presents its application in our work. We show our experimental settings and results in Section 5; and discuss related works in Section 6. Finally, we conclude in Section 7.

2 Markov Logic Networks

Markov logic network, introduced by Domingos and Richardson[17] is a well founded model for Statistical Rational Learning (SRL). Since MLNs are combinations of first order logic and Markov Networks, we firstly review these two parts briefly and then explain how they are used in our framework.

2.1 First Order Logic

First order logic is a formal language which describes the world by means of *constants*, *variables*, *functions*, *predicates* and *formulas*.

Constants are the elements in the world. In the scenario of coreference resolution, constants can be all the mentions in a document, such as "President Bush" and "he" in Figure 1.

A *Variable* is used to represent a set of constants. With typed variables, we can refer to different elements conveniently. For example, if we want to distinguish pronoun and noun phrases in a document, we can define two types of variables: *pronoun* and *noun.* Then we can use a variable p of the type pronoun to stand for phrases like "he", "she" and other pronouns; a variable n of the type noun to stand for phrases like "President Bush".

Functions refer to mappings between elements. For example, function *SemanticClass(Mention n)* can map a mention n to its semantic class. If n represents "President Bush", then the value of *SemanticClass(n)* is the constant *human.*

Predicates, which map a number of elements to a truth value, indicate properties of an element or relations between elements. For example, *IsFemale(Mention n)* indicates specify the gender of the mention n. The objective of coreference resolution can also be described by predicate. In this paper, we define the objective as *coreference(Mention n1, Mention n2)*, indicating whether mention n1 and n2 are coreferential.

Formulas are constructed from predicates using logical connectives and quantifiers and represent our knowledge of the world. We can formalize the interactions between the predicate *coreference* and other predicates into a set of formulas, which in first order logic is called a *knowledge base.*

2.2 Markov Networks and MLNs

First order logic uses a set of hard constrains (knowledge base) to describe the world. All the formulas in the knowledge base are treated equally, which means violating any of these constrains will be given an equal penalty. MLNs pack these constrains with weights, thus making the penalties higher for violations of higher weighted constrains. These weights are modeled by Markov Networks.

A Markov network (also known as Markov random field) is a model for the joint distribution of a set of variables $(X_1, X_2, ..., X_n) \in \chi$. Let G be an undirected graph with n nodes, each of which represents a variable. The model has a potential function for each clique in G. The joint distribution is given by

$$P(X = x) = \frac{1}{Z} \prod_k \phi_k(x_k) \tag{1}$$

where (x_k) is the state of the kth clique. And $\phi_k(x_k$ is the potential function on the kth clique. Z, known as the *partition function*, is given by $Z = \sum_{x \in \mathbf{X}} \prod_k \phi_k(x_k)$. Equation 1 can also be expressed in a log-linear form:

$$P(X = x) = \frac{1}{Z} \exp(\sum_j \omega_j f_j(x)) \tag{2}$$

where $f_j(x)s$ are feature functions indicating the state of cliques.

An MLN defines the probability of variable X in a similar way:

$$P(X = x) = \frac{1}{Z} \exp(\sum_j \omega_j n_j(x)) \tag{3}$$

where $n_j(x)$ is the number of true groundings[3] of $formula_j$ given x; partition function $Z = \sum_{x \in X} \sum_j \omega_j n_j(x)$[17]. Lowd and Domingos[18] propose an effective way of training the weight of MLNs. For more details please refer to their original work.

The predicates and functions of first order logic have the same expressive power as features in other probability models such as Decision Trees and Maximum Entropy. But the first order formulas give MLNs stronger expressive power in representing knowledge than any other model. The well-founded theories of Markov Networks provide us with an efficient way to perform inference according to the formulas.

3 Solution with MLNs

In this section, we will explain in detail our modeling of coreference resolution using Markov logic networks and why this modeling is able to discriminate different characteristics of pronoun and noun phrases.

3.1 Features

Following the work of Soon et al.[11], we use some lexical features, semantic features and contextual features. All these features are represented as predicates and functions (as shown in Table 1).

[3] A true grounding of formula f is a setting of constants assigned to the variables of f that satisfies f .

Table1.Predicates and functions used in MLNs

Predicates:	**Descriptions:**
Coreference (Mention, Mention)	indicate whether two mentions refer to the same entity
isPronoun (Mention)	indicate whether the mention is a pronoun
isDemostrative (Mention)	indicate whether the mention is demonstrative phrases
Overlap (Mention, Mention)	indicate whether the strings of two mention are overlapping
Apposition (Mention, Mention)	indicate whether the two mentions have an appositive relation
Functions:	**Descriptions:**
SClass (Mention)	the semantic class of the given mention, which is one of person, animal, object, time, space, unknown
Gender (Mention)	the gender of the given mention, which is one of male, female and unknown
Number (Mention)	the number of the given mention, which is one of singular, plural and unknown
SSimilarity (Mention, Mention)	the string similarity ratio between the head word of two mentions; the value is mapping into similar, normal and dissimilar bye threshold 2/3 and 1/3
SDistance (Mention, Mention)	The number of sentences between two mentions; the value is mapping into same, few and many by threshold 0 and 5

3.2 Knowledge Base

Automatically learning formulas from given training data is NP-Hard[19]. However, the MLNs framework provides us with a convenient way to explicitly combine statistical models with human knowledge, which helps a lot in resolving our problem.

In our experiment, we manually construct a few formulas as our first order knowledge base according to previous work and some basic heuristics, and use MLNs to learn the weights. These formulas mainly focus on the following aspects:

- Ordinary Features - Use features described in 3.1 to indicate whether two mentions co-refer. For example:

$$\forall u, v\ Overlap(u, v) \land (\neg isPronoun(u)) \land (\neg isPronoun(v)) \tag{4}$$

$$\Rightarrow Coreference(u, v)$$

$$\forall u, v\ SDistance(u, v) \land (\neg isPronoun(u)) \land (\neg isPronoun(v)) \tag{5}$$

$$\Rightarrow Coreference(u, v)$$

$$\forall u, v\ Apposition(u, v) \Rightarrow Coreference(u, v) \tag{6}$$

$$\forall u, v\ isDemonstrative(v) \Rightarrow Coreference(u, v) \tag{7}$$

$$\forall u, v\ Gender(u) = Gender(v) \Rightarrow Coreference(u, v) \tag{8}$$

- **Agreement Constraints - Prevent mentions that have conflict feature values** of number, gender or semantic class from co-refer. For example:

$$\forall u, v\ (Gender(u)! = Gender(v)) \wedge (Gender(u))! = \text{UNK}) \tag{9}$$

$$\wedge((Gender(v)! = \text{UNK}) => !Coreference(u, v)$$

$$\forall u, v\ (Number(u)! = Number(v)) \wedge (Number(u))! = \text{UNK}) \tag{10}$$

$$\wedge((Number(v)! = \text{UNK}) => !Coreference(u, v)$$

- **Reflexivity and Transitivity Constraints - Ensure that coreference is a equivalence relation.**

$$\forall u, v \quad Coreference(u, v) \Rightarrow Coreference(v, u) \tag{11}$$

$$\forall u, v \quad Coreference(u, w) \wedge Coreference(v, w) \Rightarrow Coreference(u, v) \tag{12}$$

An important advantage of MLNs over previously used models such as decision trees[11, 20], maximum entropy[21] and kernel based models[8] is that MLNs learn the weights of formulas instead of individual features (predicates and functions). As shown in formula 4, we can combine the string similarity feature with the type of the mentions (noun or pronoun) to get a single formula. This formula will only be effective when u and v are both noun phrases. In this way, we can effectively distinguish the similarity of noun phrases from that of pronouns.

Another advantage of MLNs is that it can perform a global inference instead of just making some local coreferential decisions[8, 11, 20, 21]. In our experiments, formula 11 and 12 are used to ensure the reflexivity and transitivity of coreference, which set up ties among all the coreferential decisions and make those decisions more consistent and reliable.

4 Correlation Clustering

Inference of above MLNs provides us with a probability of coreferential relation between every two mentions. And we use correlation clustering[22] technique to integrate all these pair-wise probabilities into a final clustering result.

Correlation clustering technique aims at providing a global metric for the clustering quality, which is helpful for deciding whether to continue clustering or not. We follow this way and define the global object function (equation 13) as to maximize the agreement within each cluster and the disagreement between clusters:

$$\max \sum_{u,v \in M} \theta(u, v) w(u, v) + \sum_{u,v \in M} (\theta(u, v) - 1) w(u, v) \tag{13}$$

$$s.t. \qquad \theta(u, v) + \theta(w, v) \leq \theta(u, w) + 1 \tag{14}$$

$$\theta(u, v) \in \{0, 1\} \tag{15}$$

$$\theta(u, u) = 0 \tag{16}$$

where M is the set of all mentions; $\theta(u,v)$ is an indicator of whether u and v are in the same cluster; $w(u,v)$ is a similarity measure of u and v; equation 14 ensures the transitivity of clustering result; equation 15 indicates this is a integer programming problem; equation 16 gets rid of the decision making between one mention and itself. In our experiments, we set $w(u,v)$ as probability of *coreference(u, v)* minus the average probability of *coreference(u, v)* for all (u,v) pairs.

As the above constrained integer optimization is NP-Complete[22], we use a greedy based, bottom-up approach to get an approximation. In each step, our approach searches for the best merging of existing clusters that can achieve the largest gain of objective function. It will execute the merging and iterate until no such merging can be found. To avoid misleading merging, we also use a compatibility test which prevents the merging of two clusters that have obviously conflicting features. For example, two clusters will not be merged if mentions of one cluster are identified as women's names, while mentions of the other are men's names.

5 Experiments

5.1 Toolkit and Corpus

We use Alchemy Toolkit[23] for training and testing with Markov logic networks. The corpus we use is the Chinese part of ACE2005 coreference resolution dataset. We skip the process of identifying mentions in the document; and instead, use the annotation of mentions provided in the dataset, which helps us focus on the resolution of coreference itself.

5.2 Systems

Two baseline systems are built following the *Uniform Strategy.* We implement a baseline system following Soon et al.[11], except that a SVM classifier is used instead of a C4.5 decision tree for coreferential relation. All predicates and functions listed in Table 1 are used as binary feature functions. Pairwise decisions are then combined using the best-first strategy[4]. We refer to this system as *SVM-Base.*

Another baseline system uses the same classifier as the first one, but uses correlation clustering for generating final results as described in Section 4. We refer to this system as *SVM-CC.*

We build two systems basing on Markov logic networks and correlation clustering, as described in Section 3. One of them is built according to the *Separate Strategy.* A first round resolution of noun phrases is performed, in which we only consider noun phrases resolution by removing all pronouns from training and testing mentions. Then, in a second round resolution, we add pronouns into the

[4] Each mention is linked to the most confident antecedent according to the output of the classifier[20].

previous result by linking them to their best antecedent. We refer to this system as *MLNs-S*.

In the last system, following the common *Uniform Strategy*, we resolve pronoun and noun phrases in a single process, and use a predicate *isPronoun(Mention)* to distinguish them. Specially designed first order logic formulas, such as formula 4 and 5 are used, so that different weights are learnt for the two mention types. We refer to this system as *MLNs-F*.

5.3 Results

MUC6 scores We use MUC6 metric to get the precision, recall and F-measure of final result. Table 2 shows the MUC6 score of our systems.

Table2. MUC6 scores of all mentions.

	Precision	Recall	F-Measure
SVM-Base	0.78884	0.71501	0.75011
SVM-CC	**0.8374**	0.69477	0.75945
MLNs-S	0.73751	0.78415	0.76011
MLNs-F	0.75972	**0.82378**	**0.79045**

As shown in the table, although we only perform a greedy search in correlation clustering, system *SVM-CC* still improves the result of system *SVM-Base* from 0.75011 to 0.75945. It is mainly because correlation clustering draws decisions according to a global scoring function rather than just local comparison like the best-first clustering. It is also worth noticing that *SVM-CC* achieves a much higher precision over *SVM-Base*, which indicates that our greedy search successfully finds a stop point rather than keeps merging small clusters up.

In *MLNs-based* systems, *MLNs-S* achieves a F-measure of 0.76011, just slightly better than *SVM-CC*. And *MLNs-F* achieves a highest 0.79045 F-measure, which is much better than all the previous systems. For further understanding of the differences between those systems, we compute the noun phrases' MUC6 scores for each system and list them below (in Table 3).

Noun phrases MUC6 scores Noun phrases' MUC6 score is computed with all pronouns removed from both the answers and system output.

Comparison between the results of *MLNs-S* (0.80242) and *SVM-CC* (0.79105) in Table 3 shows that separating the processes of pronoun and noun phrase resolution improves the noun phrase resolution. These two systems got almost the same score in all mentions (in Table 2), which indicates that the resolution of pronouns in *MLNs-S* is not good enough. We can mainly attribute this to *MLNs-S*'s *Separate Strategy* in pronoun resolution. *MLNs-S* only links each pronoun to its best antecedent and unfairly assumes that these decisions are independent of each other.

Table3. MUC6 scores of noun phrases.

	Precision	Recall	F-Measure
SVM-Base	0.79396	0.7541	0.77352
SVM-CC	**0.84597**	0.74283	0.79105
MLNs-S	0.78968	0.81557	0.80242
MLNs-F	0.76121	**0.85246**	**0.80425**

MLNs-F, although using *Uniform Strategy* again like *SVM-CC*, still achieves a comparable (actually slightly better) result (0.80425) on noun phrases resolution as *MLNs-S*. We attribute this to our specially designed formulas such as formula 4 and 5, which prevent the interfere between noun phrases and pronouns. What's more, the *Uniform Strategy* of *MLNs-F* achieves a better results on pronoun resolution than *MLNs-S*, thus bringing *MLNs-F* the highest overall F-measure.

Altogether, as evidenced by the experiment results, our modeling using MLNs successfully takes advantage of *Uniform Strategy* in pronoun resolution while avoiding the interfere between noun phrases and pronouns, and achieves significant better results over baseline systems.

6 Related Work

The exploration of feature conflict has been mentioned by Ng and Cardie[20]. They found that string similarity features were different for pronoun and other types of mentions. As a result, they suggested a split of features for each type of mentions, which did bring some improvements. However, they didn't get good enough results because of the use of much simpler models such as decision trees and an information gain based rule system called RIPPER. The modeling of MLNs is much simpler and more natural than splitting features.

The inspiration of using Markov logic networks comes from[5, 16]. Singla and Domingos[5] used MLNs in entity resolutions of database items, where several simple first order rules brought comparable results with existing methods. Culotta et al.[16] extended the knowledge source of coreferential decision making from mention pairs to mention clusters. Their work motivates us to use probabilistic graph model, such as Markov Networks, for coreference. They also mentioned the conjunction of features, which lead us to the use of logic connectives and quantifiers for building more complex formulas. However, they only reported results using a conjunction of size 2, which did not make full use of the expressive power of first order logic and was not able to capture the complex relations among features.

Yang et al.[12] proposed a twin-candidate model which considered the relation between three mentions instead of two in previous resolution framework. Denis and Baldridge[9] extended it into a candidate ranking model, which took all candidates of antecedent into consideration. Both of the two works solve this problem from a local view that only considers the antecedents for one mention at a time. But our modeling using MLNs is able to capture the global interactions not only between candidates of antecedent but also between any other two mentions in the article. And the use of correlation clustering provides a global arrangement of different coreferential relations, which may lead us to a better solution.

Our approach shares the same motivation with Choi and Cardie[14], namely, the resolving of anaphora needs structured information. They solved this problem in a framework based on conditional random field and achieved convincing results in an English corpus.

Poon and Domingos also proposed the use of Markov logic networks for coreference resolution[24]. However, they designed MLNs in an unsupervised manner, thus made their work quite different from ours. Instead of directly modeling the coreferential possibility between two lexicalized mentions, we are trying to model the latent rules for the CR task, which is more challenging and data dependent. As a large amount of CR data is usually difficult to get, some researchers began to explore unsupervised CR methods and also got promising results[24–26].

7 Conclusions and Future Work

We analyze the two strategies of pronouns and noun phrases coreference resolution, especially the relation and interference between these two mention types. Based on the analysis, we propose the use of Markov logic networks for solving these two types of coreference in a single process. Our model is able to use global information for anaphora resolution while successfully avoid the interference between pronouns and noun phrases. We also employ a correlation clustering technique which gives us a global metric during combining various coreferential relations from MLNs. Experiments show that our strategy improves the performance significantly over the baseline systems.

Future work will focus on integrating other knowledge sources like Centering Theory and syntactic information into our framework and making use of more shallow semantic information as[27].

To improve the performance, we plan to use LP chunking techniques for the solution of correlation clustering in Section 4, instead of currently used greedy based approach. We will also try to extend our pairwise target predicate to a setwise one, which may integrate the coreferential decision making and clustering into one process.

Another interesting direction is to automatically distinguish the features and knowledge bases between pronouns and noun phrases, and to further explore the interactions of pronouns and noun phrases in coreference resolution.

Acknowledgement. This work is supported by the National Natural Science Foundation of China under Grant No. 60673043, the National 863 High-Tech Program under Grant No. 2003AA010109, the National Social Science Foundation of China under

Grant No. 07BYY051, and the Natural Science Foundation of Jiangsu Higher Education Institutions of China under Grant No. 07KJB520057.

References

1. McCarthy, J.F., Lehnert, W.G.: Using decision trees for coreference resolution. In: Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence. (1995) 1050–1055
2. Cardie, C., Wagstaff, K.: Noun phrase coreference as clustering. In: Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, University of Maryland, MD, Association for Computational Linguistics (1999) 82–89
3. Morton, T.S.: Coreference for nlp applications. In: Proceedings of the 38th Annual Meeting on Association for Computational Linguistics. (2000)
4. Steinberger, J., Kabadjov, M.A., Poesio, M., Sanchez-Graillet, O.: Improving lsa-based summarization with anaphora resolution. In: Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing. (2005)
5. Singla, P., Domingos, P.: Entity resolution with markov logic. In: Proceedings of the Sixth International Conference on Data Mining, Washington, DC, USA, IEEE Computer Society (2006) 572–582
6. Wang, H., Mei, Z.: An empirical study on pronoun resolution in chinese. In: Computational Linguistics and Intelligent Text Processing, 5th International Conference, CICLing 2004, Seoul, Korea, February 15-21, 2004, Proceedings. (2004) 213–216
7. Iida, R., Inui, K., Matsumoto, Y.: Anaphora resolution by antecedent identification followed by anaphoricity determination. ACM Transactions on Asian Language Information Processing (TALIP) 4(4) (2005) 417–434
8. Yang, X., Su, J., Tan, C.L.: Kernel-based pronoun resolution with structured syntactic knowledge. In: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL, Morristown, NJ, USA, Association for Computational Linguistics (2006) 41–48
9. Denis, P., Baldridge, J.: A ranking approach to pronoun resolution. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence. (2007) 1588–1593
10. Denis, P., Baldridge, J.: Specialized models and ranking for coreference resolution. In: Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, Honolulu, Hawaii, Association for Computational Linguistics (October 2008) 660–669
11. Soon, W.M., Ng, H.T., Lim, D.C.Y.: A machine learning approach to coreference resolution of noun phrases. Computational linguistics **27**(4) (2001) 521–544
12. Yang, X., Su, J., Tan, C.L.: A twin-candidate model of coreference resolution with non-anaphor identification capability. In: Second International Joint Conference. (2005) 719–730
13. Nicolae, C., Nicolae, G.: Bestcut: A graph algorithm for coreference resolution. In: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, Sydney, Australia, Association for Computational Linguistics (July 2006) 275–283

14. Choi, Y., Cardie, C.: Structured local training and biased potential functions for conditional random fields with application to coreference resolution. In: Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics (HLT/NAACL), Rochester, New York, Association for Computational Linguistics (April 2007) 65–72
15. Yang, X., Su, J., Lang, J., Tan, C.L., Liu, T., Li, S.: An entity-mention model for coreference resolution with inductive logic programming. In: Proceedings of ACL-08: HLT, Columbus, Ohio, Association for Computational Linguistics (June 2008) 843–851
16. Culotta, A., Wick, M., Hall, R., McCallum, A.: First-order probabilistic models for coreference resolution. In: Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics (HLT/NAACL). (2007) 81–88
17. Domingos, P., Richardson, M.: Markov logic: A unifying framework for statistical relational learning. In: Proceedings of the ICML-2004 Workshop on Statistical Relational Learning and its Connections to Other Fields, Banff, Canada (2004) 49–54
18. Lowd, D., Domingos, P.: Efficient weight learning for markov logic networks. In: Proceedings of 11th European Conference on Principles and Practice of Knowledge Discovery in Databases. (2007) 200–211
19. Richardson, M., Domingos, P.: Markov logic networks. Machine Learning **62**(1-2) (2006) 107–136
20. Ng, V., Cardie, C.: Improving machine learning approaches to coreference resolution. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, Morristown, NJ, USA, Association for Computational Linguistics (2002) 104–111
21. Ng, H.T., Zhou, Y., Dale, R., Gardiner, M.: A machine learning approach to identification and resolution of one-anaphora. In: Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence. (2005) 1105–1110
22. Bansal, N., Blum, A., Chawla, S.: Correlation clustering. Machine Learning **56** (2004) 89–113
23. Kok, S., Singla, P., Richardson, M., Domingos, P.: The alchemy system for statistical relational ai. Technical report, Department of Computer Science and Engineering, University of Washington, Seattle, WA, http://www.cs.washington.edu/ai/alchemy/ (2005)
24. Poon, H., Domingos, P.: Joint unsupervised coreference resolution with Markov Logic. In: Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, Honolulu, Hawaii, Association for Computational Linguistics (October 2008) 650–659
25. Haghighi, A., Klein, D.: Unsupervised coreference resolution in a nonparametric bayesian model. In: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, Prague, Czech Republic, Association for Computational Linguistics (June 2007) 848–855
26. Ng, V.: Unsupervised models for coreference resolution. In: Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, Honolulu, Hawaii, Association for Computational Linguistics (October 2008) 640–649
27. Ng, V.: Shallow semantics for coreference resolution. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence. (2007) 1689–1694

A Ranking Approach to Persian Pronoun Resolution

Nafiseh Sadat Moosavi and Gholamreza Ghassem-Sani

Department of Computer Engineering, Sharif University of Technology, Iran
n_moosavi@ce.sharif.edu, sani@sharif.edu

Abstract. Coreference resolution is an essential step toward understanding discourses, and it is needed by many NLP tasks such as machine translation, question answering, and summarization. Pronoun resolution is a major and challenging subpart of coreference resolution, in which only the resolution of pronouns is considered. Classification approaches have been widely used for coreference/pronoun resolution, but it has been shown that ranking approaches outperform classification approaches in a variety of fields such as English pronoun resolution (Denis and Baldridge, 2007), question answering (Ravichandran, 2003), and tagging/parsing (Collins and Duffy, 2002; Charniak and Johnson, 2005). The strength of ranking is in its ability to consider all candidates at once and selecting the best one based on the model, while existing classification methods consider at most two candidate responses at a time. Persian and its varieties are spoken by more than 71 million people, and it has some characteristic that make parsing and other related processing of Persian more difficult than those of English. In this paper, we have evaluated maximum entropy ranker on Persian pronoun resolution and compared the results with that of four base classifiers.

Keywords: Natural Language Processing, Machine Learning, Ranking, Classification, Pronoun Resolution, Persian.

1 Introduction

The final goal of natural language processing (NLP) is that computers understand human languages. Different NLP research areas such as part of speech (POS) tagging, word sense disambiguation (WSD), and grammatical parsing concentrate only on a partial solution of this ultimate goal. All of these are required for a computer to understand a natural language.

NLP tasks can be divided into micro-tasks and macro-tasks. Micro-tasks focus on a word level processing or a sentence level processing such as WSD and parsing. On the other hand, macro-tasks include tasks which do a document level processing such as information retrieval and document classification. Before the introduction of machine learning approaches in NLP, higher level tasks such as semantic processing needed a variety of lower level tasks such as POS tagging and parsing. However, the use of machine learning methods may make it possible to obtain enough statistical in-

© A. Gelbukh (Ed.)
Advances in Computational Linguistics.
Research in Computing Science 41, 2009, pp. 169-180

Received 26/11/08
Accepted 15/12/08
Final version 04/02/09

formation to make these lower level tasks unnecessary. Although learning methods result in a satisfactory performance in many tasks, they are limited and hardly can provide complete necessary information.

There is a missing link between sentence level processing and document level understanding; and an important example of such a missing link is the resolution of pronouns.

Pronoun resolution is a crucial and difficult subpart of an overall task named coreference resolution. Coreference resolution determines the group of noun phrases that refer to the same real world entity.

In recent years, the ranking (re-ranking) approaches have been successfully applied to a variety of NLP tasks including English pronoun resolution (Denis and Baldridge, 2007), and the achieved results show that the ranker outperforms simple classification methods. In this paper, we presented the evaluation of a ranking model applied to the Persian pronoun resolution.

2 Related Work

Since there is no previous work on Persian coreference/pronoun resolution, we briefly review some of the most recent related works applied to English.

2.1 Classification

The usage of machine learning methods in pronoun resolution began with a simple naïve Bayes approach (Ge et al., 1998), in which the random variable was a candidate reference for a given pronoun. Soon et al. (2001) and Ng and Cardie (2002) used decision tree for coreference decisions; classification is done by pairing each candidate noun phrase and deciding whether they are coreferent or not.

Yang et al. (2003) proposed the use of competition learning in coreference problem. In their proposed method, every training sample is made by one anaphor and a pair of candidate antecedent, one positive antecedent and a negative one. In this way, the classifier has more ability to compare different candidates.

2.2 Clustering

Cardie and Wagstaff (1999) represented every noun phrase with a feature vector and then used a clustering algorithm for partitioning these feature vectors. Every resulting partition represents an entity. Avoiding triangle inconsistency is an advantage of clustering over classification. Classification makes decision about the pairs ("Mr. Green", "Green") and ("Green", "she") independently, so "she" in the second pair may be recognized as being coreferent with "Green", while its gender is incompatible with "Mr. Green", which is coreferent with "Green". However, in clustering these decisions are made dependently. Before adding a noun phrase to a cluster, its consistency with all existing noun phrases of that cluster is checked.

Wagstaff (2002) offered constrained clustering for coreference resolution. The proposed algorithm accepts the constraints in the forms of "must-link" and "cannot-link". "must-links" represent noun phrases that should be placed in the same cluster, and "cannot-links" indicate noun phrases that cannot be assigned to the same cluster.

2.3 Bell Tree

Luo et al. (2004) casted the coreference problem as a searching problem and represented the search space as a Bell tree. The root of the search tree is a single noun phrase, the second noun phrase is added to the next level of the tree, and the leaves contain the possible partitioning of all noun phrases

The goal is to find the most probable path from the root to the leaves; the leaf in the most probable path contains the resultant clustering.

2.4 Graph Partitioning

Nicolae and Nicolae (2006) introduced graph partitioning in coreference problem. Graph partitioning can be considered as a clustering algorithm in which the clustering is done by a graph cutting algorithm. Each node of the graph corresponds to a noun phrase. The weight of each edge shows how likely the two connected nodes are coreferent. These weights can be determined by any classification algorithm.

2.5 Co-training

Ng and Cardie (2003) modified the multi-view co-training algorithm (Blum and Mitchell, 1998) to fit the coreference resolution. Rather than separating the feature space into two compatible and uncorrelated views, they used two different learners in a co-training algorithm.

2.6 Conditional Random Fields

McCallum and Wellner (2004) offered three models for the use of Conditional Random Fields (CRF) in coreference resolution. All of the proposed models are conditionally-trained, undirected graphical models which, unlike previous models, are relational. In a relational model, the dependency between the training data and the input features is considered.

2.7 Data Mining

In the method proposed by Harabagiu et al. (2001), the resolution rules from a large corpus are mined and the entropy of each rule is evaluated accordingly. The partitioning of noun phrases is done in a manner in which more rules with a higher confidence accept the specified partitioning.

Bean and Riloff (2004) presented a system which mines the relations between the words and their contexts. In this way, for each noun phrase a kind of semantic rule is achieved, and it helps the improvement of coreference resolution.

Bergsma and Lin (2006) offered a method in which the likelihood of the coreference between a pronoun and its candidate antecedent is learned based on the dependency parse path between the pronoun and its candidate antecedents.

2.8 First-Order Probabilistic Model

Culotta et al. (2007) offered a particular method for doing training and inference in first-order models of coreference, in which the features are over a set of noun phrases rather than a pair of noun phrases. Their method results in a first-order probabilistic model for coreference resolution. First-order probabilistic logic is a first-order logic that associates a real-valued parameter to every predicate.

2.9 Ranking

Denis and Baldridge (2007) proposed a supervised ranking approach for pronoun resolution. The ranking enables all candidate antecedents to be evaluated together; whereas classification methods examine at most two candidate antecedents at a time. They showed that their proposed method outperforms the best classification method.

3 Persian Pronouns

Persian and its varieties are spoken by more than 71 million people in Iran, Afghanistan, and Tajikistan. Normal Persian sentences are structured as " (optional subject) (optional prepositional phrase) (optional object) verb". However, it can also have a free word order in many places, and this characteristic make parsing and other related processing of Persian more difficult than those of English.

Persian is a null-subject language, and nominal pronouns can be omitted from a sentence. It has 18 different pronouns: four first person pronouns, four second person pronouns, nine third person pronouns, and one pronoun which doesn't have number and can be used in place of every noun phrase.

Persian pronouns have special characteristics that make their resolution more difficult than that of English pronouns. Persian's third person pronouns do not have any gender information. Besides, some plural pronouns are sometimes used in place of singular human pronouns to show respect, and singular pronouns may also refer to plural non-human noun phrases. These characteristics cause that the gender and number agreement, which are two of the most effective features in pronoun resolution, become either useless or less effective in Persian pronoun resolution.

4 Ranking Approaches

Using ranking (re-ranking) approaches for solving complex natural language processing problems has increasingly received attention in recent years. The main motivation for using the ranking approaches is their ability to directly compare between different responses and pick the most proper response.

Ranking has been successfully applied in a variety of NLP tasks including machine translation (Och, 2003; Shen et al, 2004), question answering (Ravichandran et al, 2003), parsing (Collins, 2000; Charniak and Johnson, 2005), and English pronoun resolution (Denis and Baldridge, 2007).

As it has been mentioned in section 3, Persian pronoun resolution has some characteristics that make it more difficult than that of English. Thus, it can be considered as a new NLP domain for evaluating the strength of a ranking method.

4.1 Maximum Entropy Ranker

The problem of pronoun resolution can be modeled with Maximum Entropy (MaxEnt) in two different ways: classification and ranking. A MaxEnt classifier allocates each pair (consist of a pronoun and a candidate antecedent) to one of "coreferent" or "non-coreferent" classes, while a MaxEnt ranker selects a single candidate as the antecedent of a pronoun. This means that a MaxEnt classifier can select many candidates as antecedents of a single pronoun, as long as the pairs including those antecedents and the pronoun are marked as "coreferent". In contrast, a MaxEnt ranker always selects the most probable candidate antecedent as a pronoun's antecedent.

Suppose we have a set of candidate antecedents $A = \{a_1, a_2, ..., a_n\}$ for a pronoun p, and $f_k(a, p), k = 1,..., K$ are K different feature functions which calculate the features based on the pair (a, p). The maximum entropy classifier can be modeled as equation 1,

$$p(c \mid a, p) = \frac{\exp[\sum_{k=1}^{K} \lambda_{k,c} f_k(a, p)]}{\sum_{c'} \exp[\sum_{k=1}^{K} \lambda_{k,c'} f_k(a, p)]} \quad \textbf{(1)}$$

where, $\lambda_{k,c}$ $k = 1,..., K$ and $c = \{coreferent, non-coreferent\}$ are the model parameters.

Maximum entropy ranker is modeled as equation 2,

$$p(a \mid p) = \frac{\exp[\sum_{k=1}^{K} \lambda_k f_k(a,q)]}{\sum_{a'} \exp[\sum_{k=1}^{K} \lambda_k f_k(a',q)]} \tag{2}$$

where, λ_k $k = 1,\dots,K$ are the model parameters.

In the classifier model, for each class (coreferent and non-coreferent), the weights of each feature functions are computed separately, whereas in the ranker model, the weights do not depend on class labels and there is the same set of feature weights for all classes. Thus, the ranker model allows all candidate antecedents to be compared together.

5 A Persian Pronoun Resolution System

This section describes a Persian pronoun resolution system which casts the pronoun resolution as a classification and ranking task.

5.1 Data Set

In order to use a learning method for coreference resolution, a suitable coreferentially annotated corpus is needed. The development and evaluation of automatically trained coreference systems is dependent on the existence of such corpora.

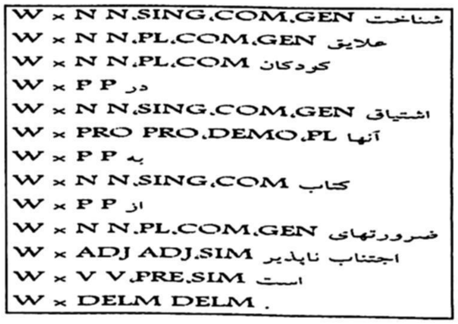
In this section, we briefly describe a Persian coreferentially annotated corpus, PCAC-2008, which was developed by appropriate annotation of another Persian corpus named Bijankhan (Bijankhan, 2004).

5.1.1 Bijankhan Corpus

Bijankhan is a corpus containing a huge number of Persian syntactic-semantic annotated documents. It contains wide variety of linguistic data in different subjects. It can be considered as a complete statistical universe of Persian documents. Bijankhan includes syntactic and semantic tagging. It is organized and tagged as a word-level corpus.

Bijankhan contains 3050 different documents and is based on daily news and common texts. One example of the supplied information in Bijankhan is shown in Fig. 1.

Each line contains a word and some syntactic and semantic features such as part of speech (POS), number information of that word. The previous applications of Bijankhan includes morphological analysis (Feyzbakhsh et al., 2008) and unsupervised grammar induction (Mirroshandel et al. 2007; Mirroshandel and Ghassem-Sani, 2008).



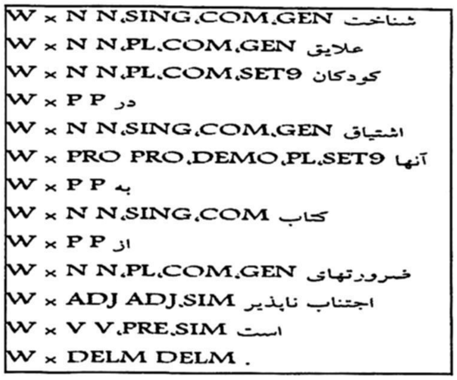
```
W × N N.SING.COM.GEN شناخت
W × N N.PL.COM.GEN علایق
W × N N.PL.COM کودکان
W × P P در
W × N N.SING.COM.GEN اشتیاق
W × PRO PRO.DEMO.PL آنها
W × P P به
W × N N.SING.COM کتاب
W × P P از
W × N N.PL.COM.GEN ضرورتهای
W × ADJ ADJ.SIM اجتناب ناپذیر
W × V V.PRE.SIM است
W × DELM DELM .
```

Fig. 1. Annotation of Bijankhan for a sample sentence: Each word of a sentence was tagged in a separate line which contains some basic syntactic-semantic knowledge of that word. Each line begin with "W *" marker and then followed by a POS of each word repeated two times. N, P, PRO, ADJ, V, DELM are the abbreviation of noun, proposition, pronoun, adjective, verb and delimiter respectively. The last word of each line is the tagged word itself.

5.1.2 PCAC-2008 Corpus

PCAC is the abbreviation of Persian Coreferentially Annotated Corpus. It is a partial extension of Bijankhan corpus, in which the coreference information has been added. Different Bijankhan articles in different topics and lengths were annotated in PCAC.

PCAC consists of 2006 labeled pronouns drawn from 30 different documents, and each pronoun is marked with its nearest antecedent. Fig. 2 shows how the annotations of Bijankhan have been extended in PCAC-2008.



```
W × N N.SING.COM.GEN شناخت
W × N N.PL.COM.GEN علایق
W × N N.PL.COM.SET9 کودکان
W × P P در
W × N N.SING.COM.GEN اشتیاق
W × PRO PRO.DEMO.PL.SET9 آنها
W × P P به
W × N N.SING.COM کتاب
W × P P از
W × N N.PL.COM.GEN ضرورتهای
W × ADJ ADJ.SIM اجتناب ناپذیر
W × V V.PRE.SIM است
W × DELM DELM .
```

Fig. 2. Annotation of PCAC-2008 for the annotated sentence of Fig. 1: : Coreference information has been added to the Bijankhan annotations by the use of "SET" feature.

5.2 Training Samples

Positive training samples were made by pairing each pronoun and its nearest antecedent, and negative samples were made by pairing pronouns and their negative antecedents. Every noun phrase between each pronoun and its nearest antecedent is considered as a negative antecedent of that pronoun.

5.3 Feature Set

We used the Denis and Baldridge (2007) feature set for evaluating the Persian pronoun resolution system. In addition to the features explained in (Denis and Baldridge, 2007), a genitive feature was also added which determines whether a pronoun is genitive or not. This feature is effective in Persian pronoun resolution and Bijankhan annotation contains this information. Besides, we have not used the gender agreement feature due to the lack of gender information in Persian pronouns.

The used feature set is composed of 25 features that can be divided into three groups: 1) features describing the pronoun, 2) features describing a candidate antecedent, and 3) features describing the relationship between the pronoun and candidate antecedents.

5.4 Learning Methods

5.4.1 Classification Algorithms

Theoretical studies in machine learning such as what was done by Wolpert and Macready (1995), have demonstrated that none of the inductive algorithms is generally superior to others. In order to see which learning algorithm is the most proper for a specific language processing task, it is necessary to compare different machine learning methods experimentally on that particular task. If the bias of a learning algorithm better fits to the properties of a specific task, the resulting model would be more suitable for the new data of the same task.

Daume (2006) mentioned that the most popular choices of the classifier for the coreference resolution task in the literatures are decision trees and MaxEnt models. However, it doesn't mean that these learning methods are also the most appropriate ones for a coreference resolution task.

We used these two classifiers plus Perceptron and SVM classifiers, which are regarded as two of the most effective methods for the binary classification problems.

5.4.2 The Ranking Algorithm

Maximum Entropy modeling has been extremely successful for many ranking tasks (Denis and Baldridge, 2007; Charniak and Johnson, 2005; Elwell, 2008; Ji and Grishman, 2005; Kim and Hovy, 2005; Nguyen and Kim, 2008; Wellner and Pustejovsky, 2007; etc). Thus, we used it for evaluating the effect of ranking in Persian pronoun resolution.

5.5 Testing the Trained System

After training the ranker and classification algorithms, the trained learners are used to guide the resolution of pronouns. First, the learning samples are made by finding candidate antecedent for each pronoun and pairing them. After preparing the learning samples, and in the case of classifiers, each pair is examined by the classifier and is classified as a "coreferent" or "non-coreferent". The candidate antecedent of each coreferent pair is considered as an antecedent of that pair's pronoun. Thus, each pronoun may have several antecedents.

In the case of ranker, all pairs made for the same pronoun are considered at the same time, and the candidate antecedent of the most probable pair is selected as the pronoun's antecedent.

6 Evaluations

The evaluation of the ranker model in contrast with the classification methods is presented in this section.

The same preprocessing modules and feature set were used for evaluation of classification methods and the ranker model; we performed a 10-fold cross validation to evaluate each of the examined methods. The performance is reported in terms of precision, recall and F_1-measure of the referential pronouns.

Regarding setting the learner-specific parameters, we used the default values for all examined learners unless otherwise stated. In the case of MaxEnt, we used 100 iterations of the improved iterative scaling algorithm using Gaussian prior.

The SVM learner was evaluated by RBF, sigmoid, and polynomial kernels and with different degrees for polynomial kernel. The SVM result reported in Table 1 is the best achieved result (i.e., the polynomial kernel of degree 3).

6.1 Results and Discussion

The results of classification methods in comparison with that of the ranker model are presented in table 1. The results show that the MaxEnt ranker significantly improves the MaxEnt classifier; however, its results are not better than that of the C4.5 and Perceptron base classifiers (while MaxEnt ranker performs better than the best classification method for English pronoun resolution (Denis and Baldridge, 2007)). Thus, the achieved results confirm the Wolpert and Macready (1995) studies and show that the decision tree learner has a better performance than the other examined methods in the Persian pronoun resolution.

Table 1. Results of C4.5, Perceptron, SVM and MaxEnt classifiers in comparison with MaxEnt ranker

Learning method	Recall	Precision	F_1
C4.5	31.70	75.99	44.73
Perceptron	27.47	49.64	35.36
SVM	17.00	79.12	27.98
MaxEnt classifier	4.01	19.92	6.68
MaxEnt ranker	30.34	30.34	30.34

6.2 Error Analysis

Denis and Baldridge MaxEnt ranker (2007) achieved the f_1-measure of 74.0%, while the achieved result for Persian pronoun resolution is 30.34%. Our error analysis reveals that the poor performance of the two examined system (classification and ranker systems) can mainly be attributed to the following reasons:

1. Our non-statistical parser and the Persian free word order grammar that result in a highly unbalanced training data in which positive samples are only 2.8% of the whole data. The parser is used for determination of noun phrases; a non-statistical parser with a free word order grammar finds more noun phrases between a pronoun and its nearest antecedent, and thus results in a highly unbalanced data set.
2. The lack of gender and number agreement features that was addressed in section 3.

7 Conclusions

We have evaluated the maximum entropy ranker and four base classifiers for Persian pronoun resolution in order to evaluate the effect of ranking model in this domain.

The results show that the MaxEnt ranker significantly outperforms the MaxEnt classifier: improving the F_1 measure from 6.68% to 30.34%. However, it does not outperform all the examined classifiers; C4.5 and Perceptron classifiers are more suitable and had better performance in Persian pronoun resolution.

One can suggest several possible types of futures works to improve the performance of the presented system. We used the MaxEnt ranker because it is the most common ranking technique in NLP literatures; however, the use of the probability estimation trees (PETs) is an important avenue to explore further. PETs are trees which estimate the probability of class membership and can be used as a ranker (Breiman et al, 1984, Provost and Domingos, 2000, Margineantu and Dietterich, 2001). As it was shown, the decision tree classifier achieved the best results for the Persian pronoun resolution. On the other hand, the results show that the maximum entropy ranker significantly outperforms the maximum entropy classifier. Thus, the use of PETs for Persian pronoun resolution can be considered as an important extension of this study, which may improve the results considerably.

The use of syntactic parse tree as a structured feature is another important area for future works. Yang et al. (2006) presented a method in which a syntactic parse tree was used as a structured feature, and then proper kernels were applied to such a feature, together with other ordinary feature. Their results showed that the system including the structured feature could increase the success rate significantly. We weren't able to use such a structured feature due to the lack of a non-commercial probabilistic parser. For example, our non-statistical parser finds 2050 different parse trees (and 16 different noun phrases) for a simple sentence with POSs like "N N N N P N N N V". Thus, the use of convolution tree kernels seems impractical with these huge number of parse trees for each sentence.

References

1. Bean, D., Riloff, E.: Unsupervised learning of contextual role knowledge for coreference resolution. In: HLT-NAACL, pp. 297--304 (2004).
2. Bergsma, Sh., Lin, D.: Bootstrapping Path-Based Pronoun Resolution. In: COLING/ACL-06, pp. 33--40 (2006).
3. Bijankhan, M.: The rule of linguistic corpora in writing a grammar: an introduction to a software. Iranian Journal of Linguistics, vol 19 (2004).
4. Blum, A., Mitchell, T.: Combining Labeled and Unlabeled Data with Co-training. In: COLT, pp 92--100 (1998).
5. Breiman, L., Friedman, J. H., Olshen, R. A., Stone, C. J.: Classification and regression trees. Wadsworth international group (1984).
6. Cardie, C., Wagstaff, K.: Noun Phrase coreference as clustering. In: EMNLP/VLC, pp. 82--89 (1999).
7. Charniak, E., Johnson, M.: Coarse-to-fine n-best parsing and maxent discriminative reranking. In: ACL (2005).
8. Collins, M.: Discriminative reranking for natural language parsing. In: ICML-2000 (2000).
9. Culotta, A., Wick, M., Hall, R., McCallum, A.: First-Order Probabilistic Models for Coreference Resolution. In: NAACL HLT 2007, pp. 81--88 (2007).
10. Daume, H.: Practical Structured Learning Techniques for Natural Language Processing. Ph.D. dissertation, Department of computer science, University of Southern California (2006).
11. Denis, P., Baldridge, J.: A Ranking Approach to Pronoun Resolution. In: IJCAI, pp. 1588--1593 (2007).

12.Elwell, R. B.: Robust Methods for Automated Discourse Connective Argument Head Identification. Master thesis, University of Texas at Austin (2008).

13.Feyzbakhsh, M., Sadraei, M., Ghassem-Sani, Gh. R.: Unsupervised Morphology of Persian Words. In: CSICC'2008 (2008).

14.Ge, N., Hale, J., Charniak, E.: A statistical approach to anaphora resolution. Sixth Workshop on Very Large Corpora (1998).

15.Harabagiu, S., Bunescu, R., Maiorano, S.: Text and knowledge mining for coreference resolution. In: NAACL, pp. 55--62 (2001).

16.Kim, S. M., Hovy, E.: Identifying Opinion Holders for Question Answering in Opinion Texts. In: AAAI (2005).

17.Luo, X., Ittycheriah, A., Jing, H., Kambhatla, N., Roukos, S.: A mention-synchronous coreference resolution algorithm based on the bell tree. In: ACL, pp. 136--143 (2004).

18.Margineantu, D. D., Dietterich, T. G.: Improved class probability estimation from decision tree models. In: Nonlinear Estimation and Classification (2001).

19.McCallum, A., Wellner, B.: Conditional models of identity uncertainty with application to proper noun coreference. In: NIPS (2004).

20.Mirroshandel, A., Ghassem-Sani, Gh.: Using of the Constituent Context Model to Induce a Grammar for a Free Word Order Language: Persian. In: LTC2007, pp. 443--447 (2007).

21.Mirroshandel, A. Ghassem-Sani, Gh.: Unsupervised Grammar Induction Using a Parent Based Constituent Context Model. In: ECAI (2008).

22.Ng, V., Cardie, C.: Improving machine learning approaches to coreference resolution. In: *ACL*, pp. 104--111 (2002b).

23.Ng, V., Cardie, C.: Bootstrapping Coreference Classifiers with Multiple Machine Learning Algorithms. In: EMNLP, pp. 113--120 (2003).

24.Nguyen, N. L.T., Kim, J. D.: Exploring Domain Differences for the Design of Pronoun Resolution Systems for Biomedical Text. In: Coling 2008, pp. 625--632 (2008).

25.Nicolae, C., Nicolae, G.: BESTCUT: A graph algorithm for coreference resolution. In: EMNLP, pp. 275--283 (2006).

26.Och, F. J.: Minimum error rate training for statistical machine translation. In: ACL (2003).

27.Provost, F., Domingos, P.: Well-trained PETs: Improving probability estimation trees. CeDER working paper (2000).

28.Ravichandran, D., Hovy, E., Och, F. J.: Statistical qa - classifier vs re-ranker: What's the difference? In: ACL Workshop on Multilingual Summarization and Question Answering-Machine Learning and Beyond (2003).

29.Shen, L., Sarkar, A., Och, F. J.: Discriminative reranking for machine translation. In: NAACL/HLT (2004).

30.Soon, W. M., Ng, H. T., Lim, D. Ch.: A machine learning approach to coreference resolution of noun phrases. In: Computational Linguistics 27(4), pp. 521--544 (2001).

31.Wagstaff, K.: Intelligent Clustering with Instance-Level Constraints. Ph.D. dissertation, Department of computer science and engineering, Cornell University (2002).

32.Wellner, B., Pustejovsky, J.: Automatically identifying the arguments of discourse connectives. In: EMNLP-CoNLL, pp. 92--101 (2007).

33.Wolpert, D., Macready: No Free Lunch Theorems for Search. Technical report *SFI-TR-95-02-010*, Santa Fe Institute (1995).

34.Yang, X., Zhou, G., Su, J., Tan, Ch. L.: Coreference resolution using competitive learning approach. In: ACL, pp. 176--183 (2003).

35.Yang, X., Su, J., Tan, Ch. L.: Kernel-Based Pronoun Resolution with Structured Syntactic Knowledge. In: COLING/ACL-06, pp. 41--48 (2006).

Classification of Clinical Conditions: A Case Study on Prediction of Obesity and Its Co-morbidities

Archana Bhattarai, Vasile Rus and Dipankar Dasgupta

Department of Computer Science, The University of Memphis,
209 Dunn Hall
Memphis, TN 38152-3240, USA
{abhattar, vrus, dasgupta}@memphis.edu

Abstract. We investigate a multiclass, multilabel classification problem in medical domain in the context of prediction of obesity and its co-morbidities. Challenges of the problem not only lie in the issues of statistical learning such as high dimensionality, interdependence between multiple classes but also in the characteristics of the data itself. In particular, narrative medical reports are predominantly written in free text natural language which confronts the problem of predominant synonymy, hyponymy, negation and temporality. Our work explores the comparative evaluation of both traditional statistical learning based approach and information extraction based approach for the development of predictive computational models. In addition, we propose a scalable framework which combines both the statistical and extraction based methods with appropriate feature representation/selection strategy. The framework leads to reliable results in making correct classification. The framework was designed to participate in the second i2b2 Obesity Challenge.

Keywords: Text classification, Information Extraction, natural language processing

1 Introduction

Medical informatics is chiefly inductive and information intensive science where observation and analysis of comprehensive clinical data can lead to complex and powerful evidence based decision support systems. One of the primary goals of these automated systems is to make information more accessible, representative and meticulous in a quick span[4]. Furthermore, they have gained increased importance in the recent years as it can even outperform a human expert in some cases in diagnosing diseases as the process is highly subjective and fundamentally depends on the experiences of the assessor and his/her interpretation on the information[4]. Conversely, most medical institutions are still keeping a large amount of medical data in narrative form resulting in huge volume of potential information with limited or no utility and accessibility. An effort to exploit this data poses multiple challenges as it involves processing free text data with the presence of acronyms, synonyms, negation

© A. Gelbukh (Ed.)
Advances in Computational Linguistics.
Research in Computing Science 41, 2009, pp. 183-194

Received 11/11/08
Accepted 11/12/08
Final version 06/02/09

and dependence on temporality. Thus, in this work we try to explore different challenges that arise while trying to identify obese patients and the co-morbidities exhibited by them based on the narrative patient record.

The work was done specifically to participate in the "Obesity Challenge (A Shared-Task on Obesity): Who's obese and what co-morbidities do they (definitely/likely) have?" under Second i2b2 Shared-Task and Workshop Challenges in Natural Language Processing for Clinical Data organized by "Informatics for Integrating Biology and the Bedside, i2b2, a National Center for Biomedical Computing"[1].

The problem of identifying obese patients and the co-morbidities exhibited by them can be intuitively modeled as a document classification problem. Traditional approach of document classification task uses keyword based document representation (for example: bag of words representation) along with some statistical techniques to classify relevant and irrelevant documents[4]. These statistical techniques automatically identify useful keywords based on large training corpus. These methods have gained popularity because of ease and full automation. However, these methods exhibit serious limitations on deep semantics representation[5]. Some of the limitations of traditional classification methods in the context of medical reports are as follows:

Synonymy/Hyponymy/Acronym Consideration: It is a common practice to represent the same concept in different forms or words in medical domain. The use of acronyms is also highly predominant in the area. This factor of presence of synonyms/hyponyms and acronyms cannot be captured with word based classification system as traditional classification systems primarily performs keyword based classification without consideration of the context.

Negation handling: Medical documents contain a prominent amount of negative qualifiers. These qualifiers exhibit their significance only in local contexts. For example, in the sentence "the patient does not have diabetes", the negative qualifier "not" should be associated with diabetes. This characteristic cannot be captured fully in word or even n-gram feature based models.

Temporality: Statistical classification algorithms cannot capture the time varying components significantly present in medical reports. For example, a co-morbidity may be present in the past, but not in the present. Identification of such aspect demands complex semantic analysis.

Experiencer: The reference of a keyword, a phrase, sentence or the paragraph may be subjective to a different person and not the patient. For example, the sentence "FAMILY HISTORY: No family history of kidney disease or heart disease" talks about the patients family and not specifically about the patient. Such semantic nformation is not captured in statistical classification algorithms.

Thus we propose to apply the concept of Information Extraction based classification to address some of the problems explained above. Our work mainly focuses on the automated identification of synonymous/hyponymous words and acronyms. The method also learns the interdependence of the co-morbidities to exploit the property of high interdependence of co-morbidities. Finally, we apply existing negation handling algorithm, NegEx[2] in our work. We have not included the temporality and experiencer problem in our work. However, the results are promising enough without their consideration.

2 Related Work

The task of medical document classification has been successfully applied to several real world medical diagnosis problems [4]. Most of the medical classification task apply statistical machine learning algorithms such as Naïve Bayes' classification, kernel based algorithms (Support vector machines (SVM), rules developing algorithms (decision trees) for different problem diagnosis [4]. Applications of pattern recognition in medical domain have also used Artificial Neural Network (ANN) [6]. However, very little has been done related to deep semantic analysis. Negation handling and word sense disambiguation is relatively more explored area compared to automated synonym/acronym extraction. Several works have been done in the recent years in negation handling. Chapman et al [2] developed a regular expression based negation determination algorithm. It defines a wide range of negation phrases that either appears before or after a finding in medical domain. It also defines a window size 'n' within which if the negation word occurs, the finding is considered to be negated. Another algorithm developed by Aronow et al [3] exploits syntactic processing techniques to identify noun phrases or conjunctive phrases to determine negation scope. Machine learning based algorithms have also found their application in identifying negative patterns in the text. Averbuch et al [7] developed an information gain based selection algorithm to automatically learn negative patterns from the text. Goryachev et al[8] did a comparative analysis on negative algorithms and developed a system with modified Negex and Aronow algorithm. They have also implemented Naïve Bayes' and Support Vector Machine based algorithm to automatically learn negation detection process using a set of manually annotated discharge summaries. Based on the observation on related works and simplicity, we implemented a simplified form of NegeX[2] algorithm to detect negated co-morbidities in our work.

For synonym/acronym extraction, thesauri based methods have been found to be useful in query expansion for information retrieval [9]. Domain independent thesauri such as WordNet [10] has proven to be helpful in a generalized information retrieval scenario[11]. However, such thesauri give very poor or no coverage for highly domain specific hyponym extraction scenario such as ours. Manual construction of such domain specific thesauri is expensive. As a result, automatic domain specific synonyms/hyponyms extraction has been started without being successful for substantial accuracy[12][13][14]. Term variation based hyponym detection[15] and distributional similarity[16] based synonym extraction methods have also been explored. In the term variation based method, variations of term such as "mouth cancer" and "cancer of mouth" [17] is studied and analyzed. In the distributional similarity based method, terms occurring in the proximity of known terms are taken to be similar which has been shown to increase recall at the cost of precision[16]. McCrae et al[17] used automated regular expression based patterns starting from few seed patterns to discover more patterns with a heuristic search method. All the above mentioned methods still do not capture all the problem scenarios. For example, in our case, co-morbidity is mentioned in one form in one report and in another form in another which confronts the challenge of detecting synonyms from unassociated data. Our work is most closely related to the work in Riloff et al[5] which explains different information extraction based algorithms for high precision text classification. The idea is to automatically build domain specific dictionary from the given training corpus which is then used for information extraction task. The method is fully scalable and portable to any domain. On the negative side, this method is good for high precision results only with a compromise on recall.

3 Methods

The identification of obesity and its co-morbidities is a multiclass, multilabel classification problem where each patient may have multiple diseases and each disease can be marked with any of the labels such as Y for "Yes" meaning the patient has the co-morbidity, N for "No" meaning the patient does not have the co-morbidity, "Q" for Questionable meaning questionable whether the patient has the co-morbidity or "U" for unmentioned meaning the co-morbidity is not mentioned in the record. The judgments are provided in two forms; "textual judgments" and "intuitive judgments". Textual judgments are strictly based on text and intuitive judgments are based on implicit information in the narrative text.

The basic idea in our work (as explained in section 1) is to combine the best parts of traditional statistical learning methods and information extraction based methods. Traditional learning methods exhibit high recall with relatively good precision whereas extraction based methods exhibit high (near to perfect) precision. We evaluate various machine learning algorithms to obtain best classification result for the problem domain. We then apply extraction based method to refine the results obtained from statistical method to obtain better precision retaining high recall. In the process, we address specific challenges posed by both the statistical classification method and extraction based method.

Statistical machine learning based methods exhibit promising results in most of the general cases. However, the methods cannot perform to its best when the data exhibits a very high dimension. Moreover, these methods also cannot give accurate results when there is a high interdependence between the classes of a multiclass problem. We discuss these problems and our approach of solution in the following sub-sections. Similarly information extraction based methods also encompass challenges such as representative entity extraction, synonyms/hyponyms identification, negation handling etc which are discussed in the following sub-sections.

3.1 Multiclass Multilabel Classification

Our problem involves both the multilabel classification and multinomial or multiclass classification. Multiclass classification is a classification problem where a document can belong to one of several classes (more than two classes). The classes in multiclass classification are mutually exclusive. Multilabel classification is defined as a classification problem where a document can belong to several classes simultaneously or to a single class or to none of the classes [18]. Most of the multilabel classification algorithms consider that the classes are independent of each other. With this assumption, the classification problem turns down to multiple binary classification problems. However, this generalization in our case is very expensive as all the co-morbidities are highly related to each other and the presence of one helps to induce the other.

3.2 Information Extraction

Information extraction (IE) is a type of information retrieval which extracts structured information, i.e. categorized and contextually and semantically well-defined data from a certain domain, from unstructured machine-readable documents automatically [19]. A complete translation of the semantics of a document requires an in-depth natural language processing which is computationally expensive. However, information extraction is a more focused and well-defined task. The advantage of information extraction is that the document that is not relevant to the context can be ignored thus reducing the computational cost effectively.

In our work, we tried to extract representative medical terms from the report that could be used as important keywords to characterize the report. We specifically extracted four medical concepts from the report namely; diseases or syndromes, sign or symptoms, body parts and clinical drugs. To extract the medical concepts, we used MetaMap Transfer (MMTx) [20]. This software processes text through a series of modules. First it is parsed into components including sentences, paragraphs, phrases, lexical elements and tokens. Variants are generated from the resulting phrases. Candidate concepts from the UMLS Metathesaurus[21] are retrieved and evaluated against the phrases. The best of the candidates are organized into a final mapping in such a way as to best cover the text.

3.3 Finding Synonyms/Acronyms and Class Interdependence: Riloff Metric

Medical people often have different word choices for the same concept. This semantic relationship can be exploited using tools such as WordNet in most cases when the problem is not domain dependent. However, discovering semantic relationship by nature is an open ended problem and highly domain specific. It is often very expensive or impossible to create a comprehensive resource by hand. Thus a corpus based discovery methods is essential to improve the coverage.

The basic intuition in identifying synonyms/hyponyms here is to find phrases which probably convey the same information. We use pre-classified training corpus to identify such phrases first. We then use the discovered synonymous/hyponymous phrases to make better predictions in test data. For this, the first task is to classify some words/phrases into major semantic categories. We used methods explained in section 3.2 to extract such semantic categories in the report. For the synonym identification task, we extracted all the disease/syndrome names mentioned in the report. Then for each co-morbidity, we calculate a riloff value defined in equation 1 for each disease/syndrome which represents the similarity of that disease/syndrome to the co-morbidity. For example, a rilloff value of 0.3820 indicates the degree of similarity of the co-morbidity "chf" with the syndrome "cardiomyopathy".

Riloff value is defined by the equation

$$Riloff\ value = \frac{R}{I}\log R \text{ .... (i)}$$

Here,
R is the number of occurrence of given syndrome when given co-morbidity is present
I is the number of occurrence of given syndrome when the corresponding co-morbidity is not present.

The same concept can also be used to study the interdependence of co-morbidities. High Riloff value of co-morbidity for another indicates that the co-morbidity is causing the former co-morbidity to occur.

3.4 Negation Handling

Prediction in medical domain cannot be accurate without consideration of negation words. Negative qualifier assigned to a medical condition may indicate the absence of the condition, so the ability to reliably identify the negation status of medical concepts affects the quality of results produced by the classification system. Let us consider a simple example sentence "the patient does not have asthma". In this sentence, if the word 'not' is not given a special attention and just considered as just another feature, the semantics of the sentence could be the exact opposite. We use simplified version of NegEx to handle negation effect in our system. We use the negation module in extraction based approach. Our hypothesis here is that if one of the text fragments is negated, the concept is reversed, but if both are negated the decision is retained (double-negation), and so forth.

4 System Framework

The framework broadly consists of traditional supervised classification system and the extraction based classification system. In the supervised classification method, initially, the corpus is preprocessed to extract important features. Preprocessing includes tokenizing, stemming, stopwords removal etc. The features for classification are represented as unigram words and bigram phrases for baseline classification. Document frequency based feature selection strategy is also applied to select important features from the feature set. Document frequency is defined as the count of number of documents in which given feature occurs. The feature weighting scheme used is the tf-idf value. The weight w for each feature in a document is calculated using the formula:

$$weight = tf \times idf = tf \times \log(\frac{N}{df}) \text{ ....(ii)}$$

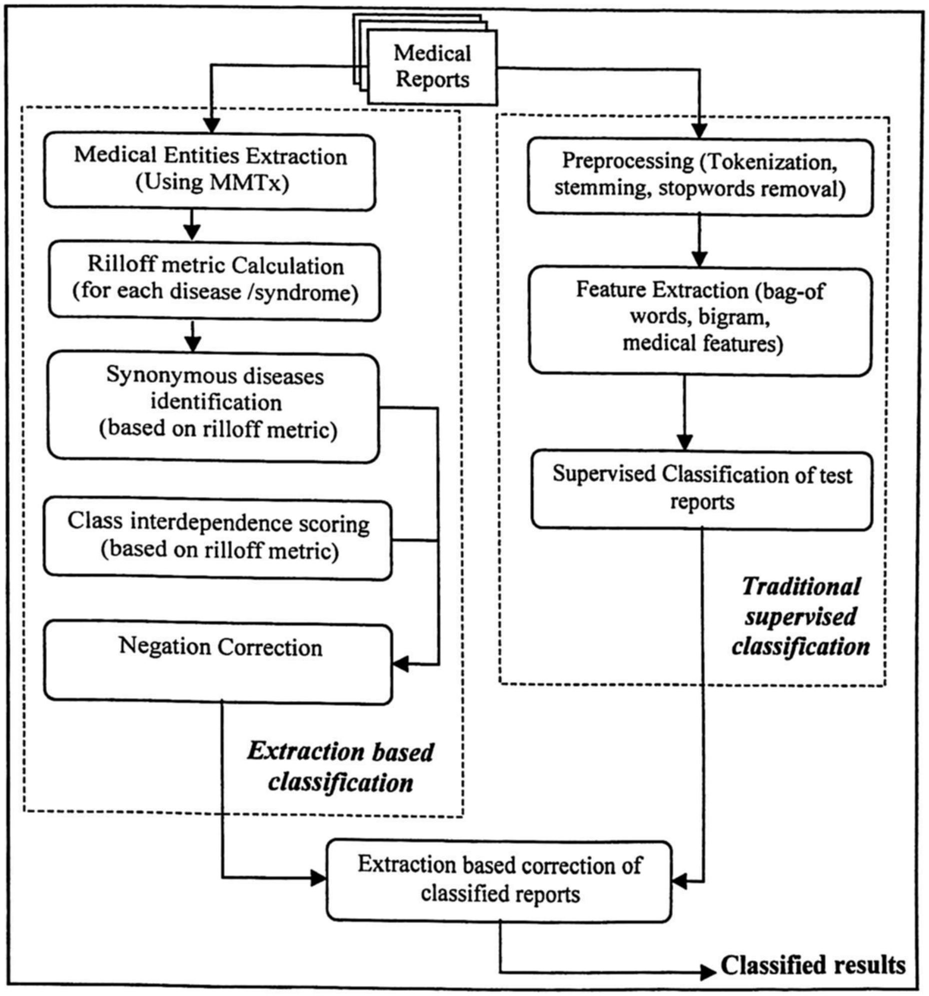

Medical Reports
Medical Entities Extraction (Using MMTx)
Rilloff metric Calculation (for each disease /syndrome)
Synonymous diseases identification (based on rilloff metric)
Class interdependence scoring (based on rilloff metric)
Negation Correction
Extraction based classification
Preprocessing (Tokenization, stemming, stopwords removal)
Feature Extraction (bag-of words, bigram, medical features)
Supervised Classification of test reports
Traditional supervised classification
Extraction based correction of classified reports
Classified results

Figure 1. System framework for a combination of both statistical and extraction based classification of medical reports

where *tf* is the term frequency(number of occurrences of the feature in the document) of the feature in the corresponding document, N is the total corpus size and *idf* is the document frequency of the term. For more advanced classification, medical phrases such as disease/syndromes, sign/symptoms, body parts and clinical drugs are also used as features. The same *tf-idf* feature weighting scheme is used for these features too. We used Java based Weka[22] API to implement and evaluated various machine learning algorithms in the work.

For the extraction based classification, initially, medical features are extracted using MetaMap MMTx. The detailed process is explained in section 3.2. Then for each identified disease/syndrome, Riloff value is calculated. The Riloff value is normalized with the corpus size to get uniform Riloff value. Based on Riloff value, synonymous terms are extracted. For this work, we have considered all the disease/syndromes with a Riloff value greater than 0.3 to be synonymous syndromes. After extracting all synonymous terms, the sentences containing those words are extracted from the reports. These sentences are then checked for negation terms. If the term is negated, it is ignored. If not, the report is considered to have the co-morbidity. The positive cases identified in this process are then used to refine the results obtained from the traditional classification approach.

5 Experimental Observations and Results

We summarize our observations and results in this section. For the evaluation of the supervised algorithms, extraction based synonym/hyponym identification and negation handling; we used 611 narrative medical reports for training the system and 119 reports for testing. The initial unique feature set size was 185527. Features with document frequency greater than 9 were only selected for classification purpose. The final feature set size was 6650.

5.1 Dataset

The dataset used is the de-identified discharge summaries of patients obtained from different healthcare organizations for the obesity challenge. Each document is marked as present, absent, questionable or unmentioned with respect to every co-morbidity and obesity. For each document, both the textual judgments (what the text explicitly states) and intuitive judgments (what the text implies) are provided. Altogether, there are sixteen co-morbidities namely; Obesity, Diabetes mellitus (DM), Hypercholesterolemia, Hypertriglyceridemia, Hypertension (HTN), Atherosclerotic CV disease (CAD), Heart failure (CHF) , Peripheral vascular disease (PVD), Venous insufficiency , Osteoarthritis (OA), Obstructive sleep apnea (OSA), Asthma, GERD, Gallstones / Cholecystectomy, Depression and Gout. There are reports in the dataset.

5.2 Synonym/hyponym Extraction

Table 1 shows some of the synonymous/hyponymous words extracted using the Riloff metric from the medical reports. The Riloff value indicates the degree of similarity of a co-morbidity with the synonym set. The Riloff value for each co-morbidity has different scale as this value depends on the corpus size which is relevant for the co-morbidity.

5.3 Classification Results (Accuracy)

Here, we compare the performance of Naïve Bayes' classification, Support Vector Machine (SVM) classification, J48 decision tree based classification and our system which incorporates the combination of J48 and extraction based classification.

Table 1. Extraction of hyponyms for each co-morbidity based on Riloff metric

Co-morbidity	Synonym set	Riloff val
gout	Abdominal aortic aneurysm	0.44
	Chronic obstructive pulmonary disease	0.61
Diabetes	Nph	0.42
hypercholesterolemia	Hyperlipidemia	0.30
	Elevated cholesterol	0.34
obesity	Fibromyalgia	0.31
	Obese	0.82
depression	chronic pain	0.51
	Migraines	0.45
gerd	gastroesophageal reflux	1.01
	Fibromyalgia	0.33
pvd	peripheral vascular disease	0.67
	Vascular disease	0.66
asthma	Asthma flare	0.8
	Tracheobronchitis	0.48
cad	coronary artery disease	0.49
chf	Congestive heart failure	0.49
	Heart failure	0.79
	Ischemic cardiomyopathy	0.40
osa	obstructive sleep apnea	0.81
	Sleep annea	1.01
	Pulmonary hypertension	0.59
oa	Djd	0.922
	Arthritis	0.67
	Fibromyalgia	0.59
	Degenerative joint disease	0.69
	Osteoarthritis	1.38
Hypertension	Htn	0.54

The graph in figure 2 summarizes the performance of the above mentioned algorithms. Naïve Bayes' algorithm does not show good result in predicting the co-morbidities although it attains accuracy over 90% in the case of hyperglyceridemia. Support vector machine performs relatively better than Naïve Bayes'. However, J48 decision tree performs the best in predicting obesity and its co-morbidities. J48 algorithm has exhibited accuracy over 90% for almost all the co-morbidities, some being nearly perfect. The refinement of the J48 results with the extraction based results has given even better accuracy of although not of significant value.

Similarly, figure 3 below shows the performance of different algorithms on the intuitive judgment. The overall accuracy of all the algorithms is worse than for the textual judgment. Among all the algorithms, Naïve Bayes' performed the worst with accuracy less than 70% for all the co-morbidities.

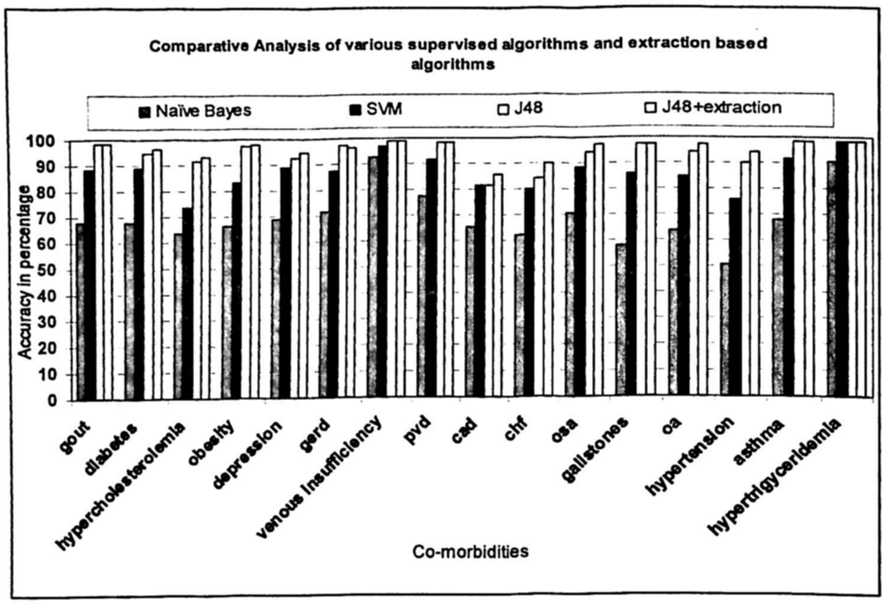


Figure 2. Accuracy of Various algorithms and combination of J48 and extraction based algorithm for co-morbidities classification based on textual judgment

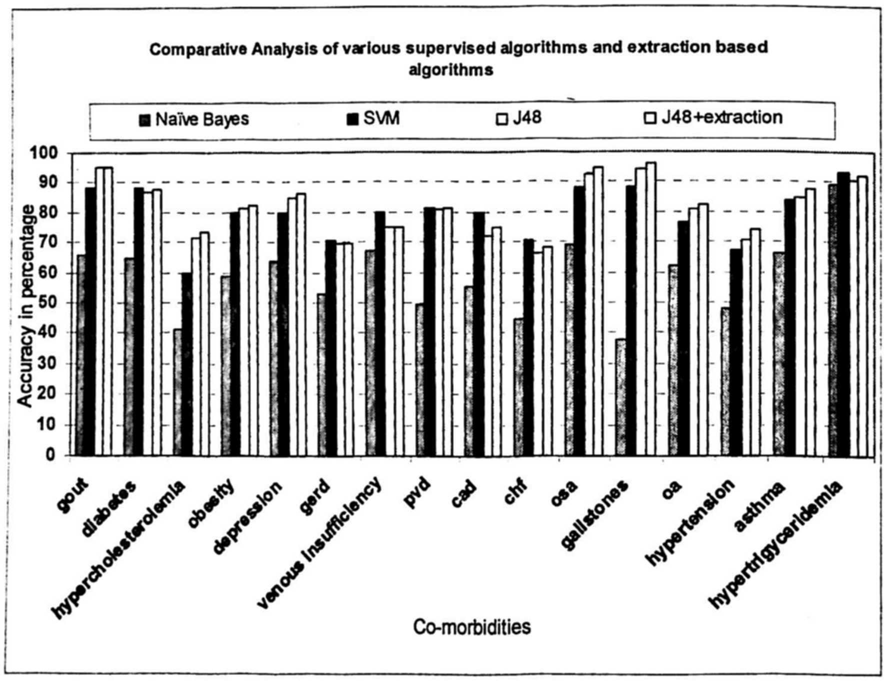


Figure 3. Accuracy of Various algorithms and combination of J48 and extraction based algorithm for co-morbidities classification based on Intuitive judgment

Support Vector Machine (SVM) based classification and J48 decision tree based classification shows comparable and relatively better results. In some co-morbidity, SVM classifier even outperforms J48 and combination of J48 and extraction based classifier.

6 Conclusion

We have explored various problems associated with free text processing of narrative medical reports and have also presented approaches to address some of those. We explored the semantic aspects of medical reports such as synonymy/hyponymy extraction, negation handling and multi-classes interdependence. The experimental results show that this method does help in increasing the accuracy of the results obtained from statistical algorithms. The identification of synonymous/hyponymous words can help not only in document classification, but can be important many other applications. One of the advantages of automated hyponyms extraction to thesauri based extraction is that the technique can be ported to any domain easily and is very domain dependent capturing the best of the context. Another advantage is that it can retain the semantics of the context without a full document analysis (works just by extracting relevant part of the document). This makes it computationally less expensive too. As a future work, we intend to explore other semantic aspects such as temporality, experiencer etc to make better predictions.

References

1. Informatics for Integrating Biology and the Bedside, https://www.i2b2.org/NLP/Main.php
2. Chapman, W.W., Bridewell, W., Hanbury, P., Cooper, G.F., Buchanan, B.G., A Simple Algorithm for Identifying Negated Findings and Diseases in Discharge Summaries. Journal of Biomedical Informatics 34, 301--310 (2001)
3. Aronow, D.B., Feng, F., Croft, W.B., Ad Hoc Classification of Radiology Reports. Journal of the American Medical Informatics Association, pp. 393-411 (1999)
4. Lu, C., Probabilistic and machine learning approaches to medical classification problems, ph.d. dissertation (2005)
5. Riloff, E., Lehnert, W., Information Extraction as a Basis for High-Precision Text Classification, ACM Transactions on Information Systems (1994)
6. Bishop, C. M., Neural Networks for Pattern Recognition. Oxford University Press, Oxford (1995)
7. Averbuch, M., Karson, T., Ben-Ami, B., Maimon, O., Rokach, L. Context-Sensitive Medical Information Retrieval. in Proc. of 11th World Congress on Medical Informatics (MEDINFO-2004). San Francisco, CA: IOS Press (2004)
8. Goryachev, S., Sordo, M., Zeng, Q. T., Ngo, L., Implementation and Evaluation of four different methods of Negation Detection (2006)
9. Gonzalo J, Verdejo F, Chugur I, Cigarran J: Indexing with WordNet synsets can improve Text Retrieval. In Proceedings of the COLING/ACL '98 Workshop on Usage of WordNet for NLP Montreal, Canada (1998)
10. WordNet: A Lexical Reference System and its Application. MIT Press, Cambridge, MA., pages 265—283 (1998)

11. Pedersen, T., Patwardhan, S., and Michelizzi, J. "WordNet::Similarity - Measuring the Relatedness of Concepts" In Proceedings of Fifth Annual Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL-2004), pp. 38-41. Boston (2004)
12. Snow, R., Jurafsky, D., Ng, A., Learning syntactic patterns for automatic hypernym discovery. In Proc of Neural Information Processing Systems Vancouver, Canada 1297-1304 (2004)
13. Pereira, F., Tishby, N., Lee, L., Distributional Clustering of English Words. In Proc of ACL-1993 Columbus, Ohio, USA .pp. 183-190 (1993)
14. Yu, H., Hatzivassiloglou, V., Friedman, C., Rzhetsky, A., Wilbur WJ: Automatic extraction of gene and protein synonyms from MEDLINE and journal articles, Proc AMIA Symp .pp. 919-923 (2002)
15. Morin, E., Jacquemin, C., Automatic Acquisition and Expansion of Hypernym Links. Computers and the Humanities 363-396 (2004)
16. Dumais, S., Furnais, G., Landauer, T., Indexing by Latent Semantic Analysis, American Society for Information Science, pp. 391-407 (1990)
17. McCrae, J., Collier, N., Synonym set extraction from the biomedical literature by lexical pattern discovery, BMC Bioinformatics (2008)
18. http://nlp.stanford.edu/IR-book/html/htmledition/classification-with-more-than-two-classes-1.html
19. Wikipedia. Information Extraction http://en.wikipedia.org/wiki/Information_extraction
20. MetaMap transfer (MMTx) mmtx.nlm.nih.gov/
21. Unified Medical Language System (UMLS) www.nlm.nih.gov/research/umls/
22. Weka, Data mining with open source machine learning software www.cs.waikato.ac.nz/ml/weka/

Analysis of Stemming Alternatives and Dependency Pattern Support in Text Classification

Levent Özgür and Tunga Güngör

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

Abstract. In this paper, we study text classification algorithms by utilizing two concepts from Information Extraction discipline; dependency patterns and stemmer analysis. To the best of our knowledge, this is the first study to fully explore all possible dependency patterns during the formation of the solution vector in the Text Categorization problem. The benchmark of the classical approach in text classification is improved by the proposed method of pattern utilization. The test results show that support of four patterns achieves the highest ranks, namely, *participle modifier*, *adverbal clause modifier*, *conjunctive* and *possession modifier*. For the stemming process, we benefit from both morphological and syntactic stemming tools, Porter stemmer and Stanford Stemmer, respectively. One of the main contributions of this paper is its approach in stemmer utilization. Stemming is performed not only for the words but also for all the extracted pattern couples in the texts. Porter stemming is observed to be the optimal stemmer for all words while the raw form without stemming slightly outperforms the other approaches in pattern stemming. For the implementation of our algorithm, two formal datasets, Reuters - 21578 and National Science Foundation Abstracts, are used.

Key words: Text Classification, Dependency Patterns, Stemmer Analysis, Information Extraction

1 Introduction

Text Classification (TC) is a learning task, where pre-defined category labels are assigned to documents based on the likelihood suggested by a training set of labelled documents.

Most of the approaches used in this problem study it in bag-of-words (bow) form, where only the words in the text are analyzed by some machine learning algorithms for TC [1]. In this approach, documents are represented by the widely used vector-space model, introduced by Salton et al. [2]. In this model, each document is represented as a vector *d*. Each dimension in the vector *d* stands for a distinct term (word) in the term space of the document collection.

Received 17/11/08
Accepted 15/12/08
Final version 03/02/09

Classical bow approach is solely based on the words of the sentence without any further study on the implicit concepts behind them. Although the success scores are found to be satisfactory in most studies, this approach may not be so adequate for some complex cases. Dealing with semantic similarity and concepts is a critical and challenging subject in TC.

WordNet is the most experienced lexical tool in text related studies [3]. Briefly, WordNet introduces *synset* concept which corresponds to *synonym set.* Basic studies utilizing WordNet in TC have not yielded outstanding results because of mainly the disambiguation problem [4] [5]. There are also positive results but with specific exceptions, like manual disambiguation [6]. On the other hand, recent boosting algorithms have yielded successful results [7]. Also supplementary packages of WordNet (i.e: QueryData, Similarity Package) have been utilized recently which have been stated to improve the performance but by also increasing the complexity of the solution [8].

Almost all of these approaches lack the fact that the meaning of a sentence may not always be explicitly presented within the words; the sentence may contain implicit facts that can only be sensed through a deep analysis of the whole sentence by examining its syntactic and semantic structure. In order to fill this gap, we benefit from Information Extraction (IE) discipline, which aims to extract structured information from unstructured machine-readable documents. Patterns and sentence dependencies are recent topics in this discipline which we analyze in this study. To make an exploration of this new possible feature, we use 22 different pattern types and enrich our solution vector separately with these new features.

Stemmer analysis in text processing is our other major concern in this paper. In almost all previous studies, we see that morphological stemming, in which stemming is based on only morphological issues that are completely independent from the syntactic and semantic structure of the sentence, is always performed as a standard preprocess operation while forming the solution vector. Both inflections and derivational affixes are removed in this type. Utilizing the stemmed form of the word instead of its raw form may be preferred in straightforward approaches (i.e. bow approach), but our perspective in this paper introduces the dependency couple of words which increases the occurrence of several words, so a reexamination of stemming algorithms in this integrated approach will be the contribution of this paper. Moreover, the effectiveness of the straightforward style of morphological stemming should also be questioned systematically. In this paper, in addition to the morphological stemmers, we also analyze syntactic stemmers. We emphasize the name of this alternative type as syntactic stemming because this type of stemming is performed during the syntactic analysis of the sentence in which POS information and lexical dependencies are analyzed. Different from the morphological parsing, only inflections are removed by keeping the derivational affixes in this type of stemming. For example, the word *arrivals* is stemmed as *arrive* in Porter stemmer while the base form is found as *arrival* in Stanford stemmer by keeping the derivational affix. We utilize both ways of stemming by employing two well-known tools: Porter Stemmer [9] as the morphological stemmer and Stanford Stemmer, the built-in stemmer of Stanford Parser, as the syntactic one. In this paper, we also count the WordNet synset utilization as another alternative stemming style because WordNet usage is also highly related with stemming. Porter stemmer is not compatible with WordNet because the POS information of the raw word is lost due to the removal of the derivational affixes. Stanford stemming is implemented before WordNet utilization and the synset id, which is extracted by WordNet after stemming, is also stated as another stemming type.

The paper is organized as follows: Details of the pattern concept are discussed in Section 2. Our proposed model is covered in Section 3 and test results are given in Section 4. We conclude the paper and propose future work in Section 5.

2 Dependency Patterns

2.1 Related Studies Based on Patterns

A critical problem in IE is to develop systems which can be easily adapted to new domains as automatically and correctly as possible [10]. Solutions to this problem attempt to learn the domain-specific information, named as patterns. Patterns can be structured in many different ways with different levels of linguistic analysis. In a detailed analysis between different pattern structures [11], four different pattern models were analyzed which are predicate-argument model (SVO), chains, linked chains and subtrees. Riloff devised an original algorithm in her remarkable study, which automatically generates significant extraction patterns with noun generalization from untagged texts [12].

Lexical dependency is a different way of representing the structure of the sentences which extracts grammatical relations (*object*, *subject*, *preposition* etc.) between words in a sentence [13]. An increasing interest in using lexical dependency properties for different NLP tasks from machine translation to question answering is observed in the related studies. A recent study has focused on dependency support for text classification which has yielded successful results but with a narrow view utilizing all the dependencies together without a further and detailed analysis [14]. To make an exploration of this new possible feature, we perform a further analysis by studying each pattern specifically as an extension of the standard bow approach which is explained in Section 2.2.

Parser utilization is inevitable for syntactic study in which the phrases, POS information and dependency of the words in the sentences are identified. Our implementation details for this subject are given in Section 3.1.

2.2 Dependency Pattern Utilization

22 grammatical relations are employed in the tests from the list of 48 relations given in [13]. Table 1 shows these relations; including their definitions and some examples. Our selection criterion is highly motivated by their utilization frequencies and also their generalization capacities. Dependencies are eliminated

including number contents (i.e. *num - numeric modifier*). The content of these features are so generic that their contribution will not be meaningful. Some of the similar dependencies are combined in the hierarchy (e.g. *dobj, iobj* and *pobj* as *obj*) in order to sum up their frequencies and discriminative power.

Table 1. Dependency Patterns and Their Examples

Symbol	Pattern Type	Example Couples	Symbol	Pattern Type	Example Couples
subj	subject-verb	they-break	**obj**	object-verb	glass-break
aux	auxiliary auxpassive	expected-are	**conj**	conjunctive	energy-petrochemical
attr	attributive	remain-year	**comp**	complement	decline-disclose
complm	complementizer	is-that, have-that	**mark**	mark	account-while
rel	relative	sell-of	**acomp**	adjectival complement	turn-bad
agent	agent	approve-bank	**adv**	adverbal clause modifier	quickly-open
rel	relative clause modifier	begin-season	**amod**	adjectival modifier	scientific-experience
infmod	infinitival modifier	way-invest	**rcmod**	relative clause modifier	begins-season
app	appositional modifier	monitoring-detection	**nn**	noun compound modifier	source-laser
poss	possession modifier	Asia-nations	**prt**	phrasal verb participle	cover-up
part	participle modifier	costs-related	**prep**	prepositional modifier	focus-research

3 Proposed Model

3.1 Modules

Syntactic Tool Stanford Parser is known to be the most powerful and efficient parser in the subject. In our tests with this parser, the parser is observed to avert syntactic ambiguities. In the most recent version, its structure gives only the first probable parse as the result. It is compared with two other systems and rated as the parser with the least error rate [11]. It has an integrated capability of extracting both the POS information and the dependencies between the words in a sentence. PCFG parser mode is selected in our implementation which extracts this information directly instead of a factorized solution [15].

Lexical Tool WordNet, as explained above, is our lexical database in this study. WordNet version 2.0 is utilized for extraction of the synsets of pattern couples in the texts.

Data mining Tool Support Vector Machine (SVM) is the data mining module for the main classification part. Recent studies have compared the performance of various classification algorithms including SVM with linear kernel, SVM with polynomial kernel of various degrees, SVM with RBF kernel with different variances, k-nearest neighbor algorithm and Naive Bayes [1]. In these experiments, SVM with linear kernel was consistently the best performer. These results confirm the results of the previous studies by Yang and Liu [16], Joachims [17] and Forman [18]. Thus, in this study we prefer SVM with linear kernel as the classification technique which is supposed to give best results standalone in recent comparable studies [18, 19]. For our experiments we used the SVMlight system, which is a rather efficient implementation by Joachims [19] and has been commonly used in previous studies. Classification of SVM is performed as one-versus-all for all dataset topics [18].

For the preprocessing of the datasets; each document is parsed, non-alphabetic characters and mark-up tags are discarded, case-folding is performed, and stopwords are eliminated. We utilize the list of 571 stopwords used in the Smart system [2]. For the term weighting approach, tf-idf technique is selected. The comparative study of different term weighting approaches in text retrieval have concluded that the commonly used tf-idf weighting outperforms other types [20]. Each document vector is normalized to account for documents of different lengths [1].

3.2 Dataset Selection

UCI Machine Learning Repository is inquired and standard Reuters - 21578 (Reuters) and National Science Foundation Research Award Abstracts (NSF) datasets are selected for our study [21]. The reason for this selection is that both datasets hold sentence information and the style of the texts are formal which has ordered sentences. These properties are crucial for efficient parsing in our IE approach for TC.

Reuters is a well-known dataset, which has been used for many TC algorithms [1, 16]. Standard ModApte split is used in which there are 9,603 training documents and 3,299 test documents. All the topics that exist both in the training and the test sets are utilized in the experiments. Our dataset thus consists of 90 classes and is highly skewed. For example, most of the classes have less than ten documents while seven classes have only one document in the training set. Also it allows multiple topics, which mean that documents in the corpus may belong to more than one existing topic.

NSF dataset consists of 129,000 abstracts describing NSF awards for basic research between the years 1990 and 2003 [21]. Due to this huge size, year 2001 is selected randomly and five sections (four sections for training and one section for test) are picked out from this year. Totally, there are 2368 training documents and 610 test documents with 122 existing topics. NSF is also skewed and allows multiple topics like Reuters.

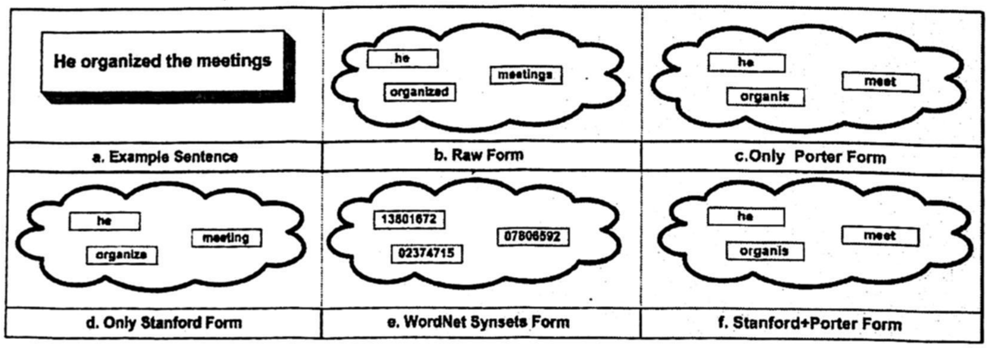
3.3 Test Scenarios

Our main goal in this paper is to compare the supplementary benefit of all possible dependencies and also analyze the optimal stemming algorithm for both raw words and dependency couples existing in the documents. For this purpose, we devise a two-stage analysis for our problem. We name the first and second stages as AllWords Analysis and AllWordsPlus Analysis, respectively.

Five distinctive stemming styles are employed according to stemmer choice and WordNet utilization for both stages to be used in both AllWords Analysis and AllWordsPlus Analysis:

1. Raw Form: No stemming process is implemented for the classification algorithm and the words are used in their raw forms.
2. Only Porter Form: Morphological Porter stemmer is used for stemming.
3. Only Stanford Form: Syntactic Stanford stemmer is used for stemming.
4. WordNet Synsets Form: After stemming by the Stanford stemmer, WordNet is employed to extract the synset variations of all the words. Porter stemmer is not implemented as an alternative because output of this stemmer is not compatible for WordNet integration for two basic reasons. First, we need the correct POS information and the root form for our semantic analysis but this stemmer does not conserve the POS information of the derived word by extracting the possible shortest base form; and second, the outcome of this stemmer is not always in the standard base forms (i.e. *earli* instead of *early*, *continu* instead of *continue*, etc.) required for the WordNet interface.
5. Stanford+Porter Form: Stanford stemmer is implemented initially for inflection removal, then Porter stemmer is used for the removal of derivational affixes of the same word.

AllWords Analysis In this stage, classical bow approach is implemented with the above stemming variations in order to find the optimal strategy for the bow approach. An example sentence is represented in Fig. 1.a. According to our classification in the previous paragraph, Fig. 1.b shows *Raw Form* for the example sentence without any stemming process. *Only Porter Form* and *Only Stanford Form* are represented in Fig. 1.c and Fig. 1.d. Fig. 1.e represents *WordNet Synsets Form* while *Stanford+Porter Form* is shown in Fig. 1.f.



a. Example Sentence

b. Raw Form

c.Only Porter Form

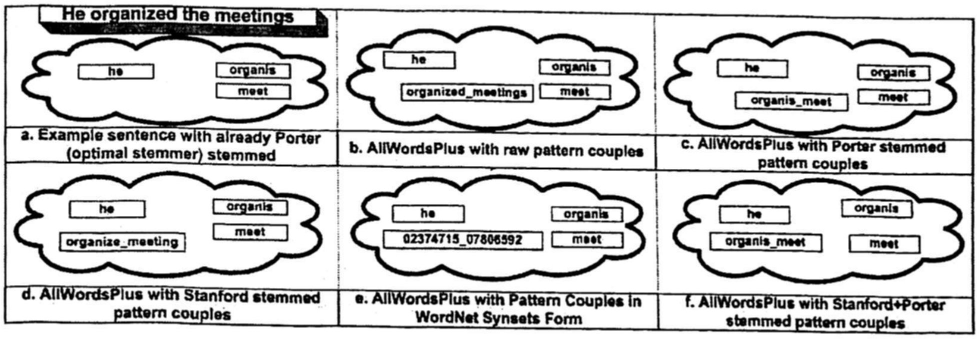
d. Only Stanford Form

e. WordNet Synsets Form

f. Stanford+Porter Form

Fig. 1. Sample Keyword Formations due to Stemming Alternatives for AllWords Approach

AllWordsPlus Analysis In the second stage, we perform the re-examination of stemming alternatives; this time, not for the words but for the pattern couples. From another related perspective, we extend the bow approach by including the pattern couples which are varied by the stemming alternatives as shown in Fig. 2. For the required format of the bow approach for all words, we use the optimal solution which can be extracted from AllWords Analysis as seen in Fig. 2.a. According to our classification of stemming alternatives, Fig. 2.b shows *Raw Form* utilization for the patterns in addition to the optimally stemmed words for the example sentence. *Only Porter Form* and *Only Stanford Form* are used for pattern stemming as shown in Fig. 2.c and Fig. 2.d, respectively. Fig. 2.e represents *WordNet Synsets Form* while *Stanford+Porter Form* is shown in Fig. 2.f. Briefly, we use the optimally preprocessed *allwords* in the documents as the base keyword features for our algorithm and extend it with the pattern variations in each alternative implementation.



a. Example sentence with already Porter (optimal stemmer) stemmed

b. AllWordsPlus with raw pattern couples

c. AllWordsPlus with Porter stemmed pattern couples

d. AllWordsPlus with Stanford stemmed pattern couples

e. AllWordsPlus with Pattern Couples in WordNet Synsets Form

f. AllWordsPlus with Stanford+Porter stemmed pattern couples

Fig. 2. Sample Keyword Formations due to Stemming Alternatives for Patterns in AllWordsPlus Approach

4 Test Results

4.1 Evaluation Metrics

In order to evaluate the performance of our implementation we use the commonly used F-measure metric, which is equal to the harmonic mean of recall and precision [1]. F-measure score can be computed by two different alternatives, Micro-averaged F-Measure (MicroF) and Macro-averaged F-Measure (MacroF). MicroF gives equal weight to each document and is therefore considered as an average over all the document/category pairs while MacroF gives equal weight to each category so it is influenced more by the classifier's performance on rare categories. Keyword number (Key#) is another performance criterion which is the number of selected features for the solution vector of SVM.

4.2 Results and Discussion

Table 2. All Words Stemming

	Reuters			NSF		
Approach	Key#	MicroF	MacroF	Key#	MicroF	MacroF
Raw	27094	85.63	43.86	21632	61.41	46.75
Only Porter	20292	85.58	43.83	14878	61.74	47.34
Only Stanford	23094	80.88	45.57	18062	61.21	46.02
WordNet Synsets	25202	80.62	44.87	21510	58.73	44.44
Stanford+Porter	18253	80.75	45.29	14186	61.74	47.42

AllWords Analysis The results for Allwords analysis is shown in Table 2. As can be seen in the table, morphological stemming of Porter Stemmer is found to be the optimal approach (*Raw* form outperforms it in Reuters but far behind it in NSF with also much more keywords) for all words stemming in both datasets, with low keyword numbers and high success rates. So, Porter stemmer is selected for stemming process of the words in our dataset for the next step.

Stanford stemmer, which is a more complicated syntactic parser, has mainly lower success rates with much more keyword numbers with respect to Porter stemmer. The main reason for this difference is that the outcomes of the stemmers are different in many cases. Stanford stemmer comes out with many different forms for the same base form of the word because it only removes inflection, conserving the derivational affixes. For example, for the words *arrivals* and *arrived*, Stanford stemmer finds the base forms as *arrival* and *arrive*, respectively. On the other hand, Porter stemmer finds the same base form *arrive* for both words by removing both the derivational affixes and inflections.

Table 3. Pattern Stemming

	Reuters			NSF		
Approach	Key#	MicroF	MacroF	Key#	MicroF	MacroF
Raw	27387	85.60	43.90	19534	61.91	47.21
Only Porter	27152	85.62	43.87	20631	61.90	47.15
Only Stanford	25561	85.60	43.85	19856	61.91	47.19
WordNet Synsets	26184	85.60	43.83	19892	61.92	47.20
Stanford+Porter	26693	85.61	43.82	20487	61.88	47.20

AllWordsPlus Analysis For this approach, we can analyze the results from two points of view. First, we compare all the stemming approaches in our pattern utilized solution. We calculate the average of all pattern utilized results for each approach in both datasets. The comparison values are reported in Table 3. We see that, differences between stemming approaches is low, in terms of both keyword number and success rate, when compared with AllWords stemming alternatives summarized in Table 2. A possible reason for this low difference is the fact that pattern utilization integrates the related words in only certain forms to form couples, which decreases the role of stemming. In parallel to this idea, we can say that *raw* stemming process, which is the simplest form without any stemming process, is slightly better than the others in both datasets.

Table 4. Pattern Performance Ranks in Descending Order for Reuters and NSF

	Reuters	Key#	MicroF	MacroF		NSF	Key#	MicroF	MacroF
Rn	Patterns	avg	avg+/-std	avg±std	Rn	Patterns	avg	avg±std	avg±std
1	Part	25937	85,71±0,01	44,08±0,05	1	Adv	21112	62,22±0,11	47,51±0,01
2	Subj	29721	85,68±0,07	44.10±0,30	2	Comp	19975	62,24±0,12	47,49±0,04
3	Adv	28425	85,71±0,03	44,04±0,10	3	Cls	16227	62,00±0,04	47,53±0,04
4	Conj	32026	85,64±0,12	44,03±0,07	4	Part	18255	62,07±0,06	47,41±0,01
5	Poss	27401	85,68±0,03	43,89±0,04	5	Poss	16186	61,95±0,08	47,52±0,01
6	Amod	31690	85,66±0,06	43,91±0,12	6	Mark	15672	61,89±0,02	47,48±0,03
7	Rcmod	26673	85,60±0,02	43,94±0,04	7	Conj	29144	62,00±0,09	47,35±0,20
8	Agent	21603	85,63±0,01	43,91±0,04	8	Complm	15386	61,83±0,03	47,46±0,01
9	App	24676	85,61±0,02	43,91±0,02	9	App	15993	61,83±0,03	47,41±0,00
10	Comp	34414	85,73±0,02	43,78±0,13	10	Prt	15018	61,88±0,05	47,35±0,01
11	Obj	34907	85,67±0,09	43,79±0,08	11	Rcmod	18365	61,81±0,03	47,36±0,00
12	Acomp	20705	85,59±0,01	43,86±0,00	12	Infmod	15417	61,81±0,00	47,34±0,00
13	Attr	20378	85,58±0,00	43,83±0,00	13	Rel	16250	61,77±0,10	47,35±0,02
14	Cls	24202	85,55±0,01	43,86±0,01	14	Agent	15529	61,77±0,04	47,33±0,01
–	**Benchmark**	**20292**	**85,58±0,00**	**43,83±0,00**	15	Attr	14892	61,74±0,00	47,34±0,00
15	Complm	21442	85,57±0,01	43,83±0.02	–	**Benchmark**	**14878**	**61,74±0,00**	**47,34±0,00**
16	Prt	21122	85,55±0,03	43,84±0,03	16	Acomp	15035	61,67±0,00	47,36±0,04
17	Infmod	21922	85,54±0,01	43,82±0,00	17	Amod	27535	62,13±0,16	46,83±0,06
18	Mark	22638	85,53±0,01	43,82±0,01	18	Obj	27737	61,88±0,13	47,06±0,33
19	Rel	20977	85,52±0,05	43,80±0,05	19	Subj	29355	62,28±0,17	46,49±0,02
20	Prep	33487	85,71±0,05	43,55±0,16	20	Prep	31546	62,15±0,22	46,35±0,33
21	Nn	31220	85,45±0,07	43,66±0,04	21	Nn	27726	61,79±0,09	46,23±0,06
22	Aux	29528	85,45±0,04	43,50±0,27	22	Aux	19390	61,19±0,17	46,60±0,44

In our second analysis, we compare the pattern utilization performance by analyzing the results through all stemming modes. Patterns are ranked (*Rn*) according to their MicroF and MacroF average scores (*avg*) with standard deviations (*std*) according to the alternative stemming modes in Table 4. As can be seen from the table, 14 patterns in Reuters and 15 patterns in NSF out of the possible 22 pattern types, have the power of outperforming the benchmark. The critical point is that, nine positive patterns achieve this performance in both datasets consistently with also very low standard deviations. Out of these nine *positive* patterns, we focus on the highest four patterns, namely : *part-participle modifier*, *adv-adverbal clause modifier*, *conj-conjunctive* and *poss-possession modifier.*

Fig. 3 shows the incremental effect of these patterns in Reuters and NSF. This improvement is shown by the dark colored part over the white color of the benchmark score for each pattern in the figure. These pattern types, when utilized standalone, improve the benchmark by around 0.4%-0.5%. *Part* pattern is the integration of a participle which is a derivative of a non-finite verb and the word modified by this participle (e.g. They compared it with *estimates derived* from ...). *Adv* pattern integrates the predicate with the supplementary word in the adverbial clause (e.g. We *must quickly* open our markets). *Conj* pattern relates the words which are connected by conjuncts (e.g. *Businessmen* and *officials* said that ...) while *poss* pattern relates the words by the possession feature (e.g. Japan has raised fears among many of *Asia's* exporting *nations...*). According to our observations, the common property of these patterns seem that they contain, not usually the word couples belonging to the closed classes (e.g. preposition, conjunction, etc.), but mostly the open class (e.g. noun, verb, etc.) words, which hold mutual characteristic relations within the couple. Another common feature is their frequencies which are adequate when compared with the number of all words in the datasets. For example, there are approximately 14,000 *conj*, 6,500 *adv*, 3,500 *part* and 1500 *poss* distinct patterns to be integrated with about 15,000 words in NSF. On the other hand, for example, *Attr* pattern with only 15 keywords performs ineffectively when integrated with all the words in this dataset.

There is also consistency for the three most unsuccessful pattern types in both Reuters and NSF: *preposition modifier - prep*, *noun compound modifier - nn*, and *auxiliaries-aux*. *Aux* pattern gives the worst results for both datasets in both MicroF and MacroF values. The main factor for this failure seems to be the fact that the relationship within the pattern structure is generic (e.g. : make-is, running-are) which is not feasible for the classification problem. *Nn* pattern is the second worst pattern with low scores in both MicroF and MacroF measures. *Prep* pattern is an interesting pattern with leading scores in MicroF but lowest results with MacroF in both datasets.

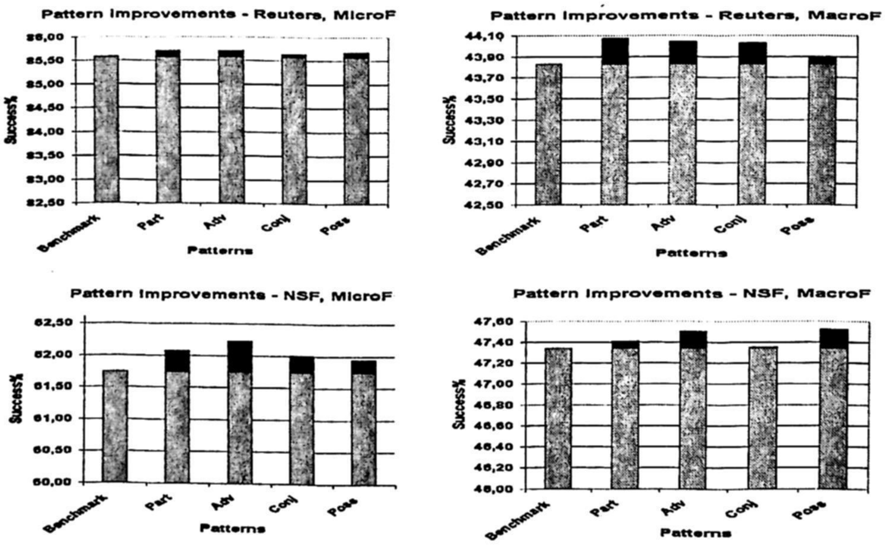


Fig. 3. Improvements of Successful Patterns Over the Benchmark

4.3 Hardware Specifications and Time Complexities

All experiments were implemented in Hp Workstation xw6200 with Xeon CPU 3.2 GHz and 4 GByte RAM.

Dataset parsing is the most time consuming part of the overall process which takes more than 10 hours for both datasets. However, this parsing operation is performed only once to be utilized for all the test modes for that dataset.

For a single test mode, the most time consuming part is the creation of tf-idf values for all existing terms in the training and test phases. This process takes approximately 90 minutes with about 30,000 keywords.

Due to memory and time limitations, maximum keyword number was limited with 36,000.

5 Conclusion and Future Work

To the best of our knowledge, this is the first study to fully explore all possible dependency patterns during the formation of the solution vector in the TC problem. The benchmark of the classical approach in TC is improved by the support of pattern utilization and further analysis can be performed to increase the improvement. In this performance, support of four patterns achieve the highest ranks, namely: *participle modifier*, *adverbal clause modifier*, *conjunctive* and *possession modifier* patterns. These pattern types improve the benchmark by around 0.4%-0.5%. There is also consistency for the three most unsuccessful pattern types in both Reuters and NSF. We have the motivation to analyze the syntactic and semantic features of both successful and unsuccessful patterns in more detail in order to utilize them more efficiently in our problem to increase the improvement.

Another contribution of this paper is its approach in stemmer utilization. Stemming is performed not only for the words but also for all the extracted pattern couples in the texts. Porter stemming is observed to be the optimal stemmer for all words while the raw form without stemming slightly outperforms the other approaches in pattern stemming.

We are planning to examine the use of selected keywords instead of all the words in the dataset for this problem. We consider to use the same distinction for keyword selection as the stemming approach in this study: keywords of all words and keywords of patterns. Selecting keywords separately from these groups and then combining them may yield better performance in terms of accuracy and time.

Acknowledgements. This work has been supported by the Boğaziçi University Research Fund under the grant number 05A103D.

References

1. Ozgur, A.: Supervised and Unsupervised Machine Learning Techniques for Text Document Categorization. MS Thesis. Bogazici University (2004)
2. Salton, G., Yang C.S., Wong A.: A Vector-Space Model for Automatic Indexing. Communications of the ACM 18 no.11 (1975)

3. Miller, G.: WordNet: a lexical database for English. Communications of the ACM, Volume 38, Issue 11 , pp.39-41 (1995)
4. Mansuy, T., Hilderman, R.: A Characterization of WordNet Features in Boolean Models for Text Classification. In: Proceedings of the 5th Australasian Data Mining Conference (AusDM'06), Sydney, Australia, November, pp. 103-109 (2006)
5. Moschitti, A., Basili, R.: Complex Linguistic Features for Text Classification. In: A Comprehensive Study. ECIR 2004. pp. 181-196 (2004)
6. Hidalgo, J.M.G, Rodriguez, M.B: Integrating a Lexical Database and a Training Collection for Text Categorization. In: ACL/EACL Workshop on Automatic Extraction and Building of Lexical Semantic Resources for Natural Language Applications (1997)
7. Bloehdorn, S., Hotho, A.: Boosting for text classification with semantic features. In: Proceedings of the MSW 2004 workshop at the 10th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 70-87 (2004)
8. Bloehdorn, S., Moschitti, A.: Combined Syntactic and Semantic Kernels for Text Classification. In: ECIR 2007:307-318 (2007)
9. Porter, M.: An Algorithm for Suffix Stripping. In: Program 14, 130–137 (1980)
10. Stevenson, M., Greenwood, M.: A Semantic Approach to IE Pattern Induction. In: Proceedings of the 43rd Annual Meeting of the ACL, Ann Arbor (2005)
11. Stevenson, M., Greenwood, M.: Comparing Information Extraction Pattern Models. In: Proceedings of the Workshop on Information Extraction Beyond the Document, pp. 12-19, Sydney (2006)
12. Riloff, E.: Automatically Generating Extraction Patterns from Untagged Text. In: Proceedings of the 13th National Conference on AI - AAAI-96, pp. 1044-1049 (1996)
13. Marneffe, M.C., MacCartney, B., Manning, C.: Generating Typed Dependency Parses From Phrase Structure Parses. In: LREC2006 (2006)
14. Nastase, V., Shirabad, J.S.,Caropreso, M.F.: Using Dependency Relations for Text Classification. In: AI 2006, the nineteenth Canadian Conference on Artificial Intelligence, Québec City, Quebec, Canada (2006)
15. Klein, D., Manning, C.: Fast Exact Inference with a Factored Model for Natural Language Parsing. NIPS, volume 15. MIT Press (2003)
16. Yang, Y., Liu, X.: A Re-examination of Text Categorization Methods. Proceedings of SIGIR-99. In: 22nd ACM International Conference on Research and Development in Information Retrieval, Berkeley (1999)
17. Joachims, T.: Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In: European Conference on Machine Learning (ECML) (1998)
18. Forman, G.: An Extensive Empirical Study of Feature Selection Metrics for Text Classification. Journal of Machine Learning Research 3, 1289–1305 (2003)
19. Joachims, T.: Advances in Kernel Methods-Support Vector Learning. chapter Making Large-Scale SVM Learning Practical. MIT-Press (1999)
20. Salton, G., Buckley C: Term Weighting Approaches in Automatic Text Retrieval. Information Processing and Management 24 no. 5, 513–523 (1988)
21. Asuncion, A., Newman, D.: UCI Machine Learning Repository. http://www.ics.uci.edu/~mlearn/MLRepository.html. Irvine, CA: University of California, School of Information and Computer Science (2007)

Investigating Variations in Adjective Use across Different Text Categories

Jing Cao[1] and Alex Chengyu Fang[2]

Department of Chinese, Translation and Linguistics
City University of Hong Kong
Hong Kong SAR, China
[1]cjing3@student.cityu.edu.hk, [2]acfang@cityu.edu.hk

Abstract. Adjectives are an informative but understudied linguistic entity with good potentials in sentiment analysis, text classification and automatic genre detection. In this article, we report an investigation of the variations in adjective use across different text categories represented in a sizable corpus. In particular, we report the distribution of adjectives across a range of categories grouped together as academic prose in the British National Corpus. We shall measure inter-category similarity in the use of adjectives and demonstrate with empirical data that adjectives are an effective differentia of text categories or domains, at least in terms of arts and sciences as the two major sub-categories within academic prose.

Key Words: corpus, text category, adjective, similarity, BNC

1 Introduction

Adjectives are an informative but understudied linguistic entity [1, 2], drawing more and more attention within the research community. Focus has been mostly on the semantic aspect of adjectives for practical research in sentiment analysis applicable to automatic evaluations of email communication [3], blogs [4] and customer reviews in [5]. Studies in this respect typically focus on evaluative adjectives [6] and size adjectives [7]. In addition to the semantic approach, adjectives are also used for purposes of text categorization and genre detection in [8]. In this respect, [2] and [9] have generally shown with corpus evidence that adjectives occur more often in written texts than in spoken ones, and more frequently in informative writing than in imaginative writing. According to [8], 'the literature suggests that adjectives and adverbs will vary by genre because of their unique patterns of usage in text' (p. 4).

This paper describes one of the recent attempts to study adjectives from the perspectives of text categorization and genre detection. In particular, we investigate the variations of adjective use across various types of academic writing selected from a large-sized corpus. We attempt to ascertain whether adjective-based indices will be able to classify texts in such a way that conforms to manual classification. As we shall show in this article with empirical data, adjectives do differ by text categories and therefore appear to be an important differentia of text categories. More importantly,

Received 10/11/08
Accepted 15/12/08
Final version 05/02/09

such a difference in adjective use (in terms of token similarity and type similarity between categories) may offer new insights into text categorization and automatic term recognition. Since the texts we used are samples of academic prose grouped according to domain, our study therefore suggests that the grouping criterion offered by adjectives seems to be a semantic one, therefore a useful complement to other studies that have shown adjectives to effectively distinguish between speech and writing in the first place, and formal and informal writings as varying degrees of formality.

The rest of the article will be organized as follows. Section 2 will briefly review three related studies. Section 3 will present a description of the corpus material after a discussion of our methodology. Section 4 will attempt to present the results and demonstrate that similarities in adjective use (in terms of tokens and types) seem to be able to group academic prose according to domains. We shall then draw some initial conclusion in Section 5.

2 Previous Studies

In this section we provide a review of three previous studies on adjectives across text categories. A classic corpus-based study [10] analyzed the distribution of the major word classes across four core fields, such as conversation, fiction, news and academic prose, in the Longman Spoken and Written English Corpus. What concerns us is the use of adjectives across the chosen genres. The results show that adjectives are more common in written texts than in spoken texts. Among written texts, adjectives are most common in academic prose. News has fewer adjectives than academic prose but more adjectives than Fiction. The findings seem to suggest a correlation between adjective use and formality of texts.

Later, [2] studied the 100 most frequent adjectives across genres in three written corpora and also analyzed the syntactic and semantic features of those adjectives. Nevertheless, they also reported and compared the distributional features of adjectives as a whole in Wellington Corpus of Written New Zealand English, Brown Corpus and the LOB corpus. [2] also shows that adjectives are used unevenly across different written texts in all the three corpora. To be more specific, adjectives appear most often in academic prose, reviews and hobbies, while they are less frequent in fiction. The findings echo the results in the written texts in [10].

The two studies touch upon the distribution of adjectives across text categories, whereas [8] not only analyzed adjectives and adverbs across genres and also attempted to examine whether they can discriminate different genres. Rittman [8] employed 44 trait adjectives, 30 speaker-oriented adverbs, and 36 trait adverbs to examine the three chosen genres (i.e. academic, fiction, and news) in the British National Corpus (BNC). First, the investigation was made among the three genres, academic vs. fiction vs. news, or called 'one-against-one' classification. Secondly, a one-against-many classification was made when each of the chosen genres used as a host category and the rest of other genres in the BNC as the guest category. For example, the comparison was made between 'academic' vs. 'not-academic' (fiction, news, non-fiction, other, and spoken). The results show that the one-against-one classification tends to be more effective that the one-against-many. The study also demonstrates that using adjective and adverb features is generally superior to other models containing features such as nouns, verbs or punctuation. Moreover, among the three features employed, the speaker-oriented adverbs are more effective than the class of trait adjectives and adverbs.

To sum up, previous studies have shown that adjectives can tell speech from writing, and among writing, academic from fiction. Yet, it is still unclear whether adjective use differs across a set of subject domains, which will be the goal of the current study.

3 Methodology and Corpus Description

As mentioned in the last section, previous studies have shown that adjectives can tell speech from writing and also rank texts in a continuum scale of 'formalness'. Nevertheless, it is still unclear whether the variations of adjective use can illustrate domain similarities. When adjective use differs in different text categories from the same genre such as academic writing, the difference is more likely due to the semantic use of adjectives in different domains than due to stylistic difference in the texts. In other words, if the distribution of adjectives differs among various text categories in academic writing, we have reason to conclude that the variations of adjective use can distinguish texts of different domains. To be more specific, if the distribution of adjectives can cluster text categories in academic writing into two broad sub-categories such as arts and sciences, it would be reasonable to say that adjectives can be used as an indicator to distinguish these two different domains. If we can show with empirical data that our assumption is true, variations of adjective use can not only be applied to the ranking of texts according to degrees of formality, but more importantly to the categorization of texts according to different domains.

Given the purpose of our study, the XML Edition of the 100-million-word British National Corpus (BNC) [11] is used as the basis of our experiment. Such a large, balanced and annotated corpus serves effectively the purpose of examining certain word class (in our case, adjectives) across different text categories. According to [12], the texts in the BNC are classified into six genres, namely, academic prose, fiction, newspapers, non-academic prose, other published writing, unpublished writing, conversation, and other spoken. To investigate the variations of adjective use across text categories within a same genre, academic prose (or ACPROSE) is chosen, which has six component text categories: 'humanities and arts', 'medicine', 'natural science', 'politics, law and education', 'social science' and 'technology and engineering'. 500,000 words from each component category were randomly sampled at the text level to compose a 'sub-corpus' as the basis of our experiments. Table 1 summarizes the six categories in terms of tokens, types and type/token ratios.

Table 1. A summary of the six text categories sampled from ACPROSE

Text Category	Text Code	Word Token	Word Type	Type/Token Ratio
humanities and arts	HUM	524224	30780	5.9
Medicine	MEDI	504856	24497	4.9
natural science	NAT	536499	28448	5.3
politics, law and education	POLIT	511935	20137	3.9
social science	SOC	511655	22581	4.4
technology and engineering	TECH	535251	18456	3.4

Since the categories were sampled on a text basis, they do not have exactly the same number of tokens. As is also evident from Table 1, the six categories do not have the same vocabulary size, with 'humanities and arts' (HUM) being the highest in terms of number of types and 'technology and engineering' (TECH) the lowest for that matter.

4 Results and Discussions

On the basis of the sub-corpus created from ACPROSE, the frequencies of adjectives in the six component categories were obtained and summarized in Table 2, which lists the numbers of tokens and types of adjectives in each category as well as type/token ratios for the adjectives. Again, the category HUM has the highest type/token ratio for adjectives and TECH the lowest.

Table 2. Basic data of adjectives in the six text categories

Text Code	ADJ Token	ADJ Type	Type/Token Ratio
HUM	45157	5709	12.6
MEDI	56659	4979	8.8
NAT	54242	6086	11.2
POLIT	43867	3880	8.8
SOC	51602	4240	8.2
TECH	46650	3814	8.2

We next attempt to examine whether adjectives can illustrate the relation between different text categories and to what extent they can achieve that. To be more exact, we aim to measure the similarity or dissimilarity between component categories in terms of adjective use. We therefore define *token similarity*, a measure that reveals the proportion of adjective tokens in common use by any two categories. We also define *type similarity*, which refers to the proportion of types of adjectives that are observed in common use between categories. In this section, we describe our observations made when each text category, serving as the host category, compares with the other five categories (guest categories). We shall first explain the key concepts of *type* and *token similarity*, and then present our data in terms of type similarity and token similarity.

4.1 Key Concepts

In this sub-section, we describe how we calculate *type similarity* and *token similarity*. It takes two steps to determine *type similarity*: First, the number of types of adjectives in common use between a host category and a guest category is calculated, and then the proportions of those shared adjective types in the host category is obtained, which is called *type similarity* of the host category to the guest category, denoted by S_{type}:

$$S_{type} = \frac{\text{Number of shared types by host and guest categories}}{\text{Total number of types in host category}} \times 100\% \quad (1)$$

For example, text A is the host category and text B is the guest category. *Type similarity* of text A to text B is the proportion of shared types of adjectives by texts A and B over the total number of types in text A. The higher the proportion, the higher degree of similarity of text A has towards text B. It is also worth mentioning that *type similarity* is directional, interpreting from the viewpoint of the host category. In other words, the type similarity of text A to text B is not necessarily the same as that of text B to text A because the denominators differ.

Next, based on the shared types by host and guest categories, *token similarity* is then computed. Again the *token similarity* of a host category to a guest category is computed in two steps: Firstly, we count the frequency of shared adjective types in a given host category. Secondly, we calculate the proportions of total number of those shared adjectives in a host category, which is called *token similarity* of the host category to the guest category, denoted by S_{token}:

$$S_{token} = \frac{\text{Frequency of shared types in host category}}{\text{Total number of tokens in host category}} \times 100\% \quad (2)$$

Same as *type similarity*, *token similarity* is also directional, interpreted from the viewpoint of the host category.

4.2 Type Similarity

As for the six chosen text categories, each category is treated as the host category and its *type similarity* to the other five guest categories is calculated according to Equation (1) respectively. Table 3 presents the type similarities between the six chosen text categories, and the similarity scores are interpreted vertically from the viewpoint of host categories.

According to Table 3, with HUM as the host category, it has higher type similarities with POLIT and SOC, both above 35%. On the other hand, HUM has a slightly lower type similarity with the other three guest categories of sciences by a little over 2%. In addition, the guest categories of arts are observed to be grouped together on the top of the similarity scale, while the guest categories of sciences grouped towards the bottom of the scale. When serving as the host category, POLIT has the highest type similarities with HUM and SOC, both above 50%. The other guest categories are grouped together, all under 46%, a 4% difference between the two groups. Once again, the guest categories are noticed to be grouped neatly into the two broad categories of arts and sciences. When looking into the relation between SOC and its guest categories, we observe the same expected tendency. SOC has a closer similarity to HUM and POLIT, while NAT, MEDI and TECH are grouped together with comparatively lower similarity scores.

Table 3. Type similarity between the six text categories

			S_{type} **of Host Category**					
			Arts			Sciences		
			HUM	POLIT	SOC	MEDI	NAT	TECH
Guest Category	Arts	HUM	/	57.6	48.5	37.4	31.1	39.8
		POLIT	39.2	/	45.8	35.6	27.7	38.8
		SOC	36.0	50.0	/	36.1	30.6	42.3
	Sciences	MEDI	32.6	45.7	42.4	/	34.2	39.4
		NAT	33.1	43.4	43.9	41.8	/	44.0
		TECH	26.6	38.1	38.0	30.2	27.6	/

On the other hand, the category of MEDI has the highest degree of similarity to NAT, followed by the three guest categories of arts with similar similarity scores. We notice the unexpected behavior of TECH in this group in that it appears at the bottom of the similarity scale. When it comes to NAT as the host category, the three guest categories of arts are grouped together in the similarity scale as we expected. Compared with the sciences category, NAT has the strongest degree of similarity to MEDI, which echoes what can be observed when MEDI is the host category. It can be observed again that TECH is at the bottom of the similarity scale. It is quite within our expectation that TECH is closely related to NAT, and has the weakest relation with POLIT. It is also noted that MEDI, with the second smallest number of adjective types, ranks a little lower than SOC and HUM in the scale of type similarity.

The above observations seem to suggest a division between the two groups (i.e. arts and sciences) in terms of adjective use with a few exceptions. We therefore compute the mean of type similarities between two broad groups of arts and sciences, and the results are presented in Tables 4 and 5.

Table 4. Type similarity from the viewpoint of Arts

		S_{type} **of Arts (Host Category)**		
			Sub Mean	Mean
Guest Category	Arts	HUM	37.6	46.2
		POLIT	53.8	
		SOC	47.1	
	Sciences	MEDI	30.8	38.2
		NAT	42.4	
		TECH	41.4	
	Difference			8.0

Table 5. Type similarity from the viewpoint of Sciences

		S_{type} **of Sciences (Host Category)**		
			Sub Mean	Mean
Guest Category	Sciences	MEDI	36.0	36.2
		NAT	30.9	
		TECH	41.7	
	Arts	HUM	36.3	35.5
		POLIT	29.8	
		SOC	40.3	
	Difference			0.7

According to Table 4, the mean of type similarity between the host and guest categories of arts is 46.2%, and the similarity mean between the host category of arts and the guest categories of sciences is 38.2%. The 8% difference between the two groups apparently suggests a distinction between the arts category and the sciences category. With the sciences category as the host category, Table 5 shows that the mean type similarity between the host and guest sciences categories (36.2%) is slightly higher than the one between the host sciences category and the guest arts category (35.5%). Therefore, the type similarity of adjective use may also be used as an indicator of text categorization in that it has differentiated the arts category from the sciences category.

4.3 Token Similarity

Based on the shared types, the *token similarity* of each host category to the five guest categories is computed according to Equation (2) and the results are presented in Table 6.

Table 6. Token similarity between the six text categories

			S_{token} **of Host Category**					
			Arts			Sciences		
			HUM	POLIT	SOC	MEDI	NAT	TECH
Guest Category	Arts	HUM	/	77.6	87.9	67.5	73.0	79.8
		POLIT	79.2	/	87.7	68.4	69.8	80.0
		SOC	78.2	89.7	/	69.3	75.3	83.9
	Sciences	MEDI	72.3	85.8	86.9	/	77.6	81.2
		NAT	72.4	80.5	85.2	78.3	/	84.0
		TECH	63.9	80.2	83.8	63.9	73.3	/

When examined across all the guest categories, HUM has the highest token similarities with POLIT and SOC, both above 78%. These two guest categories are often believed to belong to a broader sense of category of 'Arts' as opposed to 'Sciences'. On the other hand, HUM has a comparatively lower token similarity with the other three guest categories of sciences, all under 73%, a 5% difference between the two groups. In other words, sub-categories of the 'Arts' seem to have a closer relation with each other as opposed to a looser relation with sub-categories of the 'Sciences'. The host category POLIT is closely related to SOC in terms of token similarity and at same time is at a reasonable distance from the sciences category including MEDI, NAT and TECH. It is also worth noticing that although HUM is at the bottom of the guest-category list, the token similarity to the host category is as high as 77.6%. With SOC as the host category, the similarity scores do not show a significant gap between the arts and sciences categories. However, we still observe that the arts are grouped together while the science categories are grouped together at the bottom of the scale.

Next we take a look at the categories of sciences. With MEDI as the host category, the token similarities to the guest categories range from 63.9% to 78.3% according to Table 6. It is significant that the guest category, which belongs to the sciences, demonstrates a greater token similarity with MEDI. The arts categories, in contrast, show a less degree of token similarity, all under 70%. It is interesting, in this regard, to note that TECH has the lowest degree of similarity with MEDI, an unexpected observation that we shall discuss later. As expected, the science categories show a stronger affinity with each other than with the arts categories when NAT has the highest degree of similarity to MEDI and a comparatively lower degree of similarity to SOC, HUM and POLIT. Yet, TECH is observed again to be grouped with the arts categories. The category of TECH is strongly related to NAT by a token similarity of 84.0, which again indicates that they belong to a broad category of the 'Sciences'. The arts categories have a comparatively lower degree of similarity to the host category, with MEDI unexpectedly fall into the same group.

The above observations again seem to suggest a division between the two groups in terms of adjective use. We further examine the mean differences between the arts and sciences categories and the results are summarized in the tables 7 and 8.

4.4 Discussion

Our observations in terms of both *type similarity* and *token similarity* show that text categories can be categorized in a meaningful way according to the proportions of shared adjectives between text categories. A text category of arts often achieves a higher degree of similarity to other text categories of the same broad group but a comparatively lower degree of similarity to text categories of sciences. It is the same case with text categories of sciences. That is, text categories of sciences tend to have a stronger similarity with each other than their similarity to text categories of arts. However, we also observe some unexpected phenomena of text categorization. The text category of TECH is a typical example. TECH is normally to be considered as a sub-category of sciences by intuition. However, the empirical data in our study shows that TECH, as a guest category, is towards the bottom of similarity scale in both token and type similarities when compared with the host categories of MEDI and NAT. In other words, the similarity score between either MEDI and TECH or NAT and TECH is closer to the score of the arts category. There are two possible explanations. One is that the variations of adjective use may not be a perfect differentia to classify text categories although they can distinguish text categories of arts from those of sciences in most cases. The other reason lies in the inconsistency of text categories in the BNC.

Table 7. Token similarity from the viewpoint of Arts

		S_{token} of Arts (Host Category)		
			Sub Mean	Mean
Guest Category	Arts	HUM	78.7	83.4
		POLIT	83.7	
		SOC	87.8	
	Sciences	MEDI	69.5	79.0
		NAT	82.1	
		TECH	85.3	
	Difference			4.4

Table 8. Token similarity from the viewpoint of Sciences

		S_{token} of Sciences (Host Category)		
			Sub Mean	Mean
Guest Category	Sciences	MEDI	71.1	76.4
		NAT	75.5	
		TECH	82.6	
	Arts	HUM	68.4	74.1
		POLIT	72.7	
		SOC	81.3	
	Difference			2.3

According to [12], texts on Linguistics in the BNC are found to be classified into both the category of social science and the category of applied science. Therefore, the unexpected results in our investigation could also be caused by such an inconsistency in the pre-defined text categories.

5 Conclusion

In this paper, we described our investigation into the variations of adjective use across different text categories. Our assumption is that when differences in adjective use can be observed across different text categories from the same genre, those differences are more likely pertaining to characteristics of adjectives rather than stylistic features of genres. Six text categories under the same genre 'academic prose' were sampled from the British National Corpus for our investigation. By examining the proportions of adjectives shared between categories, we measure similarity of adjective use in terms of tokens and types, which we define as *token similarity* and *type similarity*. The empirical data show that, when measured in both tokens and types, adjectives in common use do differ across different text categories. Generally speaking, the differences have effectively classified the six text categories into two broad groups of arts and sciences. To put it differently, a text category belonging to arts tends to have a stronger similarity to the other text categories of the arts, but a comparatively weaker similarity to text categories of sciences. On the other hand, a text category of sciences often achieves a higher degree of similarity to other text category of sciences, and a lower degree of similarity to text categories of arts. Since such categories are constructed according to their domain content, we have found it reasonable to conclude that adjectives demonstrate affinities according to domain and therefore can be used to classify texts according to domain. Our experiment results indicate that the variations of adjective use seem to be a quite reliable indicator to categorize different text categories in a meaningful way.

Acknowledgement. The research reported in this article was supported in part by a research grant (No 7002190) from City University of Hong Kong.

References

1. McNally, L., Kennedy, C.: Adjectives and Adverbs: Syntax, Semantics, and Discourse. Oxford University Press (2008)
2. Yamazaki, S. Distribution of Frequent Adjectives in the Wellington Corpus of Written New Zealand English. In: Saito, T., Nakamura, J., Yamazaki, S. (eds.) English Corpus Linguistics in Japan, pp. 63--75. Rodopi (2002)
3. Oberlander, J. and Gill, A.J. Language with Character: A Stratified Corpus Comparison of Individual Differences in E-mail Communication. Discourse Processes. 42, 239--270 (2006)
4. Chesley, P., Vincent, B., Xu, L., Srihari, R. K.: Using Verbs and Adjectives to Automatically Classify Blog Sentiment. In: AAAI-2006 Spring Symposium on "Computational Approaches to Analyzing Weblogs", pp. 27--33. Stanford, CA. (2006)
5. Hu, M., Liu, B.: Mining and Summarizing Customer Reviews. In: 2004 ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 168--177. ACM Press (2004)
6. Samson, C.: *...Is Different From...*: A Corpus-Based Study of Evaluation Adjectives in Economics Discourse. IEEE Transaction on Professional Communication. 49 (3), 236--245 (2006)
7. Sharoff, S. How to Handle Lexical Semantics in SFL: A Corpus Study of Purposes for Using Size Adjectives. In: Hunston, S., Thompson, G. (eds.) System and Corpus: Exploring Connections, pp. 184--205. David Brown BK. Co. (2004)
8. Rittman, R.: Automatic Discrimination of Genres: The Role of Adjectives and Adverbs as Suggested by Linguistics and Psychology. VDM Verlag (2008)
9. Rayson, P., Wilson, A., Leech, G.: Grammatical Word Class Variation within the British National Corpus Sampler. Language and Computers. 36, 295--306 (2001)
10. Biber, B., Johansson, S., Leech, G., Conrad, S. and Firegan, E. Longman grammar of spoken and written English. Harlow, England; [New York] : Longman. (1999)
11. The British National Corpus, Version 3 (BNC XML Edition), http://www.natcorp.ox.ac.uk/ (2007)
12. Lee, D.: Genres, registers, text types, domains and styles: clarifying the concepts and navigating a path through the BNC jungle. Language Learning & Technology, 5(3), 37--72 (2001)

Multi-Category Support Vector Machines for Identifying Arabic Topics

Mourad Abbas, Kamel Smaili and Daoud Berkani

CRSTDLA, Speech Processing Lab.
1 rue D. E. Alafghani, Algeria
INRIA-LORIA, Parole team
B.P. 101, Villers les Nancy, France
Polytechnic School, Signal and Communication Lab.
10 rue H. Badi, Algeria
m_abbas04@yahoo.fr, kamel.smaili@loria.fr, dberkani@enp.edu.dz

Abstract. It is known that Support Vector Machines were designed for binary classification. Nevertheless, it would be fruitful to extend this operation to what is called Multi-category classification. That is why Multi-category Support Vector Machines (MSVM) become nowadays the current subject of several serious researches, aiming to achieve high levels of multi-category classification tasks. This technique has been assessed recently in some fields as text categorization, Cancer classification, etc. We should notify that experiments which have been realized until now using MSVM are limited to small data sets, since its computation is more expensive. In this paper we are interested in the use of this method, for the first time in topic identification. The experiments conducted concern topic identification of Arabic language. The corpora are extracted from Alwatan newspaper. Achieved results lead to an improvement of MSVM performance in comparison to the baseline SVM method. Nevertheless, SVM still outperforms MSVM when using larger sizes of the vocabulary.

1 Introduction

The main objective of topic identification is to assign one or several topic labels to a flow of textual data. Labels are chosen from a set of topics fixed a priori. Talking about topics conduct us to clarify the definition of a topic. In [1], each keyword is considered as a topic. Whereas in other works, topics are more sophisticated corresponding to specific subject, for example politics and sports [2]. In our case, we are dealing with six topics: Culture, Religion, Economy, Local news, International news and sports.

Topic identification is used in several areas: to adapt language models for speech recognition and for machine translation, to focus on a specific use for search engines,...etc. In spontaneous speech recognition process the vocabulary has to be as large as possible. Enlarging the vocabulary increases the search space and consequently could reduce system's performance.

A language model is one of the knowledge sources which is used by a speech

© A. Gelbukh (Ed.)
Advances in Computational Linguistics.
Research in Computing Science 41, 2009, pp. 217-226

Received 10/11/08
Accepted 15/12/08
Final version 17/02/09

recognition system, in order to find out the best hypotheses respecting linguistic criteria. One way to improve the results of a speech recognition system is to adapt the language model in accordance to the concerned utterance context. The problem of topic adaptation has already been largely addressed. In [3–8], topic information is exploited in different ways, resulting in a significant reduction of the perplexity of the baseline language model and sometimes in an improvement of the word error. Hence, these studies highlight the importance of topic adaptation.

Our objective is to identify one topic among a set of others. For that, six domains have been chosen to realize the related experiments. In this paper we will focus on the use of the MSVM based on the method developed by Guermeur [9–12]. To our knowledge, this is the first time when this method is used to identify topics. Obviously, several studies have been achieved for topic identification by using SVM or a combination of classifiers [13, 7]. The method proposed by Guermeur was initially used to combine protein secondary structure prediction methods. It leads to an enhancement of the prediction by nearly 2%, when compared to the three well-known individually used methods (Gor IV, Sopma and Simpa) [14–16]. In this paper, we will adapt it to our purpose: topic identification. In section 2 we give some information about representation of texts. In sections 3 and 4, a brief description of both SVM and Multi-category SVM are presented. And finally, section 5 describes the experiments conducted for the assessment of this method in the topic identification area.

2 Corpus Representation

Topic identification is based on topic training corpora, which represent the specificities of each topic. Training corpus has to be transformed. Each document *d* is transformed into a compact vector form. This operation is generally done after the tokenization of the corpus. The dimension of the vector corresponds to the number of distinct words or tokens in the vocabulary. Each entry in the vector represents the weight of each term. For our purpose, after removing the non content words, we calculated both the frequency of each word "Term Frequency", and the documents frequency of a word, which means the number of documents in which the word w occurs at least once [17]. A general vocabulary is constructed using word frequencies. It is based on the use of the Arabic newspaper corpus *Alwatan* which contains many thousands of news articles corresponding to nearly 10 millions words.

3 An Overview of SVM

Support Vector Machine is a supervised technique based on statistical learning theory. It is used for both classification and regression [18]. In classification, it is used to generate a class label from a set of features.

Let us consider a training set described by the couple $(x_i, y_i), i = 1...m$, where $x_i \in R^n$ and $y_i \in \{1, -1\}^m$, SVM requires a solution of the optimizing problem given by the equation 1 [19,20]:

$$min_{\omega,b,\xi}\ \frac{1}{2}\omega^T\omega + C\sum_{i=1}^{m}\xi_i \tag{1}$$

subject to

$$y_i(\omega^T\phi(x_i)+b) \geq 1-\xi_i \qquad \xi \geq 0$$

SVM approach consists in finding a linear separating hyperplane with a maximal margin in a higher dimensional space, in where, training vectors x_i are mapped by the function ϕ . C is the penalty parameter of the error which is also called the capacity of the model. Support Vector Machine is based on the so-called structural risk minimization inductive principle. The objective is to minimize an upper bound on the risk with respect to the parameters of the model [11]. This bound is composed of two terms: the empirical risk and the confidence interval. The last one is a function of the model capacity, which can be expressed in terms of different measures, the most common one is known as the VC (Vapnik-Chervonenkis) dimension [11,21]. In order to minimize the risk in this case, these two terms are jointly minimized.

4 Multi-Categoy SVM

At first, multi-category Support Vector Machines had been realized through the so-called one-versus-rest strategy [22,23]. After that, more methods have been introduced like the pairwise-coupling decomposition [24,25] and the so-called k-class SVM proposed by Vapnik in [18].

According to [11] these methods cannot find a satisfactory compromise between training performance and complexity since they are not related to an explicit uniform convergence result, therefore they fail to implement the structural risk minimization principle. The used multi-class classification method which considers all classes at once has been implemented by Guermeur. It is based on the uniform strong law of large numbers.

Being faced to a k-category classification problem, with k superior or equal to 3, an architecture to perform the discriminant analysis is required. the idea is to consider all topics at once, and then many hyperplanes are calculated for the separation of all classes or topics in one step.

Let us consider a set of elements $x = \{x_j\}$ belonging to a subset of R^d. Each element x_j is labeled with the class C$_i$. i varies from 1 to k. A linear classification can be described as a set of functions f from R^d into R^k. f is then written as follows:

$$f(x) = \alpha x + b \tag{2}$$

$$\alpha = \begin{pmatrix} \alpha_1^T \\ \alpha_2^T \\ \vdots \\ \alpha_k^T \end{pmatrix} \quad and \quad b = \begin{pmatrix} b_1^T \\ b_2^T \\ \vdots \\ b_k^T \end{pmatrix}$$

Moreover, a non linear classification can be realized by introducing a kernel k_e which satisfies Mercer's conditions [26]. $f(x)$ is then given by equation 3:

$$f(x) = \begin{pmatrix} \langle \alpha_1, \phi(x) \rangle \\ \langle \alpha_2, \phi(x) \rangle \\ \vdots \\ \langle \alpha_k, \phi(x) \rangle \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_k \end{pmatrix} \tag{3}$$

Where ϕ is a non linear function. The kernel can be defined by the expression 4:

$$\forall (x_1, x_2) \in R^d \times R^d,\ k_e(x_1, x_2) = \langle \phi(x_1), \phi(x_2) \rangle \tag{4}$$

More theoretical and practical studies about MSVM, can be found in [27, 12, 25, 28].

5 Experiments and Results

In order to realize our experiments we collected several thousands of articles from an on-line Arabic newspaper: Alwatan. The articles that we are interested in belong to the following topics: culture, religion, economy, local news, international news and sports. Nevertheless, some articles can be characterized by more than one topic. We should notify that as some topics are miscellaneous, they need to be subdivided to other subtopics to avoid performance degradations. In addition, the entire corpus need to be processed. Indeed, for the topic identification task, the non content words are not necessary, that is why we eliminated them. The size of the non content words attains 27 % of the entire corpus.

The size of our corpus is about 10 millions of words. It seems to be relatively small compared to corpora of Indo-European languages used in similar experiments. In fact, the size of the corpus extracted from the French newspaper "Le monde" of 4 years, is 80 millions words [13]. Whereas, the size of the corpus extracted from AFP Arabic Newswire of almost 7 years, and released in 2001 by LDC [29, 30] is 76 millions tokens. This gap between the two sizes is justified by the compact form of Arabic words.

In the two forthcoming subsections, we will show performances of the SVM method by realizing the well-known one-versus-rest approach, and compare results to MSVM ones.

5.1 One-versus-rest Approach

The one-versus-rest approach has been widely used to handle the multi-category problem. In our case we trained six one-versus-rest classifiers and assigned a new document d to the topic T_i giving the largest value of $g_i(d)$ for $i = 1, \ldots, 6$, where $g_i(d)$ is the SVM solution from training topic i versus the rest. Training topic i means documents covering the topic T_i. In this experiment, the corpus set dedicated to training contains 4000 documents, while vocabulary size is 3000 and each document contains 150 words. Using these data, we achieved a recall of 81.5% and a precision of 89.16% (see table 1). In this case, MSVM outperforms SVM by 4.33 % in term of recall "Table 5".

Table 1. Performances by using the one-versus-rest method with a vocabulary size 3000

Topics	Recall (%)	Precision (%)
Culture	80	92.31
Religion	53.33	100
Economy	80	92.31
Local news	93.33	70
International news	86.67	92.86
Sports	93.33	87.5
Average	81.11	89.16

Nevertheless, we should point out that SVM leads to better scores when using large corpora and bigger sizes of the vocabulary. Consequently, as it is computationally easier to do one-versus-rest classification using SVM, many related works have been carried out. Indeed, in [31] SVM has been used to identify topics of Arabic texts with a vocabulary size of 43000 words. The resulted Recall and Precision rates are respectively 97.26 % and 98.52%.

5.2 MSVM Experiments

In order to know if the topics number has either a slight or an important influence on results, we preferred starting by identifying three topics: Culture, Religion and Economy. After that we achieved additional experiments by increasing the number of topics to six. Each test document contains a number of words equal to 120. The vocabulary size is 1000 words. We should notify that for all conducted experiments we have attributed 150 documents per topic in the training phase. Tables 2 and 3 show respectively performances of MSVM and SVM methods.

Performance of SVM is largely inferior to MSVM one. In this case the difference, in terms of Recall, is about 20 %. Nevertheless, in forthcoming experiments, we will see that SVM outperforms MSVM when using a vocabulary size equal to 8000 (see table 5).

The identification of three topics using MSVM yields a good performance, 91.66%

Table 2. Performances of MSVM using a vocabulary size 1000, "Identification of three topics".

Topics	Recall (%)	Precision (%)
Culture	90	86
Religion	95	95
Economy	90	90
Average	91.66	90.33

Table 3. Performances of SVM using a vocabulary size 1000, "Identification of three topics".

Topics	Recall (%)	Precision (%)
Culture	80	91.66
Religion	53.33	100
Economy	80	92.33
Average	71.11	94.66

in term of recall. Results decrease when we augment the number of topics to six, with a vocabulary size 1000 "see table 4". Indeed, performances in terms of Recall corresponding to the topics "Economy, International news, Sports, Religion" vary from 80 % to 92%. The other two topics caused an important degradation of the mean result, in fact, their Performances correspond respectively to 66 % and 60 %.

MSVM has a good theoretical background [27], and results should be better than what we achieved. However, many reasons are behind low performances of the aforementioned two topics. As we cited in the previous subsection, the two last ones cover other subtopics and then necessitate to be subdivided. Increasing the training corpus size is also an important factor which contributes to enhance results.

Table 4. Performances of MSVM using a vocabulary size 1000, "Identification of six topics".

Topics	Recall (%)	Precision (%)
Culture	66	60
Religion	92	96
Economy	80	74
Loc. news	60	64
Int. news	82	84
Sports	86	91
Average	77.66	78.16

For that, we conducted more experiments to improve MSVM performance. We used different vocabulary sizes. The size Documents varies from 120 to 150 words and the training corpus is composed of 4000 articles. In table 5 we summarize performances in terms of Recall (R) and Precision (P) resulted from the four realized experiments.

Table 5. Performances of MSVM by using different vocabulary sizes "Identification of six topics".

	Exp1	**Exp2**	**Exp3**	**Exp4**
Vocab. size	1000	2000	3000	8000
R(%) (MSVM)	77.66	80.41	85.55	89.75
P(%) (MSVM)	78.16	82.66	85.44	88.32
R(%) (SVM)	76.50	78.33	81.11	93.45
P(%) (SVM)	80.12	79.56	89.16	90.44

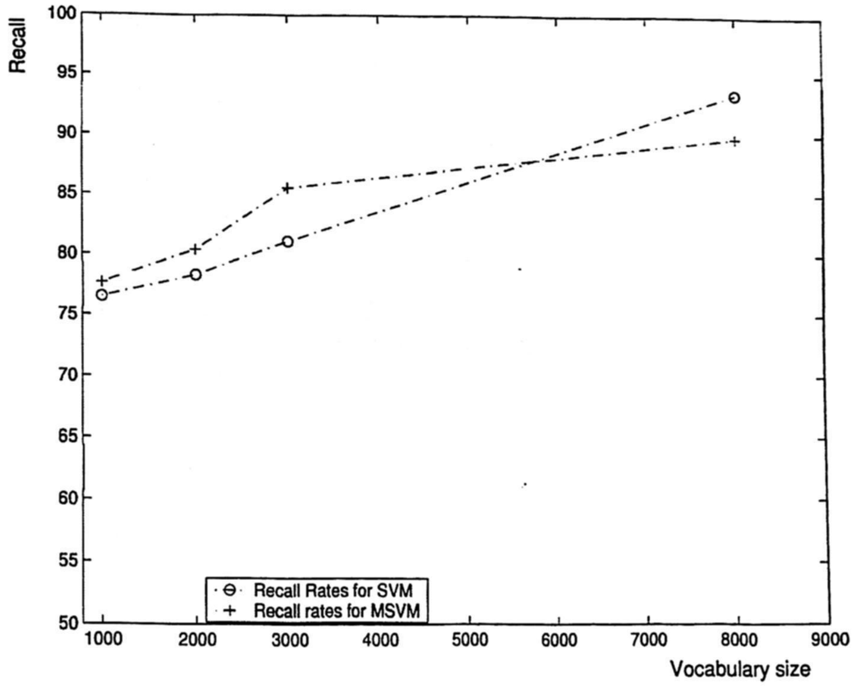


Fig. 1. Recall rates versus vocabulary size, for SVM and MSVM

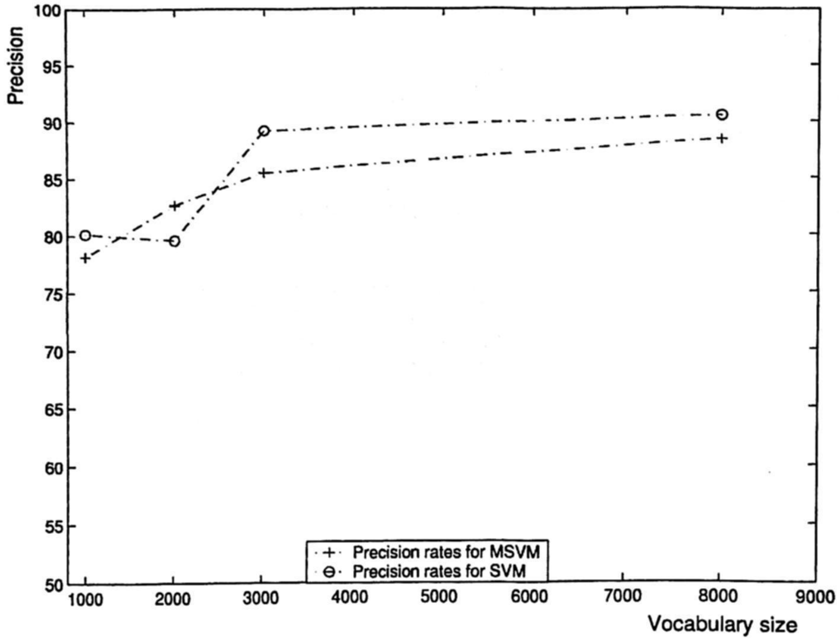


Fig. 2. Precision rates versus vocabulary size, for SVM and MSVM

According to results shown in table 5, and illustrated by figures 1 and 2 it is clear that the vocabulary size improve performances. Indeed, for MSVM we notice that recall varies from 77.66% to 89.75% according to the vocabulary size, while performances of SVM are less than MSVM, except for the size 8000 where SVM gave better results. We can consider MSVM results as an encouraging step in accordance with the computation complexity of the method.

6 Conclusion

In this paper we focused on identifying six topics for Modern Standard Arabic, using a new multi-class classification method based on a uniform convergence result. In fact, this was realized using a method proposed by Guermeur [12]. This is the first use of this method in an important domain like topic identification. For real applications, MSVM is more suitable than SVM, since we do not need to do many binary decisions. In our case we had not to calculate each time the optimal hyperplane to separate two topics. Indeed, when using MSVM, the idea was to consider all topics at once, and then many hyperplanes are calculated for separating all topics in one step.

Due to the computational complexity of the used method we were not able to use a larger vocabulary. In perspective we aim to improve this result by finding a way to overcome the computation complexity, since experiments showed that increasing the size of training corpora and also that of vocabulary always lead to satisfactory results.

References

1. Seymore, K., Rosenfeld, R.: Using story topics for language model adaptation. In: Proceeding of the European Conference on Speech Communication and Technology, Rhodes, Greece (1997)
2. Yamashita, Y., Tsunekawa, T., Mizoguchi, R.: Topic recognition for news speech based on keyword spotting. In: IEEE International Conference on Spoken Language Processing, Sydney, Australia (1998)
3. Martin, S., Liermann, J., Ney, H.: Adaptive topic-dependent language modelling using word-based varigrams. In: 3rd European Conference on Speech Communication and Technolog, Rhodes, Greece (1997)
4. Mahajan, M., Beeferman, D., Huang, X.: Improved topic-dependent language modeling using information retrieval techniques. In: Proc. of the Int. Conf. on Acoustics, Speech, and Signal Processing. (1999)
5. Yang, Y.: An evaluation of statistical approaches to text categorization. Information Retrieval **1** (1999) 69–90
6. Bigi, B., De Mori, R., El-Bèze, M., Spriet, T.: A fuzzy decision strategy for topic identification and dynamic selection of language models. Special Issue on Fuzzy Logic in Signal Processing, Signal Processing Journal **80** (2000)
7. Bigi, B., Brun, A., Haton, J., Smaili, K., Zitouni, I.: Dynamic topic identification: Towards combination of methods. In Recent Advances in Natural Language Processing (RANLP), Tzigov Chark, Bulgary (2001) 255–257
8. Brun, A., Smaïli, K., Haton, J.: Contribution to topic identification by using word similarity. In: International Conference on Spoken Language Processing (ICSLP2002), Denver, USA (2002)
9. Guermeur, Y.: Technical documentation of the multi-class SVM. Technical report, Loria, France (2004)
10. Guermeur, Y.: Combining discriminant models with new multi-class SVMs. Technical Report NeuroCOLT2, 2000-086, Loria, France (2000)
11. Guermeur, Y., Elisseeff, A., Paugam-Moisy, H.: A new multi-class svm based on a uniform convergence result. In: International Joint Conference on Neural Networks (IJCNN00). Volume IV., Come (2000) 183–188
12. Guermeur., Y., Pollastri, G., Elisseeff, A., Zelus, D., Paugam-Moisy, H., Baldi, P.: Combining protein secondary structure prediction models with ensemble methods of optimal complexity. Neurocomputing **56** (2004) 305–327
13. Brun, A.: Détection de thème et adaptation des modèles de langage pour la reconnaissance automatique de la parole. PhD thesis, Henri Poincaré University, Nancy1 (2003)
14. Garnier, J., Gibrat, J.F., Robson, B.: GOR method for predicting protein secondary structure from amino acid sequence. Methods Enzymol **266** (1996) 540–553
15. Geourjon, G., Deleage, G.: SOPMA: significant improvments in protein secondary structure prediction by consensus prediction from multiple alignments. CABIOS **11** (1995) 681–684

16. Levin, J.M.: Exploring the limits of nearest neighbour secondary structure prediction. Protein Eng. **10** (1997) 771–776
17. Joachims, T.: Text categorization with support vector machines: Learning with many relevant features. In: European Conference on Machine Learning (ECML), Chemnitz, Germany (1998) 137–142
18. Vapnik, V.N.: Statistical learning theory. John Wiley & Sons, Inc., N.Y. (1998)
19. Boser, B., Guyon, I., Vapnik, V.: A training algorithm for optimal margin classifiers. In: the Fifth Annual Workshop on Computational Learning Theory, Pittsburgh (1992) 144–152
20. Cortes, C., Vapnik, V.: Support-vector network. Machine Learning **20** (1995) 273–297
21. Vapnik, V., Chervonenkis, A.Y.: On the uniform convergence of relative frequencies of events to their probabilities. Theory of probability and its applications **16** (1971) 264–280
22. Scholkopf, B., Burges, C., Vapnik, V.: Extracting support data for a given task. In: ICKDDM'95, Menlo Park, CA, AAAI Press (1995) 252–257
23. Vapnik, V.: The Nature of Statistical Learning Theory. Spinger, New York (1995)
24. Mayoraz, E., Alpaydin, E.: Support vector machines for multi-class classification. Technical report, IDIAP (1998)
25. Weston, J., Watkins, C.: Multi-class support vector machines. Technical report, Royal Holloway, University of London, Department of Computer Science (1998)
26. Aizerman, M., Braverman, E., Rozonoer, L.: Theoretical foundations of the potential function method in pattern recognition learning. Automation and Remote Control **25** (1964) 821–837
27. Lee, Y., Lin, Y., Wahba, G.: Multicategory support vector machines: Theory and application to the classification of microarray data and satellite radiance data. Journal of the American Statistical Association **99 No. 465** (2004) 67–81
28. Bredensteiner, E.J., Bennet, K.P.: Multicategory classification by support vector machines. In: Computational Optimizations and Applications. Volume 12. (1999) 53–79
29. Abdelali, A., Cowie, J.: Regional corpus of modern standard arabic. In: second international conference on Arabic Language Engineering. Volume 1., Algiers, Algeria (2005) 1–12
30. Abdelali, A., Cowie, J., Soliman, H.: Building a modern standard corpus. In: Workshop on Computational Modeling of Lexical Acquisition, The Split Meeting, Split (2005)
31. Abbas, M., Smaili, K.: Comparison of topic identification methods for arabic language. In: Recent Advances in Natural Language Processing RANLP05, Borovets, Bulgaria (2005) 14–17

USUM: Update Summary Generation System

C Ravindranath Chowdary and P Sreenivasa Kumar

Department of Computer Science and Engineering
Indian Institute of Technology Madras
Chennai 600 036, India
{chowdary,psk}@cse.iitm.ac.in

Abstract. *Huge amount of information is present in the World Wide Web and a large amount is being added to it frequently. A query-specific summary of multiple documents is very helpful to the user in this context. Currently, few systems have been proposed for query-specific, extractive multi-document summarization. If a summary is available for a set of documents on a given query and if a new document is added to the corpus, generating an updated summary from the scratch is time consuming and many a times it is not practical/possible. In this paper we propose a solution to this problem. This is especially useful in a scenario where the source documents are not accessible. We cleverly embed the sentences of the current summary into the new document and then perform query-specific summary generation on that document. Our experimental results show that the performance of the proposed approach is good in terms of both quality and efficiency.*

1 Introduction

Currently, the World Wide Web is the largest source of information. Huge amount of data is present on the Web and large amount of data is added to the web constantly. Often the information pertaining to a topic is present across several web pages. It is a tedious task for the user to go through all these documents as the number of documents available on a topic will range from tens to thousands. It will be of great help for the user if a query specific multi-document summary is generated. Summary generation can be broadly divided as abstractive and extractive. In abstractive summary generation, the abstract of the document is generated. The summary so formed need not have exact sentences as present in the document. In extractive summary generation, important sentences are extracted from the document. The generated summary contains all such extracted sentences arranged in a meaningful order. In this paper, generated summaries are extractive. Summary can be generated either on a single document or on several documents. In multi-document summary generation, other issues like time, ordering of extracted sentences, scalability etc. will arise.

Summary can be either generic or query specific. In generic summary generation, the important sentences from the document are extracted and the sentences

© A. Gelbukh (Ed.)
Advances in Computational Linguistics.
Research in Computing Science 41, 2009, pp. 229-240

Received 10/11/08
Accepted 15/12/08
Final version 05/02/09

so extracted are arranged in appropriate order. In query specific summary generation, the sentences are scored based on the query given by the user. The highest scored sentences are extracted and presented to the user as summary. For a set of documents on a topic and a query related to the topic, suppose a summary is available. If a new document is now made available to the system then summary has to be regenerated with the new document included into the input set by running the query-specific summarizer. But this is not a good solution as it takes considerable amount of time to run the summarizer afresh and a lot of space to store all the original documents. Also, most of the times the documents used for summarization may not be accessible.

Broadly the methodology used to summarize multiple documents is to combine all the documents into a unified structure in an intelligent (each system/approach has its own methodology) fashion and then the summary is generated by taking the unified structure as the input. The construction of unified structure is a time taking task. So, the time complexity of query-specific multi-document summarization is high.

In this paper we address the following problem: given an extractive summary that is generated for a given query on a set of documents, upon the arrival of a new document, the summary has to be updated without considering the initial set of documents. The proposed system will be given the present summary and the $((n+1)^{th})$ document as the input and the output should be the updated summary. In this paper we propose a novel and efficient model for query-specific update summary generation using extractive mechanism. To the best of our knowledge this problem is not addressed in the literature.

The rest of the paper is organized as follows: In Section 2 we discuss the related work. Generation of embedded document is discussed in Section 3. In Section 4 we introduce the methodology to accomplish the task of update summary generation. Experimental setup is discussed in Section 5. In Section 6 results are discussed and conclusions are given in Section 7.

2 Related Work

Text summarization has gained popularity in the recent years. A generic summary generation on single document is discussed by Cajun Wan et al. [1]. Both summary and keywords are extracted from a single document by following iterative reinforcement approach. To extract summary from the document, the following relations are used: sentence-sentence relation, word-word relation and sentence-word relation. A generic summary generation on multiple documents is discussed by Radev et al. in [2]. Centroid based approach is followed by this system, called MEAD, to generate summary. Given a set of documents about a particular topic i.e., a cluster of documents, the centroid of the cluster is calculated. A score is given to each sentence in the cluster with respect to the centroid. Sentences are selected in decreasing order of sentence scores and are arranged with respect to the chronological order of their respective documents.

Extractive summary generation is discussed in [2–5]. The input to extractive summarizers is the set of documents that are to be summarized and the output is the sentences extracted from the input documents. The sentences so extracted are arranged in a manner which increases coherence (logical flow) to the generated summary. In particular, the former criteria is addressed in [4].

Single document generic summary is discussed in [1], here, extraction of the sentences from a document to generate a summary is accomplished by using sentence-sentence, word-word and sentence-word relationships. Single document query-specific summary generation is discussed in [5], here, a connected subgraph of sentences are extracted from the document graph. Sentences are said to be connected if the similarity measure between them is above a threshold. Multi-document generic summary generation is discussed in [2, 6]. In [2], all the sentences from the documents are given scores and the sentences are selected into the summary in the decreasing order of their scores. In [6], sentences are given scores based on the model inspired by PageRank [7].

Multi-document query-specific summary generation is discussed in [8, 4]. In [8], query is also considered as one of the sentences in a document. Similarities between all the pairs of sentences in the documents are calculated and these similarity values are used while giving the scores to the individual sentences. In [4], two types of scores are calculated, first one is based on the similarity between sentences and the second is based on the similarity of a sentence with respect to the query.

Centrality based approaches are discussed in [9, 6, 10, 11]. In centrality based approaches, the salience of a sentence is calculated based on both the contribution of the sentence and the type of neighbouring sentences it is surrounded. Degree centrality is discussed in [9] and eigenvector centrality is discussed in [6, 10, 11]. Concept of bushy path was introduced by Salton et al. in [9]. Nodes with high degree are called bushy nodes. Bushy path is defined as a path connecting top n bushy nodes. Eigenvector centrality of a node is calculated by taking into consideration both the degree of the node and the degree of the nodes connecting to it. This is inspired by PageRank [7].

Redundancy handling is addressed in [12]. This principal is followed by many other systems. Mean marginal relevancy(MMR) principal is as follows: Node scores are calculated w.r.t the query. Summary is generated incrementally. A node with highest score is selected into the summary. All the scores of remaining nodes are recalculated based on the nodes already selected into summary and the node score they possess. From the recalculated scores, the highest scored node will be added to summary.

In all the above approaches, a summary is generated from scratch. In this paper we address the problem of updating the extracted summary with the availability of a new document. Here we update the summary for a given query. This problem of update summary generation is proposed by us and the detailed procedure to accomplish this task is explained in the following sections.

3 Generating Summary-Embedded Document

We follow a graph based approach to accomplish the task of update summary generation. Every sentence in the document is a node and the edges are placed between the nodes if the similarity score between them is above a threshold. Hereafter we use the words, "node" and "sentence", interchangeably. Similarity between the nodes is calculated using the Equation 1.

$$sim(\overrightarrow{n_i}, \overrightarrow{n_j}) = \frac{\overrightarrow{n_i}.\overrightarrow{n_j}}{|\overrightarrow{n_i}||\overrightarrow{n_j}|} \quad (1)$$

where $\overrightarrow{n_i}$ and $\overrightarrow{n_j}$ are term vectors for the nodes n_i and n_j respectively. The weight of each term in $\overrightarrow{n_i}$ is calculated as $tf * isf$. Here tf is *term frequency* and isf is *inverse sentential frequency*. *term frequency* is defined as the number of times a term occurs in a sentence. *inverse sentential frequency* is defined as $log(\frac{N}{n_t+1})$, where N is total number of sentences in the document and n_t is number of sentences in which the term is present.

In this section we propose an approach to embed the summary into the new document. Algorithm 1 sketches the details of the embedding of the current summary into the new document. The Algorithm 1 gives the method of embedding

Algorithm 1 To embed summary into the document

```
Input: CurrentSummary and NewDocument
Output: Document with summary embedded into it
if size(CurrentSummary) $\geq$ size(NewDocument) then
    Swap CurrentSummary and NewDocument
end if
Let $d_1, d_2.....d_y$ be the nodes in document
{//No. of nodes in document = "y"}
Let $s_1, s_2.....s_x$ be the nodes in summary
{//No. of nodes in the summary = "x"}
EmbeddedDocument = NewDocument
Insert the last sentence of the summary into the EmbeddedDocument(all the nodes
in the EmbeddedDocument are considered for insertion) using the strategy ex-
plained in Section 3.1
Insert the first sentence of the summary into the EmbeddedDocument(only the
nodes above the $s_x$ in the EmbeddedDocument are considered for insertion) using
the strategy explained in Section 3.1
while All the nodes of summary are not embedded into the EmbeddedDocument
(starting from $s_2$) do
    {// Consider the insertion in the summary order}
    Insert the summary node $s_i$ into the EmbeddedDocument(only the nodes be-
    tween $s_{i-1}$ and $s_x$ in EmbeddedDocument are considered for insertion) using
    the strategy explained in Section 3.1
end while
Return EmbeddedDocument
```

the sentences from summary into the document. Line 3 is very crucial, here the $size(D)$ gives the number of sentences in D. Idea is that if the size of the summary is less than the new document's size then the summary will be embedded into the new document otherwise the new document will be embedded into the summary.

3.1 Insertion Strategy

This section gives the detailed explanation of insertion strategy. A node s in the summary is placed in the document appropriately. The steps to be followed are given below:

- Similarity (calculated using Equation 1) of s is calculated with the nodes (the nodes that are specified in Algorithm 1) in the document.
- Let y be a node in the document which has maximum similarity with the node in the summary.
- Let x and z be the preceding and following nodes of y respectively.
- Calculate the similarity of s with x and z.
- s is placed in between x and y if s has greater similarity value with x than z, otherwise s will be placed in between y and z.

3.2 Handling Exceptions

When the similarity value of node s_i is zero with every node of the EmbeddedDocument then the node is inserted immediately after s_{i-1} in EmbeddedDocument. Here node s_i is the node that is following node s_{i-1} in the summary. If s_{i-1} is not present then the node is placed immediately before s_{i+1} in the EmbeddedDocument (in this case, s_{i+1} is inserted before inserting s_i). The former process is recursive in nature. Even after calling recursively if the nodes in the summary are not embedded then the summary will be appended to the document.

This exception handling module will be used rarely by the system. We assume that the new document which arrived is related to the topic and therefore it is unlikely that the sentences in summary will have similarity value of zero with the sentences in the new document. Even otherwise the strategy holds good i.e., if the new document is an outlier(document that does not contain any information related to the query) then none of the sentences will be selected from the new document and the sentences of old summary alone will be selected.

4 Update Summary Generation

In this section, summary generation on the embedded document is discussed. Here the score of the node is calculated based on the query posed by the user i.e., the node gets score based on its relevance to the query.

4.1 Node Score

Node score calculation is based on the Equation 2.

$$f(n, q_i) = \begin{cases} 1/t & \text{if } q_i \text{ is present in } n \\ 0 & \text{if } q_i \text{ is not present in } n \end{cases}$$

Here t is the number of query terms in the given query, n is the sentence and q_i is the query term. If the query term is present in the sentence then a non-zero value is assigned otherwise zero is assigned.

$$w_{q_i}(s) = d * f(s, q_i) + \frac{(1-d)}{a} \sum_{v \in adj(s)} sim(s, v) * f(v, q_i) \tag{2}$$

Here a is the number of sentences adjacent to s that have the query term and have non-zero similarity with s. d is the bias factor. In Equation 2, the first part captures the importance of the sentence with respect to the query term and the second part captures the type of neighbours (adjacent sentences). Two sentences are said to be adjacent if the similarity value between them is above a threshold(=0.001). *The score of a node is the summation of Equation 2 over all the query terms.* Unlike the node score equation in [4], the Equation 2 is not iterative. Also, this equation considers only immediate neighbours while assigning node scores. This makes the system efficient.

4.2 Summary Generation

Node scores are calculated for all the nodes and summary generation is explained in this section. In Algorithm 2, it is assumed that $SummarySize$ (number of sentences that user wants as a summary) is not greater than the number of sentences in the EmbeddedDocument. From Lines 7 to 9, the completeness of the summary is achieved. A summary is complete if all the query terms are present in it. Then the nodes are added from the remaining pool as given in Lines 12 to 15. In Line 7, Equation 3 is used to recalculate the node scores and in Line 12 the maximum scored node is selected using the Equation 4. Note that here, scores are assigned to nodes temporarily using Equation 3 and Equation 4 is used to select the highest scored node into summary. After the selection, the node scores are reverted to their original scores(as calculated in Section 4.1).

$$tempW_Q(n_i) = \kappa\lambda \sum_{1 \le k \le t} w_{q_k}(n_i) - (1-\lambda) \underset{j}{Max}\{sim(n_i, s_j)\} \tag{3}$$

$$\underset{i}{Max}\{tempW_Q(n_i)\} \tag{4}$$

Here n_i and s_j represents document and summary nodes respectively. $tempW_Q(n_i)$ is the temporary node score of n_i. Equation 3 is inspired from [12]. The sentences in the summary generated using Algorithm 2 are rearranged in the document order. This summary is complete, coherent and also non-redundant. The value of λ is taken from [12]. κ is used as a scaling factor and it is fixed empirically.

Algorithm 2 Generating Summary

```
Input: EmbeddedDocument
Output: Summary
SUMMARY = null
COUNT = null
Select the highest scored node in EmbeddedDocument into the SUMMARY
while All the query terms are not included into the SUMMARY AND (COUNT != SummarySize) do
    Recalculate the scores of the nodes using Equation 3{//This calculation is only for temporary purpose. At the beginning of each iteration the nodes are assigned their original node scores}
    Select a node into SUMMARY that maximizes number of query terms in the SUMMARY{//IF more than one such node is present then select the node that has maximum score}
    COUNT++
end while
while COUNT != SummarySize do
    Select the next highest scored node from EmbeddedDocument using Equation 4
    Add the highest scored node to SUMMARY
    COUNT++
    Calculate temporary node scores using Equation 3
end while
Return SUMMARY
```

4.3 Discussion

Complete Summary While selecting sentences into summary, the sentences which will cover maximum uncovered query terms are given highest preference. The generation of summary is carried out by adding one sentence followed by another. Fist sentence which is included into summary will be the highest scored sentence. The sentences selected after that are targeted towards maximizing the number of query terms coverage. If more than one sentence is contributing the same number of query terms then the highest scored sentence among them will be selected to be included into the summary. This process is repeated till all the query terms are included into the summary.

Coherent Summary The sentences selected into the summary are arranged in the EmbeddedDocument order. The insertion strategy discussed in Section 3 ensures that the EmbeddedDocument is coherent i.e., the sentences in the EmbeddedDocument are well connected and there is a logical flow within sentences. Therefore updated summary is coherent.

Quality Summary After achieving the task of complete summary, the nodes that are included are purely based on two criteria: First one is the node's importance w.r.t the query and second is its contribution to the summary. Contribution is the amount of new information it is adding to the summary. In other words it is non-redundancy. So, Equation 4 is used to select the sentences which ensures the non-redundancy and thus the quality of the summary. Recall that before selecting the highest scored node, Equation 3 is used to calculate the temporary node scores.

Efficiency In this system we embed the summary into new document in a coherent manner and then the summary is generated by extracting sentences from the embedded document. The complexity of the system is $O((S_i + D_j)^2)$, S_i and D_j are number of sentences in current summary and new document respectively. The complexity of a multi-document summarizer is $O((\sum D_j)^2)$.

5 Experimental Setup

Evaluating the proposed system is a difficult task. Update summary generation is evaluated on DUC 2006[1] corpus. DUC has 50 topic clusters and each topic is described in 25 documents. Initial summary is generated using the MEAD [2] system for the query and the document cluster provided by DUC. This summary is generated on the first 15 of the 25 documents. The summary generated will be the input for the update summary generation task. The 16^{th} document will be the new document into which the summary is to be embedded. The summary is generated for the given query on the embedded document and this generated summary will be embedded into 17^{th} document. The process is repeated till the summary on the last embedded document(25^{th}) is generated.

The block diagram for the experimental setup is shown in Figure 1. MEAD [2] follows centroid based approach to generate summaries. It deals with both single and multi-document summarization. In our setup we use MEAD's multi-document summarization approach. MEAD computes a score for each sentence from the given cluster of related documents by considering a linear combination of several features. We have used centroid score, position and cosine similarity with query as features with 1,1,10 as their weights respectively. MMR(Maximum Marginal Relevance) re-ranker is used for redundancy removal with a similarity threshold of 0.6. The updated summaries so formed are all stored and evaluated against the model summaries given by DUC. In DUC, the model summaries are of fixed length i.e., 250 words. So, all the generated summaries are truncated to 250 words.

5.1 Discussion on Baseline Summaries

This problem is first posed by us and therefore there is no other system available to be compared with the performance of our system. As this is an update summary generation task there is no meaningful baseline system that can be compared with our system. The following alternatives were thought of for a baseline system: 1) Generate a baseline summary using MEAD with all the 25

[1] http://duc.nist.gov

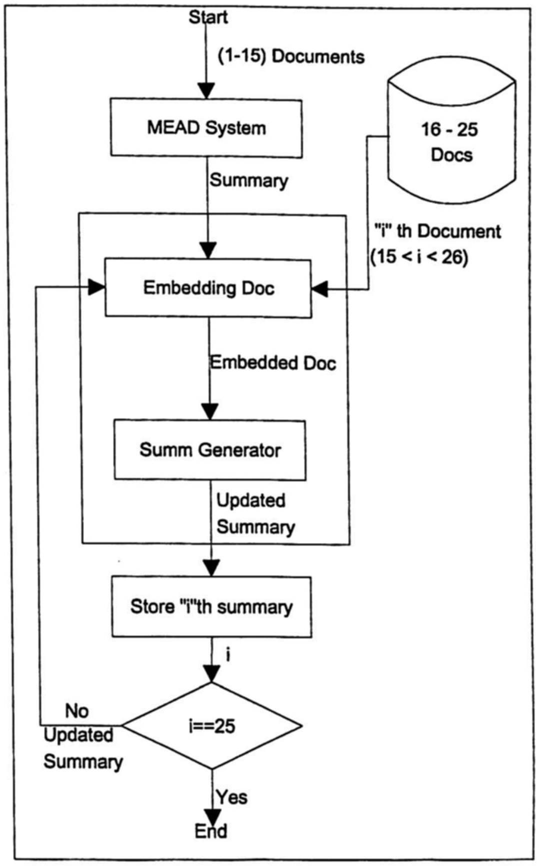


Fig. 1. A block diagram of experimental setup

documents as input. As our system generates the summary by considering only the current summary and new document, this is not a fair comparison. 2) If baseline summary for i^{th} document inclusion is available then baseline summary for $i + 1^{th}$ document inclusion can be calculated using MMR approach. But the former approach requires the presence of all the $i + 1$ documents to generate a baseline summary. So, it also will not be appropriate baseline.

So, we give the ROUGE results generated by the best performing system of DUC 2006(System-24), these values are for summaries generated by considering all the 25 documents. But USUM's ROUGE values are *not* obtained by considering all the 25 documents. So, the values of the best system of DUC 2006 would naturally be better than our systems values.

6 Experimental Results

The ten updated summaries for each cluster are evaluated according to DUC 2006 specifications. DUC uses ROUGE measures to evaluate the quality of the summary generated by comparing with the model summaries. Recall is calculated for the generated summaries w.r.t this model summaries. ROUGE[13] stands for Recall-Oriented Understudy for Gisting Evaluation. ROUGE measures the quality of a summary by comparing it to the summaries created by volunteers. ROUGE-N is n-gram recalls between system generated summaries and the summaries generated by the volunteers(models). ROUGE-N is calculated based on the Equation 5

$$ROUGE\!-\!N = \frac{\sum\limits_{s \in model\ summaries} \sum\limits_{gram_n \in s} count_{match}(gram_n)}{\sum\limits_{s \in model\ summaries} \sum\limits_{gram_n \in s} count(gram_n)} \tag{5}$$

Here n is the length of n-gram. $gram_n$ stands for n-gram. $Count_{match}(gram_n)$ is the maximum number of n-grams co-occurring in both the generated summary and in the reference summaries. ROUGE-1 and ROUGE-2 are the recall measures of unigrams and bi-grams respectively. ROUGE-W is the weighted longest common subsequences matching. In longest common subsequence matching, the distance between the words is not considered as an important issue but in weighted longest common subsequence matching, weight is given to the distance between the words. ROUGE-SU4 is the recall measure which computes the skip bi-grams with skip distance four and uni-grams are also considered while computing this measure.

In Table 1 the ROUGE values for updated summaries generated on DUC 2006 are given. The values in the table are averaged values over 50 clusters. Updated summary 1 is the summary obtained by updating the summary generated on first 15 documents with the sixteenth document. Updated summary 2 is the summary obtained by updating the updated summary 1 with the seventeenth document. We empirically found that the ROUGE values are better for κ value of 20. We also give the ROUGE values for the system-24(best performing system) of DUC 2006 in Table 2. The values in Table 2 are for the summaries generated by considering all the 25 documents of the cluster. So, the ROUGE values of Table 2 will be better than the ROUGE values of our systems. But the ROUGE values of our system are very close to the ROUGE values of the system-24. This indicates that our system is performing well.

The proposed system is implemented on the system with the following configuration: 256MB main memory, 1.7 GHz Intel Pentium processor and the operating system is FC3. The system is implemented in Java. The time taken to compute the update summaries on 50 clusters is 56 minutes. So, it is slightly greater than 1 minute per cluster. On average it is less than 7 seconds per update summary(there are 10 update summaries per cluster).

Table 1. ROUGE Values on DUC 2006 with κ value 20

Updated Summary	ROUGE-1	ROUGE-2	ROUGE-W	ROUGE-SU4
1	0.38980	0.08179	0.09429	0.13757
2	0.38660	0.07905	0.09321	0.13552
3	0.38919	0.08196	0.09418	0.13786
4	0.38871	0.08239	0.09351	0.13713
5	0.38457	0.08024	0.09274	0.13472
6	0.38467	0.08060	0.09297	0.13490
7	0.38547	0.08058	0.09339	0.13518
8	0.38282	0.08004	0.09245	0.13389
9	0.38358	0.07955	0.09281	0.13390
10	0.38432	0.08031	0.09282	0.13419

Table 2. ROUGE Values of System24 on DUC 2006

Updated Summary	ROUGE-1	ROUGE-2	ROUGE-W	ROUGE-SU4
System24	0.41108	0.09558	0.11068	0.15529

7 Conclusions

In this paper, the current summary is cleverly embedded into the new document in a meaningful and coherent way. A query specific summary is generated on the embedded document. The sentences which are extracted from the document form a *complete* summary. The algorithm proposed will not select sentences which have redundant information. All the sentences are arranged in the embedded document order to maintain the coherence and flow in the summary. The system is efficient and the quality of the update summary is satisfactory. The results are highly encouraging. USUM gives efficient solution for update summary generation which is a challenging and useful task.

References

1. Wan, X., Yang, J., Xiao, J.: Towards an iterative reinforcement approach for simultaneous document summarization and keyword extraction. In: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, Prague, Czech Republic, ACL (2007) 552–559
2. Radev, D.R., Jing, H., Styś, M., Tam, D.: Centroid-based summarization of multiple documents. Inf. Process. Manage. **40**(6) (2004) 919–938
3. Conroy, J.M., Schlesinger, J.D., Stewart, J.G.: CLASSY query-based multi-document summarization. In: Proceedings of the Document Understanding Conference (DUC-05) at NLT/EMNLP, Vancouver, Canada (2005)
4. Sravanthi, M., Chowdary, C.R., Kumar, P.S.: QueSTS: A query specific text summarization system. In: Proceedings of the 21st International FLAIRS Conference, Florida, USA, AAAI Press (2008) 219–224
5. Varadarajan, R., Hristidis, V.: A system for query-specific document summarization. In: CIKM '06: Proceedings of the 15th ACM international conference on

Information and knowledge management, New York, NY, USA, ACM Press (2006) 622–631

6. Erkan, G., Radev, D.R.: Lexpagerank: Prestige in multi-document text summarization. In: Proceedings of EMNLP, Barcelona, Spain, Association for Computational Linguistics (2004) 365–371
7. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. In: Proceedings of the 7th International World Wide Web Conference, Brisbane, Australia (1998) 161–172
8. Wan, X., Yang, J., Xiao, J.: Manifold-ranking based topic-focused multi-document summarization. In: IJCAI, Hyderabad, India (2007) 2903–2908
9. Salton, G., Singhal, A., Mitra, M., Buckley, C.: Automatic text structuring and summarization. Inf. Process. Manage. **33**(2) (1997) 193–207
10. Mihalcea, R., Tarau, P.: TextRank: Bringing order into texts. In: Proceedings of EMNLP, Barcelona, Spain, Association for Computational Linguistics (2004) 404–411
11. Mihalcea, R.: Graph-based ranking algorithms for sentence extraction, applied to text summarization. In: Proceedings of the ACL 2004 on Interactive poster and demonstration sessions, Morristown, NJ, USA, Association for Computational Lingu (2004) 20
12. Carbonell, J.G., Goldstein, J.: The use of mmr, diversity-based reranking for reordering documents and producing summaries. In: SIGIR, Melbourne, Australia, ACM (1998) 335–336
13. Lin, C.Y., Och, F.J.: Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In: ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, Morristown, NJ, USA, Association for Computational Linguistics (2004) 605–612

Towards the Speech Synthesis of Raramuri: A Unit Selection Approach based on Unsupervised Extraction of Suffix Sequences

Alfonso Medina Urrea,[1] José Abel Herrera Camacho[2] and Maribel Alvarado García[3]

[1] GIL-II, Universidad Nacional Autónoma de México
04510 Coyoacán, DF, MEXICO
amedinau@ii.unam.mx

[2] FI, Universidad Nacional Autónoma de México
04510 Coyoacán, DF, MEXICO
abelh@verona.fi-p.unam.mx

[3] Escuela Nacional de Antropología e Historia
14030 Tlalpan, DF, MEXICO
marvarado1978@yahoo.com.mx

Abstract. This work deals with the design of a synthesis system to provide an audio database for Raramuri or Tarahumara, a Yuto-Nahua language spoken in Northern Mexico. In order to achieve the most natural speech possible, the synthesis system is proposed which uses a unit selection approach based on function words, suffix sequences (derivational and inflectional morphemes) and diphones of the language. In essence, the unknown suffix units were extracted from a corpus and recorded, along diphones and function words, in order to build the audio database that provides data for Text-to-Speech synthesis.

1 Introduction

The ultimate objective of Text-to-Speech (TTS) synthesis systems is to create applications which listeners, and users in general, cannot easily determine whether the speech he or she is hearing comes from a human or a synthesizer.

Synthesized speech can be produced by concatenating recorded units (waveforms) selected from a large, single-speaker speech database. The primary motivation for using a database with a large number of units that covers wider prosodic and spectral characteristics, gives us the great benefit to produce a synthesized speech that sounds more natural than those produced by systems that use a small set of controlled units (*e.g.* diphones) [1]. There is a paradigm for achieving high-quality synthesis that uses a large corpus of recorded speech units; it is called *unit-selection synthesis*. Unit selection is a method in which we can concatenate waveforms from different linguistic structures such as sentences, words, syllables, triphones, diphones and phones. Due to the increasing computer's storage capacity, we are able to create a corpus of prerecorded

Received 15/11/08
Accepted 11/12/08
Final version 05/02/09

units. Furthermore, there are efficient searching techniques that allow a real-time searching into huge databases looking for sequences of units in order to build up the synthesized utterance.

The objective of unit selection systems is to search an audio database in order to find the optimal sequence that makes up a target utterance. The unit selection is based on minimal acoustic distortions (cost) between selected units and the target spectrum [2]. As Zhao establishes [3], "the cost function measures the distortion of the synthesized utterance; this is a summation of two subcost functions: a target cost, which describes the difference between the target segment and the candidate segment, and a concatenation cost, which reflects the smoothness of the concatenation between selected segments."

We are motivated to work with the Raramuri language group, which is constituted by a cluster of five of variants, because it is one of the relatively least endangered groups. It is worth noting that 364 variants, belonging to 68 language groups and 11 linguistic families, have been recognized rather recently as official[4] proper languages of Mexico; a true linguistic continent. Certainly, such linguistic wealth deserves to be studied in order to develop technologies that so far have been considered necessary only for the dominant languages of the world. And Raramuri seems a good place to start with. Regarding its relatively unendangered status, although government statistics are subject to question, in 1970 more than 25 thousand Raramuri speakers were counted, whereas today around 75 thousand speakers are estimated. Also, the phonological resemblance between Raramuri and Spanish and the restricted syllable structure of the former (CV) are additional motivations for our team to work with Raramuri; especially this last point has a positive impact on our TTS approach. The main challenge is that the language is not sufficiently known in order to be able to find somewhere in the bibliography enough data about the units to be used in the system we propose. This should illustrate the importance of conducting basic linguistic research in order to develop language technologies, since it is no secret that most world languages are not sufficiently documented.

2 Synthesizer

TTS is defined as "the production of speech by machines, by way of the automatic phonetization of the sentences to utter" [4]. The two characteristics used to describe the quality of a speech synthesis system are naturalness and intelligibility. The most common methods for speech synthesis are: articulatory, formant, and concatenative synthesis. Nowadays, the last two are the most used methods [5]. The formant synthesis have produced the most natural voice. However these systems provide excellent quality for some phrases and robotic voice for others [6, 7]. Our experience is that concatenative methods are more consistent than the formant or sub-phoneme ones.

[4] Instituto Nacional de Lenguas Indígenas, *Catálogo de las lenguas indígenas nacionales: Variantes lingüísticas de México con sus autodenominaciones y referencias geoestadísticas*, Mexico, http://www.inali.gob.mx/catalogo2007/.

In concatenative systems, segments of prerecorded speech are chained together to form words, phrases and so on. These methods provide not much naturalness to the output speech, and it also results in audible glitches. However several methods based on overlapping adds have been applied to provide naturalness [8]. Also, it is usual to smooth the output waveform at the point of concatenation in order to gain naturalness. Three main sub-methods are used, based on the type of unit:

- unit-selection
- diphones
- domain-specific synthesis

Regarding concatenative synthesis, domain-specific synthesis requires a corpus of prerecorded words and phrases. It works well for systems with a very limited vocabulary (*e.g.* talking clocks). Also, a diphone is a segment that contains the stable parts of two adjacent phones; according to the specific phonotactics of the target language, the number of potential diphones for a given language is the square of its phones. Some must be discarded because they are incompatible with the phonotactics of the language. The exclusive use of diphones results in relatively small speech database, but the lack of clarity with the resulting speech is a disadvantage. Unit-selection requires a larger database; its corpus typically includes diphones, syllables, words, phrases and sentences. The synthesizer determines in real-time the best units from the database for the output utterance.

One of the major problems with all concatenative systems is how to deal with the boundaries between segments. It is clear that minimizing the number of occurrences of boundaries is likely to improve the quality of speech; reducing the number of boundaries involves, of course, using longer units. The point is: the longer the unit, the greater the number and detail of boundaries within them [9].

In theory, at the phone level there has to be an entry for every possible combination of phones and of phones and silence. But there are a number of combinations that do not exist in Raramuri and can be excluded. The original speech recording needs to be as monotonous as possible to reduce discontinuities between different segments and to reduce as much as possible any need for signal processing. This database must be stored in uncompressed PCM format to reduce compression-induced degradation of the signal. Once we have a stream of diphones the last step is to join them into a complete utterance.

The quality of concatenative synthesizers highly depends on the quality of the recorded speech units. Because of the speech sounds range from x to y, the speech corpus will be recorded in WAV format, under a sampling rate of 22,050 kHz with a 16-bit resolution. The recording sessions will be made in a professional studio. However, the recordings will be reduced later to a band of 4 kHz.

There are different types of units to be recorded. As mentioned above, we aim at function words, affix sequences and diphones. These segments were obtained from recorded utterances, preserving their suprasegmental features. In our system, we use the three units forming a concatenative combinational system. These systems have better performance [9]; and the software designed to decide what unit will be used, and to query the increased database, does not pose a great challenge.

Another reason to use diphones as our basic units is that they model the joints well most of the time. Despite the price of storing a large number of permutations of units, this choice is much more convenient than using syllables because they would need a considerably larger number of permutations.

At first sight, it might be suggested that "the longer the unit length, the less troublesome will be any errors because larger 'amounts' of semantic content will be captured by each unit. If this is the case then errors in conjoining small units like phones will be the most critical for perception. This is precisely the reason why some researchers prefer diphones as the small units rather than phones —the very structure of the model is designed to minimize listener awareness of error, and errors are most likely to occur at coarticulated joins between segments." [9].

The idea of using larger units is simple, since coarticulation problems happen between individual units, for example between phones; The larger the unit, the greater the number of joints that will not require postprocessing. Therefore, whole function words and suffix sequences will be selected for recording, taking the most frequent sequences of function words found in the corpus.

Here, we give a brief description of the units involved in the system, ordered hierarchically from the lexical level to the phonetic one:

Function Words. Words are syntactic units; in this case, words are obtained from phrases. As it would be expected of any language in general, function words are the most frequent ones in the Raramuri corpus. The least frequent ones are normally content words which may be constructed by means of diphone concatenations.

Affix Sequences. Raramuri exhibits a small set of inflectional suffixes which tends to follow an interesting set of derivational suffixes.

Diphones. Diphones are defined as a stretch from the least varying (most stable or steady-state) part of a phone to a similar point in the next phone. The idea of introducing diphones was to capture the transition between phones within the acoustic model in order to reduce mismatches between phones.

Units which stand higher in this hierarchy already have internal boundaries modelled correctly by definition. That is, when they are used for concatenation, their suprasegmental features are implied.

The task of labeling consists of analyzing waveforms and spectrograms as well as making annotations to the waveforms of the recorded speech in order to extract information about the recorded units. In general, unit selection systems require phonetic labeling to identify limits between segments (phrases, words, diphones). It is also necessary to apply prosodic labels to give us information about tone and stress. Phrasal labels identify limits of each phrase recorded in the corpus. Word labels consist in time markers at the beginning and end of words. Tone labels are symbolic representations of the melody of the utterance. This job is usually done by automatic speech labeling tools because of the database's size. For phonetic labeling, speech recognizers are used in forced alignment mode, where the recognizer finds the boundaries between segments. Automatic prosodic labeling tools work from a set of linguistically motivated acoustic features (e.g., normalized durations, maximum/average pitch ratios) plus some binary features looked up in the lexicon (e.g., word-final vs. word-initial stress) [10]. Unfortunately we don't have this kind of tools and the development of them would requiere more time to mark up text; because of this, we marked up the recorded units manually.

3 Identifying Units for Raramuri

Raramuri or Ralamuli, also known as Tarahumara, is a Yuto-Nahua or Uto-Aztecan language spoken in northern Mexico. It is more an agglutinative language than a fusional one. Word formation is mainly accomplished by means of suffixation. As could be expected, stems are followed by derivational suffixes, and these by inflectional ones. Also, its syllable structure is mainly CV, although syllable V is possible. Since unit-selection synthesis presupposes the units of the language that must be recorded in order to compile the database that will be used to build up the synthesized utterance, these facts are relevant in order to pick the appropriate units for the synthesizer. Hence, given the predominant syllable structure, it makes sense to pick diphones as the basic units. Table 1 shows the possible CV diphones according to the phonotactics of the language. Additionally, there are 50 VV diphones (two syllables) possible in the language.

Table 1. Possible CV Syllables of Raramuri

CV	**a**	**e**	**i**	**o**	**u**	**a'**	**e'**	**i'**	**o'**	**u'**
m	ma	me	mi	mo	mu	ma'	me	mi'	mo'	mu'
n	na	ne	ni	no	un	na'	ne'	ni'	no'	un'
k	ka	ke	ki	ko	ku	ka'	ke'	ki'	ko'	ku'
p	pa	pe	pi	po	pu	pa'	pe'	pi'	po'	pu'
t	ta	te	ti	to	tu	ta'	te'	ti'	to'	tu'
c	ca	ce	ci	co	cu	ca'	ce'	ci'	co'	cu'
g	ga	ge	gi	go	gu	ga'	ge'	gi'	go'	gu'
b	ba	be	bi	bo	bu	ba	be'	bi'	bo'	bu'
r	ra	re	ri	ro	ru	ra'	re'	ri'	ro'	ru'
h	ha	he	hi	ho	hu	ha'	he'	hi'	ho'	hu'
w	wa	we	wi	wo	wu	wa'	we'	wi'	wo'	wu
y	ya	ye	yi	yo	yu	ya'	ye'	yi'	yo'	yu'
s	sa	se	si	so	su	sa'	se'	si'	so'	su'
l (R)	la	le	li	lo	lu	la'	le'	li'	lo'	lu'
	La	Le	li	Lo	Lu	La'	Le'	li'	Lo'	Lu'

Another obviously important kind of unit consists of the most frequent graphical words, which normally correspond to function words: pronouns, determiners, pospositions (instead of prepositions), conjunctions, prominent adverbs, frequent nouns and adjectives (numbers, colors and kinship words). Some of the 98 of these items appear in Table 2.

By including function words in the database, a synthesizer can be developed with relative fewer distortions than one using merely diphones. Thus, these latter would be used to build non function words, *i.e.* content words. Content words in Raramuri, as mentioned above, exhibit suffix sequences of derivational and inflectional material. Since these sequences are to be expected in any Raramuri discourse,[5] it makes sense to include them as a third type of unit for the synthesizer. However, the language is not sufficiently studied and these sequences are not really known. Fortunately, diverse unsupervised methods for the discovery of morphemes exist that con be applied to a corpus in order to determine these suffix sequences.

Table 2. A few of the 98 function words of Raramuri

pronouns	determiners	pospositions	adverbs
nihé	ecí	yuwa	chabé
muhé	mí	hiti	sinibí
ecí	ná	jonsa	gará
tamuhé		okua	arigá
tumuhé		pacháami	wabé
yémi		mobá	wikabé

Once the units were identified, a native speaker recorded each one in a natural context. The resulting waveform was labelled and segmented in order to compile the database that is used for TTS synthesis.

3.1 Segmentation Methods

Many techniques for morphological segmentation exist.[6] Some interesting ones are minimal distance methods [14], bigram statistics [15], minimization of affix

[5] In essence, lexical items —specifically the root morphemes within them— are the carriers of discourse. They convey content information in action. Also, some morphological items, specifically modifiers, clitics and affixes, which are derivational and inflectional, carry the grammatical information that structures discourse. Hence, one might argue that the essence of language as a communication system —which is embodied in its repeatable patterns— resides in its structure or in the items which structure discourse, like affixes. Therefore, sequences of these can be taken as a promising unit for a synthesizer, while the roots of the words in which they appear can be build by means of diphones.

[6] There are several prominent approaches to word segmentation. The earliest one is due to Zellig Harris, who first examined corpus evidence for the automatic discovery

sets [16] and Bayesian statistics [17]. For the purposes of identifying from a corpus a set of suffix sequences of Raramuri, any of these methods can be applied. We used an economy-entropy based method which we have previously applied to Spanish [18]; Chuj (Mayan) [19], Czech [20] and Raramuri [21].

The approach proposed in this paper grades word substrings according to their likelihood of representing an affix or a valid sequence of affixes. The resulting candidates are gathered in a table for later evaluation by experts. In essence, two quantitative measurements are obtained for every possible segmentation of every word found in a corpus: Shannon's entropy [22] and a measure of sign economy [13] (which will be dealt with below). In short, the highest averaged values of these two measurements are good criteria to include word fragments as items in the table which will be called affix ***catalog***, ***i.e.*** a list of affix candidates and their entropy and economy normalized measurements, ordered from most to least affixal.

Information Content High entropy measurements have been reported repeatedly as successful indicators of borders between bases and affixes [18, 19, 23–25, 20]. These measurements are relevant because shifts of amounts of information can be expected to correspond to the amounts of information that a reader or hearer is bound to obtain from a text or spoken discourse. Frequent word fragments contain less information than those occurring rarely. Hence, affixes must accompany those segments of a text which contain the highest amounts of information.

Information content of a set of word fragments is typically measured by applying Shannon's method.[7] In order to identify affix sequences, the task is to measure the entropy of the word fragments which occur concatenated to a suffix candidate: where there is an actual morphological border, the content of information of stems with respect to their accompanying suffix sequences exhibits a peak of entropy. Specifically, looking for peaks of information means taking each right-hand substring of each word of the sample, determining the probabilities of everything that precedes it, and applying to these Shannon's formula to obtain the entropy measurements to be compared.

Economy Principle The other important measure used to identify Raramuri suffix sequences is based on the principle of economy of signs. In essence, we

of morpheme boundaries for various languages, [11]. His approach was based on counting phonemes preceding and following a possible morphological boundary: the more variety of phonemes, the more likely a true morphological border occurs within a word. Later, Nikolaj Andreev designed in the sixties the first automatic method based on character string frequencies which applied to various languages. His work was oriented towards the discovery of whole inflectional paradigms and applied to Russian and several other languages, [12]; and that of [13] in the seventies for French and Spanish.

[7] Recall the formula $H = -\sum_{i=1}^{n} p_i \log_2 p_i$, where p_i stands for the relative frequency of word fragment i [22].

can expect certain signs to be more economical than others because they relate to other signs in an economical way. Specifically, affixes combine with bases to produce a number (virtually infinite) of lexical signs. Although affixes do not combine with any base, certain ones combine with many bases, others with only a few. Nevertheless, it makes sense to expect more economy where more combinatory possibilities exist. This refers to the syntagmatic dimension. The paradigmatic dimension can also be considered: as they attach to bases, affixes appear in complementary distribution in a corpus with respect to other affixes (*i.e.* they alternate in that position). If there is a relatively small set of alternating signs which adhere to a large set of unfrequent signs the relations between the former and the latter must be considered even more economical.

The economy of segmentations can be measured by comparing the following sets of word fragments from each word of the corpus. Given a suffix candidate, there are two groups of word fragments:

1. *companions* — strings beginning graphical words which are followed by the given suffix sequence candidate (syntagmatic relation).
2. *alternants* — strings ending graphical word which occur in complementary distribution with the suffix sequence candidate.

Formally, let $A_{i,j}$ be the set of *companions* occurring, according to a corpus, along with word segment $b_{i,j}$. Let $A^p_{i,j}$ be the subset of $A_{i,j}$ consisting of the word beginnings which are quantitative prefixes of the language in question. Let $B^s_{i,j}$ be the set of word endings which are, also according to the corpus, suffixes of the language and occur in complementary distribution (*alternants*) with the word fragment $b_{i,j}$. One way to estimate the economy of a segmentation is:

$$k^s_{i,j} = \frac{|A_{i,j}| - |A^p_{i,j}|}{|B^s_{i,j}|} \tag{1}$$

In this way, when an right-hand word fragment is given, a very large number of companions and a relatively small number of alternants yield a high economy value. Meanwhile, a small number of companions and a large one of alternants indicate a low economy measurement. In the latter case, the word fragment in question is not very likely to represent exactly an affix, nor a sequence of them.

3.2 Building a Catalog of Raramuri Suffixes

The process to identify suffix sequences basically takes the words of the word sample and determines the best segmentation for each one using to the two measurements discussed above. Each best segmentation represents a hypothesis postulating a base and a suffix sequence. Thus, the presumed suffix sequence (and the values associated with it) are fed into a structure called Catalog.

The methods described above complement each other in order to identify Raramuri suffix sequences. Specifically, the values obtained for a given word fragment are normalized and averaged. That is, we estimated the ***suffixality*** of each sequence by means of the arithmetic average of the relative values of entropy and economy: $(\frac{h_i}{\max h} + \frac{k_i}{\max k}) * \frac{1}{2}$, where h_i stands for the entropy value associated to suffix candidate i; k_i represents the economy measurement associated to the same candidate; and $\max h$ returns the maximum quantity of h calculated for all suffixes (same idea for $\max k$).

As mentioned above, Raramuri is more an agglutinative language than a fusional one and word formation is mainly accomplished by means of suffixation. As could be expected, stems are followed by derivational suffixes, and these by inflectional ones. Since stems can be the result of other morphological processes, there might be morphemes to be discovered towards the beginning of words, but they are not necessarily affixal [21].

The corpus[8] corresponds to the Raramuri's variant from San Luis Majimachi, Bocoyna, Chihuahua. For today's corpora standards, this sample is a very small one, consisting of no more than 3,584 word-tokens and 934 word-types. Even though we cannot assume this sample's representativity of this variant, we proceeded to apply the method because it is robust for small corpora. Table 3 shows partial results of procedure.

Table 3. The 20 Most Affixal Raramuri Suffix Sequences

rank	**suffix**	**frec.**	**squares**	**economy**	**entropy**	**affixality**
1.	∼ma	35	1.00000	1.00000	0.88030	0.98050
2.	∼re	77	0.79960	0.81100	0.86060	0.82370
3.	∼sa	33	0.63640	0.93060	0.75590	0.77430
4.	∼ra	62	0.66130	0.64610	0.85080	0.71940
5.	∼si	28	0.75000	0.52570	0.83450	0.70340
6.	∼na	25	0.41140	0.72240	0.79840	0.64410
7.	∼go	4	0.21430	0.90650	0.64930	0.59000
8.	∼é	49	0.16620	0.43580	1.00000	0.53400
9.	∼ame	51	0.25210	0.30640	0.85910	0.47250
10.	∼gá	18	0.40480	0.37810	0.61360	0.46550
11.	∼ka	19	0.25560	0.28060	0.84130	0.45920
12.	∼á	67	0.13860	0.31330	0.91950	0.45710
13.	∼ré	11	0.16880	0.41020	0.73430	0.43780
14.	∼ga	50	0.18290	0.28340	0.80650	0.42430
15.	∼a	281	0.10520	0.18960	0.97250	0.42250
16.	∼ba	8	0.21430	0.30220	0.74000	0.41880
17.	∼ayá	8	0.21430	0.44320	0.57570	0.41110
18.	∼í	42	0.10200	0.26480	0.80540	0.39070
19.	∼či	39	0.10260	0.27510	0.74000	0.37260
20.	∼e	164	0.15240	0.29100	0.64290	0.36210

[8] Mainly texts collected by Patricio Parra.

Although Raramuri has only a few inflectional forms, the larger catalog exhibits more items containing inflectional material than were expected.[9] In fact, if inflectional suffixes are to be considered somehow more affixal than derivational ones, it should not be surprising to find the four most prominent Raramuri inflection affixes appear at the top of the table: *∼ma*, *∼re*, *∼sa*, and *∼si*, which mark tense, aspect and mode.

Using her own field work experience and taking into account the work of other experts, Alvarado determined the 35 most prominent nominal and verbal derivational suffixes for this language. 25 of these occurred within the first 100 catalog entries (a recall measure of 71% within this limit). The other entries are chains of suffixes (including sequences of derivational and inflectional items) and residual forms.[10] The 10 derivational suffixes which did not appear in the catalog are essentially verbal derivational forms, or modifiers of transitivity or some semantic characteristic of verbal forms. This might mean that the small sample used is more representative of nominal structures, rather than of verbal ones. These missing suffixes were added to the set of units to be processed for the synthesizer. Nevertheless, it is worth stressing that a significant part of the known Raramuri derivational system —essentially the nominal subsystem— was retrieved from a very small set of texts, which hardly constitutes a corpus of this language.

4 Stages of Text Processing for Unit Selection Synthesis

In general, the stages of text processing —which, as mentioned above, were achieved for the target language— are:

Transcription. It consists of a phonetic representation of the input text to be synthesized, keeping all punctuation and stress marks to preserve intonation clues; a new text (transcribed) will be created. It includes conversion of dates and numbers to a phonologic level.

Diphones division. The transcribed file is analyzed in order to extract its diphones, an output file is created with a list of them, organized from the most to least used. This program is sensitive to detect diphones according to their prosody and intonation. The extraction must be capable to identify the stressed syllable in each word according to the stress rules of the language (and assigns a stress mark to its related diaphone). It also takes into account which punctuation is adjacent to the last diphone to identify its intonation.

[9] This is certainly due to the fact that input texts are constituted by linguistic acts in the pragmatic act of narrating a story. Words appear therefore inflected. Obviously, using dictionary entries without inflection (lemma sets), rather than text in context, would be a much better way to obtain derivational items.

[10] The examination of residual items was especially difficult. Questions about lexicalized affixes (possibly fossilized items) and about the relationship between syllable structure and affix status emerged. These matters remain to be revised by Raramuri experts. Meanwhile, for evaluation purposes, entries with unexpected syllabic structure were not counted as acceptable suffixes nor valid sequences of them.

Most frequent words and affix sequences searching. The corpus used is the valuable source to consult the most common words in this language. Besides the diphones, the program must also be capable to identify these words.

Preliminaries of the recording session. Materials extracted from the corpus are compiled and arranged in contexts that the speaker could read in the recording session.

5 Processing

This module will be able to choose the set of units of the speech corpus, that better adjusts to a series of characteristics. The selection will be made so that it diminishes the total cost, sum of the unit costs and costs of concatenation between units. Equation 3 describes the difference between the target segment (u_i) and the candidate segment (t_i). Using equation 3 we get the concatenation cost, which reflects the smoothness of the concatenation between selected segments (u_{i-1}, u_i).

This module will be the one in charge to concatenate the different units that have been chosen by means of the selection algorithm. Consequently, it will be necessary to implement another module that obtains the phonetic marks of each file of wave from the corresponding curve. Computacional load will be reduced as posible for the developed programs. A strategy for the organization of the data base of units will be followed that allows to accelerate the searches.

$$C^t(t_i, u_i) = \sum_{j=1}^{p} w_j^t C_j^u(t_i, u_i) \tag{2}$$

$$C^c(u_{i-1}u_i,) = \sum_{j=1}^{q} w_j^c C_j^c(u_{i-1}, u_i) \tag{3}$$

Through the direct concatenation of units, we expect to get a good quality of synthesized speech because of the use of a large database and the definition of prosodic targets. Nevertheless, in order to increase synthesized speech quality, *i.e.* to make that the transitions between units are not perceivable, it will be necessary to make a processing on the result by means of algorithm TD-PSOLA.

This algorithm is used since it can be applied directly to the audio signal without the need for parametric extraction as is the case with LPC and other common algorithms used. For this algorithm to work we need to add a pitch-mark extraction phase to the database creation. This step is done offline so it carries no speed penalties during run-time.

For pitch-mark extraction we have used a dynamic programming based algorithm presented by Vladimir Goncharoff and Patrick Gries [26]. This algorithm was found to be very straightforward and highly reliable and gave out practically no extraction errors. An added bonus is that source code for the algorithm is distributed freely.

During run-time the speech signal is first divided into overlapping Hanning windowed pichmark-centered segments. The lengths of these windows must be larger than a pitch period and proportional to the pitch period. To achieve pitch modification these segments are aligned to the new position of the pitchmarks and the segments are then added together. A normalization values is also calculated from the Hanning window to eliminate energy modifications due to the overlapping [8].

It may also be necessary to duplicate or eliminate segments to maintain the duration of the signal at different pitchs or to accommodate time modification of the signal simultaneously to pitch modifications. To minimize discontinuities in the concatenation points we use this algorithm to modify the pitch of each segment, so as to make the pitch of both segments equal. The segments are then cropped so that their beginning and end correspond to a pitch mark and their magnitude is equalized [8].

Although this process is unable to eliminate all discontinuities these are greatly minimized, but at the cost of very little distortion compared to other OLA based systems with heavier processing and the processor load is also smaller to achieve acceptable speed in slower equipment

6 Closing Remarks

In this paper we have presented a method to develop a TTS synthesizer based on unit-selection for just about any language whose words are structured as a stem and a sequence of affixes, either derivational, inflectional or both. In this manner, building the utterances becomes a matter of selecting chains of function words, diaphones (to build stems) and suffix sequences to complete content words appearing in their contexts.

We expect this strategy to result in more natural speech than what could be expected using diphones alone. However, disadvantages remain in clearness, especially when comparing this simple system to top international systems. Nevertheless, this synthesizer uses a smaller amount of memory in comparison to those systems, which are tailored for some widely and very well known language like English or German.

Certainly, the system proposed can be improved in several ways, but the method can be readily applied to new languages in order to obtain relatively good synthesizers for them. Meanwhile, we are working to improve the clearness without naturalness degradation.

Acknowledgments. The work reported in this paper has been supported by a DGAPA UNAM grant for PAPIIT Project IN402008 *Glutinometría y variación dialectal.*

References

1. Campbell, N., Black, A.: Prosody and the Selection of Source Units for Concatenative Synthesis. In: Progress in Speech Synthesis. Springer Verlag (1995)
2. Hunt, A.J., Black, A.W.: Unit Selection in a Concatenative Speech Synthesis System using a large Speech Database. ATR Interpreting Telecommunications Research Labs
3. Zhao, Y., Liu, P., Li, Y., Chen, Y., Chu, M.: Measuring Target Cost in Unit Selection with kl-divergence Between Context-Dependent HMMs. Microsoft Research Asia, Beijing, China, 100080
4. Dutoit, T.: An Introduction to Text-to-Speech Synthesis. In: VoiceXML Review. Kluwer Academia Publishers, Netherlands (1997)
5. Huang, X., Acero, A., Hon, H.: Spoken Language Processing. Prentice Hall PTR (2001)
6. van Santen, J., Sproat, R., Olive, J., Hirschberg, J., eds.: Progress in Speech Synthesis. Springer (1997)
7. Black, A.W.: Speech Synthesis for Educational Technology. In: SLaTE Workshop on Speech and Language Technology in Education. (2007)
8. del Río, F., Herrera, A.: A Mexican Spanish Synthesis System Using a Pitch Syncrhonous Overlap Add. In: Proceedings of the IASTED International Conference on Signal and Image Processing. (2004)
9. Tatham, M., Morton, K.: Developments in Speech Synthesis. John Wiley, Chichester (2005)
10. Wightman, C.W., Syrdal, A.K., Stemmer, G., Conkie, A., Beutnagel, M.: Perceptually Based Automatic Prosody Labeling and Prosodically Enriched Unit Selection Improve Concatenative Text-to-Speech Synthesis. In: ICSLP 2000. Volume II., Beijing (October 2000) 71–74
11. Harris, Z.S.: From Phoneme to Morpheme. Language **31**(2) (1955) 190–222
12. Cromm, O.: Affixerkennung in deutschen Wortformen. Eine Untersuchung zum nicht-lexikalischen Segmentierungsverfahren von N. D. Andreev. Abschluß des Ergänzungsstudiums Linguistische Datenverarbeitung, Frankfurt am Main (1996)
13. Kock, J.d., Bossaert, W.: The Morpheme. An Experiment in Quantitative and Computational Linguistics. Van Gorcum, Amsterdam, Madrid (1978)
14. Goldsmith, J.: Unsupervised Learning of the Morphology of a Natural Language. Computational Linguistics **27**(2) (2001) 153–198
15. Kageura, K.: Bigram Statistics Revisited: A Comparative Examination of Some Statistical Measures in Morphological Analysis of Japanese Kanji Sequences. Journal of Quantitative Linguistics **6** (1999) 149–166
16. Gelbukh, A., Alexandrov, M., Han, S.Y.: Detecting Inflection Patterns in Natural Language by Minimization of Morphological Model. In: Congreso Iberoamericano de Reconocimiento de Patrones, CIARP-2004. LNCS (2004)
17. Creutz, M., Lagus, K.: Inducing the Morphological Lexicon of a Natural Language from Unannotated Text. In: Proceedings of the International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning (AKRR'05), Espoo, Finland (June 2005)
18. Medina-Urrea, A.: Automatic Discovery of Affixes by Means of a Corpus: A Catalog of Spanish Affixes. Journal of Quantitative Linguistics **7**(2) (2000) 97–114
19. Medina-Urrea, A., Buenrostro Díaz, E.C.: Características cuantitativas de la flexión verbal del chuj. Estudios de Lingüística Aplicada **38** (2003) 15–31

20. MEDINA-Urrea, A., HLAVÁČOVÁ, J.: Automatic Recognition of Czech Derivational Prefixes. In: Proceedings of CICLing 2005. Volume 3406 of Lecture Notes in Computer Science. Springer, Berlin/Heidelberg/New York (2005) 189–197
21. MEDINA-Urrea, A., ALVARADO-García, M.: Análisis cuantitativo y cualitativo de la derivación léxica en ralámuli. In: Primer Coloquio Leonardo Manrique, Mexico, Conaculta-INAH (September 2004)
22. Shannon, C.E., Weaver, W.: The Mathematical Theory of Communication. University of Illinois Press, Urbana (1949)
23. Hafer, M.A., Weiss, S.F.: Word Segmentation by Letter Successor Varieties. Information Storage and Retrieval **10** (1974) 371–385
24. Frakes, W.B.: Stemming Algorithms. In Frakes, W.B., Baeza-Yates, R., eds.: Information Retrieval, Data Structures and Algorithms. Prentice Hall, New Jersey (1992) 131–160
25. Oakes, M.P.: Statistics for Corpus Linguistics. Edinburgh University Press, Edinburgh (1998)
26. GONCHAROFF, V., GRIES, P.: An algorithm for accurately marking pitch pulses in speech signals. In: International Conference Signal and Image Processing. (1998)

Pronunciation Rules in Portuguese Regional Speech (PORT REG) for Coarticulation Process

Sara Candeias[1] and Jorge Morais Barbosa [2]

[1] Instituto de Telecomunicações, Department of Computers and Electrical Engineering, University of Coimbra, PORTUGAL
[2] Departement of Portuguese Language, Faculty of Letters, University of Coimbra, PORTUGAL
saracandeias@co.it.pt, jbarbosa@ci.uc.pt

Abstract. This paper describes one aspect of an ongoing work to incorporate pronunciation variability in the Portuguese (PORT) speech system. This work focuses on the linguistic rules to improve the grapheme-(multi)phone transcription algorithm that will be implemented. Portuguese 'Beira Interior' regional speech (PORT-BI REG) is considered to be in the realm of coarticulation (post-lexical) phenomena. A set of linguistic rules for most of the common vowel transformation in an utterance (vocalic segments at both the left and right edges of the word) is presented. The analysis focuses on the distinctive features that originate vowel sound challenges in connected speech. The results are interesting from the point of view of setting up models to reconstruct a grapheme-phone transcription algorithm for Portuguese multi-pronunciation speech systems. We propose that the linguistic documentation of Portuguese minority speech can be an optimal start for Portuguese speech system development process, too.

Keywords: Text-to-Speech; coarticulation (phonology); structural analysis (linguistic features); pronunciation instruction (phonetic).

1 Introduction

Several frameworks have been proposed for the grapheme-to-phone transcription module for Portuguese language, such as [2, 3, 12]. However, the problem with the Portuguese regional speech under development is the shortage of speech and text corpora. This is one of the reasons why their linguistic structure has been very poorly investigated, especially at linguistic levels such as phonetics. The applications of the Portuguese speech system are mainly based on standard Portuguese language and on isolated word recognition. It is well known that the sequence of phones spoken by a human speaker is not the same sequence as that which derives from the phonetic transcription of a word in isolation. Coarticulation (post-lexical) rules must be included in the course of phonetic transcription. In order to obtain a more natural speech, these rules must be applied to varying sequences of phones. Several methods can be used to elicit grapheme-to-phoneme rules from pre-existing lexicons. However, these automatic techniques do not cope very well with the concurrent multi-

Received 09/01/09
Accepted 09/01/09
Final version 19/02/09

pronunciations that arise from coarticulation phenomena. Specifically, word boundary events relating to the normalization of differing pronunciations have to be accounted for. One reason for studying PORT-BI REG pronunciations deriving from vocalic segment boundaries is that these pronunciations must particularly be considered at some computational cost in Portuguese speech technology. As phoneticians, we argue that speech systems would benefit from fine detail in the vicinity of segment boundaries, especially when our goal is minority Portuguese speech (pronunciation) development.

This paper has five sections. First we present the methodology used to investigate the phonological features of vowels produced in continuous speech. Section 2 describes the corpus collected for this study, including recording conditions and analysis parameters. Section 3 examines the results concerning coarticulation (post lexical rules) in different vowel contexts. Section 4 summarizes the analysis. Section 5 contains our main conclusions.

2 Methodology

2.1 Corpus Constitution

Fist, we collected sentences of varying lengths and with a specific patter, comprising a vowel segment in word-final position followed by another vowel segment in the inception of the following word. Second, we organized the vowels segments according to a post-vowels context. The 45 cases measured show that there are a number of surface representations demonstrating the application of coarticulation (post-lexical) rules.

2.2 Recording Condition

Our aim in collecting the corpus was to obtain examples of pronunciation variation that are suitable for inventorying. So, sentences were recorded from various Portuguese pronunciation sources (particularly Beira Interior Portuguese region speakers) using a portable minidisc recorder Sony MZ-R700PC. A Panasonic unidirectional microphone recorded directly onto a minidisc, and recording was carried out in surroundings satisfactory that were adequately noise cleaned. Speakers were of both sexes, natives, over 45 years of age, had poor schooling levels and good dental and mouth cavity configuration.

2.3 Speech Analysis

A perceptive experience model was used for this study, and distinctive feature analysis was implemented. This methodological option for speech research lies between phonological transcription and perceptual phonetics. It integrates phone relations and their distinctive value in a closer rendering of the physical reality of the spoken language. In accordance with the concept of functional phoneme [1, 7, 11], our phones-inventory was built up on the basis of perceptive criteria. We assumed a perceptive analysis as an operational stage of describing phones and we established it as: a) a result of stimulus/percept dichotomous process, which is correlated with the reviewing capabilities of discrimination, identification and perceptual constancy [13]; and b) a process correlated with an acoustic signal's perceptual activity (inferior level) in which most central structures of linguistic events (superior level) are implicated [5, 8, 4]. Based on these guidelines, also assumed by the economic/optimal theory of language [1, 7, 11, 10, 6], we accepted the distinctive phone as the event identified by perception in the speech in continuous.

3 Analysis

This particular analysis of phonetic coarticulation aims at describing what happens when two vowels come together in continuous speech. In theory, there is still some disagreement on the exact form of the set of features required to describe the sound patterns that occur in languages. We have taken for granted the basic proposals arising from the standard set of Portuguese phonological rules, i.e. those applying to stress or unstressed vowels in the syllabic position [11]. Accordingly, we examined types of phonological patterns that have been associated with the perceptual properties of speech sounds (see 2.3 above). In addition, the effects on the linearity parameter of utterance were analyzed and discussed. Those phonetic modifications to vowels only occur physically if there is a prosodic connection between successive words [9].

We argue that the vulnerability of the Portuguese speech system is due to vowel coarticulation being excluded and the consequent unnatural quality. In real speech there is a definite relationship between movements of the vocal tract and the properties of the emitted sound. It follows that if Portuguese speech analysis included fine and systematic phonetic variation, the intelligibility of the Portuguese speech system should increase considerably.

When two vowel segments not pertaining to the same word are pronounced together, our analysis shows some perceptive architectures as a result: the last sound of the first vowel may be affected by the second sound of the next vowel, coalescing with it, or becoming shorter or being deleted (as described in Section 4).

4 Results

Findings for vowel sound transformation in connected speech are presented in the following tables. Some results show a tendency towards dissimilation (progressive or regressive), others confirm a tendency for linking. All these results reveal a phonetic description of adjunction phenomena.

Because Portuguese orthography does not reflect most vowel sound changes, these results can be regarded as phone rules, to be mobilized in the recognition or synthesis module.

Let us present a few points about the results in the tables. When the final vowel-grapheme of the first word may be pronounced [6] or [@] and the following vowels are in stressed position, vowels [6] and [@] are deleted, as shown in Tables 1 and 3.

If we examine Table 2, we see that two situations emerge when the final vowel-grapheme of the first word is pronounced as [6] and is followed by an unstressed vowel: if the initial vowel-phone of the second word is an [e], [O], [o] or an [u], [6] is deleted; if the initial vowel-phone of the second word is another [6], the surface sound becomes [+low] [+back] as [a].

The vowel-grapheme emerging as [@] behaves differently depending on the stressed syllable of the following vowel. If the following vowel is in unstressed position, it is subject to a specific rule: [@] becomes [j] configuring a diphthong with the subsequent vowel. An exception occurs when the next vowel may be pronounced as [e] or an [6]. Given that structure, the rule observed is to delete [@] (Table 4).

If the final vowel is [u], it becomes [w], i.e., that sound emerges as glide and forms a diphthong with the following vowel, unless this second vowel is another [u]. In that case, the surface sound emerges as [u:] – compare the examples in Tables 5 and 6.

Finally, with respect to the final vowel-grapheme of the first word when it is pronounced as [i], we observe a paradigm similar to that detected in the context of [u] plus [unstressed vowel]: the final sound is pronounced and a glide [j] forms a diphthong with the following vowel. If this second vowel is a new [i], the surface sound grows as [i:] (Table 7).

Table 1. [6] final-vowel preceding prominent vowel contexts.

first word		**second word**	**examples**	**result**
vowel-final position		**vowel-initial position**		
6	**one syllable word**	**stressed position**		
		a	*ainda há*	**a**
	two or more syllable word	a	*grita alto*	**a**
		E	*para ela*	**E**
		O	*na hora*	**O**
		u	*na uva*	**u**

Table 2. [6] final-vowel preceding non-prominent vowel contexts.

first word		**second word**	**examples**	**result**
vowel-final position		**vowel-initial position**		
6	**one syllable word**	**unstressed position**		
		6	*toda a*	**a**
	two or more syllable word	6	*grita anita*	**a**
		e	*da enamorada*	**e**
		O	*para orar*	**O**
		o	*para ornamento*	**o**
		u	*para utensílio*	**u**

Table 3. [@] final-vowel preceding prominent vowel contexts.

first word		**second word**	**examples**	**result**
vowel-final position		**vowel-initial position**		
		stressed position		
@	**one syllable word**	A	*ligue à*	**a**
		E	*que é*	**E**
		e	*se eu*	**e**
		O	*ligue ao*	**O**
	two or more syllable word	a	*de abas*	**a**
		E	*que era*	**E**
		e	*que eu*	**e**
		O	*que horas*	**O**
		o	*de hoje*	**o**
		u	*de uvas*	**u**

Table 4. [@] final-vowel preceding non-prominent vowel contexts.

first word		**second word**	**examples**	**result**
vowel-final position		**vowel-initial position**		
		unstressed position		
@	**one syllable word**	@	*ligue a*	**j@**
		u	*ligue o*	**jw**
	two or more syllable word	@	*de anão*	**j@**
		a	*se amanhã*	**ja**
		e	*de enamorar*	**je**
		e~	*se embora*	**je~**
		6	*de estar*	**6**
		O	*de otorrino*	**jO**
		u	*de união*	**ju**

Table 5. [u] final-vowel preceding prominent vowel contexts.

first word		**second word**	**examples**	**result**
vowel-final position		**vowel-initial position**		
		stressed position		
u	**one syllable word**	a	*do ar*	**wa**
		E	*como é*	**wE**
		e	*como eu*	**we**
		a	*como ao*	**wO**
	two or more syllable word	E	*como ela*	**wE**
		e	*como ele*	**we**

Table 6. [u] final-vowel preceding non-prominent vowel contexts.

first word		**second word**	**examples**	**result**
vowel-final position		**vowel-initial position**		
		unstressed position		
u	**one syllable word**	@	*lavo-a*	**w@**
		u	*lavo-o*	**u:**
	two or more syllable word	@	*do amarelo*	**w@**
		a	*do actor*	**wa**
		6	*do estado*	**w6**
		u	*todo unificado*	**u:**

Table 7. [i] final-vowel preceding non-prominent vowel contexts.

first word		**second word**	**examples**	**result**
vowel-final position		**vowel-initial position**		
		unstressed position		
i	**two or more syllable word**	@	*taxi amarelo*	**j@**
		i	*júri irónico*	**i:**
		u	*júri ucraniano*	**ju**

5 Conclusions

This paper describes a set of rules consisting of phone modification phenomena that occur as a by-product of coarticulation effects in PORT-BI REG connected speech. Vowel (pseudo-)phones are created to model the coarticulation (post-lexical) phenomena.

We assume that segments are made up of distinctive features. This view makes it possible to group features into larger sets that can act together to develop Portuguese speech knowledge methods.

These findings also complement, along with Portuguese alternative pronunciation, the information on prosodic phenomena described in Portuguese grammars.

5.1 Future Developments

Although the method described is consistent, that is not a strong enough reason for not implementing and testing new methods in the future. The analysis of PORT-BI REG will be extended to other varieties of Portuguese speech. We plan to conduct tests with other corpora.

In addition to being a study on this variability, a full explanation of coarticulation phenomena also needs to be based on an examination of how syntactic and phonological rules interact.

Acknowledgments. This paper has benefited from motivating suggestions received from Instituto de Telecomunicações – DEEC/University of Coimbra team. We would particularly like to thank to Fernando Perdigão for the stimulus.
This research was funded by SFRH/BPD/36584/2007 of the Portuguese Foundation of Science and Technology.

References

1. Martinet, A.: Eléments de Linguistique Générale. Armand Colin, Paris (2001)
2. Teixeira, A., Oliveira, C., Moutinho, L. C.: On the Use of Machine Learning and Syllable Information in European Portuguese Grapheme-Phone Conversion. In: PROPOR 2006, pp. 212-215 (2006).
3. Braga, D., Coelho, L., Resende Jr., F. G. V.: A Rule-Based Grapheme-To-Phone Converter for TTS Systems in European Portuguese. In: Telecommunications Symposium, 2006 International. Fac. of Arts of Univ. of La Coruna, La Coruna (2006)
4. Pisoni, D. B.: Perceptual Evaluation of Synthetic Speech: What Have We Learned Over the Last 15 Years and Where Are We Going in the Future?. In: Second ESCA/IEEE Workshop on Speech Synthesis: 215. Mohonk Mountain House, New Paltz, NY, USA (1994)
5. Fry, D. B.: Speech Reception and Perception. News Horizons in Linguistics, pp. 29-52. Penguin Books, London (1970)
6. McCarthy, J.: A Thematic Guide to Optimality Theory. Cambridge University Press, Cambridge (2001)
7. Barbosa, J. M. B.: Introdução ao Estudo da Fonologia e Morfologia do Português. Almedina, Coimbra (1994)
8. McCarthy, J.: A Thematic Guide to Optimality Theory. Cambridge University Press, Cambridge (2001)
9. Vigário, M.: The Prosodic Word in European Portuguese. Mouton de Gruyter, Berlin/New York (2003)
10. Kager, R.: Optimality Theory. Cambridge University Press, Cambridge (1999)
11. Candeias, S. M. F. R. e C. M.: Sistema Fonológico da Beira Interior e Algumas Considerações Sintáctico-Semânticas. PhD dissertation. University of Aveiro, Aveiro (2007)
12. Paulo, S., Oliveira, L. C., Mendes, C. M. D., Figueira, L. A. D. M., Cassaca, R. M. F., Ribeiro, M. C. G. V., Moniz, H. G. S.: Dixi – A Generic Texto-to-Speech System for European Portuguese. In: PROPOR 2008, pp. 91-100 (2008)
13. Harnad, S. (eds.): Categorical Perception – the Groundwork of Cognition. Cambridge University Press, Cambridge (1987)

A Maximum Entropy (ME) Based Translation Model for Chinese Characters Conversion

Fai Wong, Sam Chao, Cheong Cheong Hao and Ka Seng Leong

Faculty of Science and Technology of University of Macau,
Av. Padre Tomás Pereira S.J., Taipa, Macao
{derekfw, lidiasc}@umac.mo

Abstract. As the growth of exchange activities between four regions of cross strait, the problem to correctly convert between Traditional Chinese (TC) and Simplified Chinese (SC) is getting important and attention from many people, especially in business organizations and translation companies. Different from the approaches of many conventional code conversion systems, which rely on various levels of human constructed knowledge (from character set to semantic level) to facilitate the translation purpose, this paper proposes a Chinese conversion model based on Maximum Entropy (ME), a Machine Learning (ML) technique. This approach uses tagged corpus as the only information source for creating the conversion model. The constructed model is evaluated with selected ambiguous characters to investigate the recall rate as well as the conversion accuracy. The experiment results show that the proposed model is comparable to the state of the art conversion system.

Keywords: Maximum Entropy, Machine Learning, Natural Language Processing, Chinese translation, Traditional Chinese, Simplified Chinese.

1 Introduction

Modern Chinese typically involves two main dialects of writing, Traditional Chinese (TC) and Simplified Chinese (SC). In Chinese computing, these two systems adapt different coding schema for the computer to process the corresponding Chinese information. Traditional Chinese uses Big5 encoding while Simplified Chinese uses GB. For a Simplified Chinese document to be opened and read in a computer with Traditional Chinese operating system, conversion from Simplified Chinese encoding system into Traditional Chinese encoding is necessary for the purpose that the document can be further processed under the Traditional Chinese computer environment, and vice versa. As addressed by Wang [1] in the meeting of the 4th Chinese Digitization Forum, although there are many conversion systems implemented and available in the market, neither one of them can produce the conversion result with satisfaction. Reviewing the nature of this problem, Simplified Chinese is actually a simpler version of Traditional Chinese. It differs in two ways from the Traditional writing system: 1) a reduction of the number of strokes per character and 2) the reduction of the number of characters in use that is two different

© A. Gelbukh (Ed.)
Advances in Computational Linguistics.
Research in Computing Science 41, 2009, pp. 267-276

Received 06/11/08
Accepted 11/12/08
Final version 02/02/09

characters (of TC) are now written with the same character (in SC). The relationship between these two writing systems is not one-to-one mapping. In numerous situations, one simplified character corresponds to two or more traditional forms, e.g. simplified "发" maps to traditional "發(emit)" and "髮(hair)". Normally only one of these is the correct one depending on the context. In some cases, one simplified character may map to multiple traditional forms, e.g. the simplified character "表" of the fragment " 有表" may map to traditional "表(form)" and "錶(watch)" and any of which may be correct according to context. There are hundreds of simplified characters which correspond to two or more traditional ones, leading to ambiguity and this is the main obstacle of the conversion task for Simplified Chinese to Traditional Chinese translation.

The conventional techniques used to automatically translate Simplified Chinese to Traditional Chinese can be classified into three different approaches [2]: code conversion, orthographic (dictionary) conversion and lexemic conversion. Code conversion is also known as character based substitution, where the code of one character set is being substituted with a target code of another character set based on mapping table between the GB and Big5 encoding systems. This straightforward conversion methodology produces most unreliable result since the mapping table translates each simplified character to one target traditional character only and ignores the other possible candidates, this frequently results in incorrect conversion. The orthographic approach does the conversion based on larger unit of compound characters instead of single character by looking up the unit from a mapping table (simplified - traditional lexicon). The unit can be a meaningful character or combination of characters, and even idiomatic phrases. This method relies on a sophisticated Chinese word segmenter [3] that identifies the boundaries of words from the stream of text before the conversion of correspondences between simplified and traditional units taken place. The conversion system developed by Xing et al. [4] is based on this paradigm. The third approach is based on lexemic conversion. This kind of conversion systems actually covers the conversion processes of orthographic and code conversions, and in addition, the system also takes the deviations of terminologies and words used for the same concept into consideration during the conversion process, e.g. in Simplified Chinese, the word computer is written as "计算机", while in Traditional Chinese, it is written as "電腦". The systems reported by Halpern et al. [2] and Xing et al. [5] are based on this conversion methodology, including the conversion tool provided in Microsoft Word [6].

However, these approaches suffer from several limitations: 1) they highly rely on human constructed knowledge from lexicon to semantic level in order to achieve high conversion accuracy. The creation of these kinds of knowledge is too labor-intensive and time-consuming. 2) Consistency of knowledge formulated in rule is difficult to maintain and sometimes could contradict with each other and thus, affect the overall system performance. In this work, we formulate the Chinese conversion as a sequential tagging problem and use a supervised machine learning (ML) technique, Maximum Entropy (ME), to construct a Chinese conversion system. The ME model is a kind of feature-based model which is flexible to include arbitrary features to help in selecting the correct correspondence for simplified character during the conversion. The major features of this model are the tags and context words from a sentence.

This paper is organized as follows. Section 2 presents the general model of Maximum Entropy. Section 3 discusses the modeling of Chinese conversion problem, and the formulation of features for constructing the ME-based conversion model will be discussed in Section 4. The experiments based on the real text collected from newspapers will be discussed in Section 5 and Section 6, followed by a conclusion to end this paper.

2 Maximum Entropy Modeling

Maximum Entropy was first presented by Jaynes and has been applied successfully in many natural language processing (NLP) tasks[7], such as Part-of-Speech (POS) tagging [8], word sense disambiguation [9], and Chinese word segmentation [3]. ME model is a feature-based probabilistic model which bases on history and is able to flexibly use arbitrary number of context features (unigram, bigram word features and tag features) to the classification process that other generative models like N-gram model, Hidden Markov Model (HMM) cannot. The model is defined over $X \times Y$, where X is the set of possible histories and Y is the set of allowable outcomes or classes for the token or character in our case of Chinese conversion problem. The conditional probability of the model of a history x and a class y is defined as:

$$p_\lambda(y \mid x) = \frac{\prod_i \lambda_i^{f_i(x,y)}}{Z_\lambda(x)} \quad . \tag{1}$$

$$Z_\lambda(x) = \sum_y \prod_i \lambda_i^{f_i(x,y)} \quad . \tag{2}$$

where λ is a parameter which acts as a weight for the feature in the particular history. The equation (1) states that the conditional probability of the class given the history is the product of the weightings of all features which are active under the consideration of (x, y) pair, normalized over the sum of the products of the weightings of all the classes given the history x as the equation (2) above. The normalization constant is determined by requiring that $\sum_y P_\lambda(y \mid x) = 1$ for all x.

In ME model, the useful information to predict the outcome y by the equation (1) based on history features is represented by binary feature functions $f()$. Given a set of features and a training corpus, the ME estimation process produces a model which allows us to compute the conditional probability of equation (1). This actually is the process to seek for the optimized set of weighting parameters λ that is associated with the features. In other words, the process is to maximize the likelihood of the training data using p:

$$L(p)=\prod_{i=1}^{n}p_{\lambda}(x_i,y_i)=\prod_{i=1}^{n}\frac{1}{Z_{\lambda}(x)}\prod_{j=1}^{m}\lambda_j^{f_j(x_i,y_i)} \quad . \tag{3}$$

A number of models can be qualified from Equation (3). But according to the ME principle, the target is to generate a model *p* with the maximum conditional entropy *H*(*p*):

$$H(p)=-\sum_{x\in X,y\in Y}p(x,y)\log p(x,y) \text{ where } 0 <= H(p) <= \log |y| \; . \tag{4}$$

3 Chinese Conversion as Tagging Problem

To model the Chinese conversion as a tagging problem, a manually tagged corpus with mapping relationships between simplified character and traditional character is required for training the conversion model based on the Maximum Entropy framework. In this work, we treat each character as a token, and it is assigned with a label sequence number, which represents the corresponding character in Traditional Chinese. For example, the simplified character "发" may map to "發(emit)" and "髮(hair)" in traditional forms. Thus in the labeled format, each ambiguous simplified character is assigned a number representing the mapping character in traditional one, as shown in Fig. 1. In the sentence, there are three ambiguous characters "发$_1$", "发$_2$" and "脏", and their corresponding traditional characters are "發(emit)", "髮(hair)" and "髒(dirty)", and are represented by the sequence number, "/1", "/2" and "/2" for each character, while the other unambiguous characters, including the punctuation marks, is assigned with "/0". The sequence number starts from 1 for each ambiguous character, until *n*, the possible number of candidates in the traditional forms. **Table 1** gives some exampled simplified characters and its correspondences in traditional form together with sequence number.

发/1 ！/0 你/0 的/0 头/0 发/2 有/0 点/0 脏/2 。/0

(Fat! Your hair is dirty.)

Fig. 1. The format of labeled sentence.

Based on tagged corpus, context information and features are collected to encode the useful information for the tagging process. In the model trained with suitable context and features, given a simplified sentence, it is able to predict each character with sequence number as the possible outcome from the tag set.

Table 1. Example of simplified characters with its possible corresponding traditional forms and sequences defined in our model.

板 → (1)板, (2)闆	参 → (1)參, (2)蔘
辟 → (1)辟, (2)闢	尝 → (1)嘗, (2)甞, (3)嚐
表 → (1)表, (2)錶	厂 → (1)厂, (2)庵, (3)廠, (4)厰
别 → (1)别, (2)彆	冲 → (1)沖, (2)衝
并 → (1)并, (2)並, (3)併, (4)竝	虫 → (1)虫, (2)蟲
卜 → (1) 卜, (2)蔔	丑 → (1)丑, (2)醜
布 → (1)布, (2)佈	仇 → (1)仇, (2)讎
才 → (1)才, (2)纔	出 → (1)出, (2)齣
采 → (1)采, (2)埰, (3)寀, (4)採	呆 → (1)呆, (2)獃
彩 → (1)彩, (2)綵	当 → (1)當, (2)噹

4 Feature Description

An important issue in the implementation of Maximum Entropy framework is the form of the function which calculates each feature. These functions are defined in the training phase and depend upon the data in the corpus. The function takes the form of Equation (5) as shown below, which is a binary-valued function:

$$f(x,y)=\begin{cases}1 & \mathit{if}\ y'=y\ \ \mathit{and}\ \ \mathit{info}(x)=v\\ 0 & \qquad\quad \mathit{otherwise}\end{cases} \qquad (5)$$

Where *info*(*x*) would be substituted with different expressions, and is referring as feature template in our work, which focuses on specific interested property that can be found from the context *x*, and *v* is a predefined value. For example, if we consider that 0 is the position of the active character, say "发" from the context "你的头发有点脏 (Fat! Your hair is dirty.)", to be learned and that *i* is related to 0, then the previous character of it is "头(head)" given by expression *PrevChr*(*x*,-1)="头". The set of features defined for the training of the conversion system mainly focus on characters, and collocations in the local context. In this work, two feature templates are adapted: C_i (i = -2 to 2), and C_iC_{i+1} (i = -2 to 1). Here C_0 represents the current character; C_i/C_{-i} represents the character which is at the i^{th} position to the right/left of C_0. These templates are basically character based features. They capture the contexts of surrounding information regarding the current character, including the form of character itself, which is also considered to the construction for the conversion model. Actually, each template groups several sets of features. Take the character sequence "你的头发有点脏" as an example, features that will be generated by Equation (5) based on the first template are: C_{-2} = “的”, C_{-1} = “头”, C_0 = “发”, C_1 = “有” and C_2 = “点”. On the other hand, features obtained based on the second template consist of: $C_{-2}C_{-1}$ = “的头”, $C_{-1}C_0$ = “头发”, C_0C_1 = “发有”, C_1C_2 = “有点”. Therefore, for each context, there will be 10 different features in total obtained and used in the training the model based on the data in the corpus.

5 Data Preparation

In order to evaluate the proposed model, we need a corpus for constructing the model, especially with tagged information. This step involves the preparation of training data and test data. Since there is no any corpus intended for the purpose of Chinese conversion from Simplified Chinese to Traditional Chinese, we prepare these data by ourselves for this evaluation purpose. In this work, both the training and test data are created based on ambiguous simplified characters and their correspondences of traditional characters. For each traditional character that corresponds to an ambiguous character in its simplified format, we collect the related sample fragments of sentences from the online corpus of *Chinese Character Frequency Statistics*[10]. The corpus covers the articles from mainland China, Taiwan and Hong Kong, of different time frames from 60’s to 90’s. Basically, we can obtain enough data for majority of the ambiguous characters. For the other characters that are “ancient” or infrequently uses, we try to search from both the Internet and dictionaries. The idea is to collect enough text or examples for all characters. Figure 2 shows the sample of collected sentence fragments for the traditional character “板(plank)” that forms the training corpus to be used for constructing the translation system.

郁的她呆板的团团的
好地方桥板并不比街
一块腊笔板随时计价
负面的刻板印象本调
抽完一斗板烟时我离
弱内的表板没有转数
杀向天花板然后像溃
臂踩得吊板吱吱格格
了云石地板的键房门
林的布景板推倒在一

Fig. 2. The fragments of sentences containing the traditional character “板(plank)”.

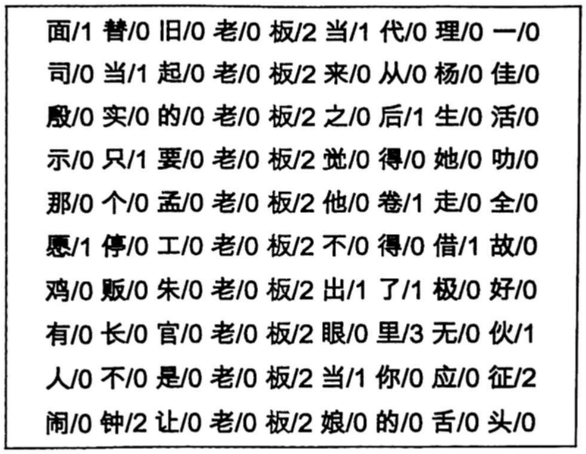
郁/1 的/0 她/0 呆/1 板/1 的/0 团/1 团/1 的/0
好/0 地/0 方/0 桥/0 板/1 并/2 不/0 比/0 街/0
一/0 块/0 腊/2 笔/0 板/1 随/0 时/0 计/0 价/0
负/0 面/1 的/0 刻/0 板/1 印/0 象/0 本/0 调/0
抽/0 完/0 一/0 斗/2 板/1 烟/0 时/0 我/0 离/0
弱/0 内/0 的/0 表/2 板/1 没/0 有/0 转/0 数/0
杀/0 向/1 天/0 花/0 板/1 然/0 后/1 像/0 溃/0
臂/0 踩/0 得/0 吊/2 板/1 吱/0 吱/0 格/0 格/0
了/1 云/1 石/0 地/0 板/1 的/0 铺/0 房/0 门/0
林/0 的/0 布/2 景/0 板/1 推/0 倒/0 在/0 一/0

Fig. 3. The processed fragments for the character "板(plank)" after adding related tag information to the characters.

The next step is to convert the corpus by adding related tag information that is the corresponding sequence number to each character as described in Section 3, shown in Fig. 3. From the sample fragments, the unambiguous characters are labeled with "/0", while the ambiguous ones are marked with sequences number representing to its character correspondence in the traditional format. Fig. 4 and Fig. 5 present the set of collected and processed data fragments for another possible translation of simplified character "板" in its traditional form "闆(boss)". In this case, the character is marked with "/2" instead of "/1" as in previous case.

面替旧老板当代理一
司当起老板来从杨佳
殷实的老板之后生活
示只要老板觉得她叻
那个孟老板他卷走全
愿停工老板不得借故
鸡贩朱老板出了极好
有长官老板眼里无伙
人不是老板当你应征
闹钟让老板娘的舌头

Fig. 4. The fragments of sentences for the traditional character "闆(boss)".



面/1 替/0 旧/0 老/0 板/2 当/1 代/0 理/0 一/0
司/0 当/1 起/0 老/0 板/2 来/0 从/0 杨/0 佳/0
殿/0 实/0 的/0 老/0 板/2 之/0 后/1 生/0 活/0
示/0 只/1 要/0 老/0 板/2 觉/0 得/0 她/0 叻/0
那/0 个/0 孟/0 老/0 板/2 他/0 卷/1 走/0 全/0
愿/1 停/0 工/0 老/0 板/2 不/0 得/0 借/1 故/0
鸡/0 贩/0 朱/0 老/0 板/2 出/1 了/1 极/0 好/0
有/0 长/0 官/0 老/0 板/2 眼/0 里/3 无/0 伙/1
人/0 不/0 是/0 老/0 板/2 当/1 你/0 应/0 征/2
闹/0 钟/2 让/0 老/0 板/2 娘/0 的/0 舌/0 头/0

Fig. 5. The fragments for the character "■(boss)" after processed.

Table 2 gives the size of the data set used for training the model. Actually, there are more than 300 interested characters that may cause ambiguity during the translation between traditional and simplified forms. Therefore, the collection of data is based on this set of characters. For each of these characters, a number of sentences are gathered to train up a conversion model for the disambiguation purpose when a simplified character is going to be converted into the traditional form.

Table 2. Size of training corpus

	Characters	Ratio
Size	919215	92.29%
Ambiguous Characters	70839	7.71%

For the test data, sentences are collected from several online Chinese newspapers of *Jornal Cheng Pou* (Cheng Pou Journal), *Jornal Cidadão* (Citizen Journal), *Jornal Informação* (Information Journal), *Jornal San Wa Ou* (San Wa Ou Journal), *Jornal Tai Chung* (Tai Chung Journal) and *Macau Daily News* (Macao Daily News) between 8th April 2008 and 8th August 2008. There are around 3,027 sentences in total. The data covers most of interested ambiguous characters. The relative data set size is presented in Table 3. This includes the count of all characters, as well as the ambiguous characters for testing.

Table 3. Size of test corpus

	Characters	Ratio
Size	862586	98.05%
Ambiguous Characters	16795	1.95%

6 Model Evaluation

Two experiments are carried out to investigate the recall rate and the conversion accuracy of the model. In both cases, only the counts of ambiguous characters are used for calculating the recall and precision, and excluding out the counts of unambiguous characters. Otherwise, the system will always obtain very high conversion accuracy, since the percentage of unambiguous characters is much higher than that of the ambiguous ones, as illustrated in Table 2 and Table 3 for different corpora.

The first experiment evaluates the recall rate. The model is trained and tested by using the training data as presented in Table 2. That is, the same data set is used to evaluate the performance of the model. The conversion accuracy (recall rate) is 99.84%.

In the second experiment, we construct the model based on the training data set and use another data set (test data) to evaluate the model's conversion precision. The accuracy of the conversion results reaches 89.94%. In order to give an idea of our model's performance, we use the tool provided by Microsoft Word to do the conversion for the same set of test data. The accuracy of the conversion result is 87.86%. This illustrates that our proposed model is comparable to systems based on other conversion methodologies.

7 Conclusion

Most of the code translation or conversion tools developed to handle the Chinese conversion problem are simply based on mapping table of code or lexicon. More sophisticated conversion systems adapt the deep analysis approach, where various Chinese analysis systems are used as the preprocessing steps, such as the segmentation of word, labeling of syntax categories (part of speech), even the syntactic and semantic analyzers of sentence. However, robust system is not always available, especially for different analytic systems. Moreover, the management of the ambiguities in these language analyzers has to tackle the combination of overall ambiguities. In this paper, a statistic approach based on Maximum Entropy model is proposed to construct a Chinese translation system for the conversion of characters between traditional and simplified forms. Similar to other Natural Language Processing tasks, the Chinese to Chinese conversion processing is transformed into a labeling problem. Experiments were performed to evaluate the performance of the constructed model in terms of recall rate and the conversion accuracy. The empirical results show that the proposed model is comparable to the conversion system provided by the MS Word.

References

1. Wang, N.: The Principle of Building Parallel Term Corpus for Simplified Chinese Characters Conversion. In: Proceedings of The 4th Chinese Digitization Forum (CDF) Macao, SAR, China (2007)
2. Halpern, J., Kerman, J.: The Pitfalls and Complexities of Chinese to Chinese Conversion. In: Proceedings of Fourteenth International Unicode Conference. Cambridge, Massachusetts (1999)
3. Leong, K.S., Wong, F., Li, Y.P., Dong, M.C.: Integration of Named Entity Information for Chinese Word Segmentation Based on Maximum Entropy. Advanced Intelligent Computing Theories and Applications. With Aspects of Theoretical and Methodological Issues, vol. 5226, pp. 962--969: Springer Berlin / Heidelberg, Shanghai, China, (2008)
4. Xin, C.S., Sun, Y.F.: Simplified-Unsimplified Chinese Conversion and Word Segmentaion. Mini-Micro System, vol. 21, no. 9, pp. 982--985 (2000)
5. Xin, C.S., Sun, Y.F.: Design and Implementation of a Simplified-Unsimplified Chinese Character Conversion System. Journal of Software, vol. 11, no. 11, pp. 1534--1540 (2000)
6. Wu, A.: Chinese Word Segmentation in MSR-NLP. In: Proceedings of The second SIGHAN workshop on Chinese language processing, pp. 172--175. Sapporo, Japan (2003)
7. Berger, A.L., Pietra, V.J.D., Pietra, S.A.D.: A Maximum Entropy Approach to Natural Language Processing. Computational Linguistics, vol. 22, no. 1, pp. 39--71 (1996)
8. Ratnaparkhi, A.: A maximum entropy model for part-of-speech tagging. In: Proceedings of Empirical Methods in Natural Language Processing (EMNLP), pp. 133--142. Association for Computational Linguistics, New Brunswick, New Jersey (1996)
9. Suárez, A., Palomar, M.: A Maximum Entropy-based Word Sense Disambiguation system. In: Proceedings of the 19th International Conference on Computational Linguistics, pp. 960--966. Taipei, Taiwan (2002)
10. "Hong Kong, Mainland China & Taiwan: Chinese Character Frequency." Chinese University of Hong Kong, http://humanum.arts.cuhk.edu.hk/Lexis/chifreq/.

Tracking Out-of-date Newspaper Articles

Frederik Cailliau[1,2], Aude Giraudel[3] and Béatrice Arnulphy[4]

[1] Sinequa Labs, 12 Rue d'Athènes, F-75009 Paris
[2] LIPN-Univ. de Paris-Nord, 99 av. Jean-Baptiste Clément, F-93430 Villetaneuse
[3] Syllabs, 15 rue Jean-Baptiste Berlier, F-75013 Paris
[4] LIMSI-CNRS, BP 133, F-91403 Orsay*
cailliau@sinequa.com giraudel@syllabs.com beatrice.arnulphy@limsi.fr

Abstract. Local newspapers rely on their local correspondents to bring you the freshest news of your home town. Some of the articles written by these correspondents are not published immediately, but are put aside to be placed in a later edition. One of the challenges the editor in chief is confronted with, is to publish only up-to-date information about a local event. In this paper we present a system that tracks out-of-date newspaper articles to prevent their publishing. It firstly dates the event the article is talking about. The date detection grammar is written in a single but complex finite state automaton based on linguistic pattern matching. Secondly, it computes an absolute date for each relative date. A freshness score can then be deduced from the time difference between the extracted and the publication date. The system has been tested on several French local newspaper corpora. Baseline for the temporal extraction is our standard date extraction that was developed for general purposes.

Keywords: extraction of temporal information, named entities, events, information retrieval, newspaper articles

1 Introduction

Local daily newspapers rely on their network of correspondents to cover local events. Once revised by a journalist, these texts are able to be published as newspaper articles. Everyday the editor in chief validates or assembles the pages that will be published in the next edition. One of the challenges he is confronted with, is to validate only those articles covering hot news or coming events and to replace out-of-date articles by more recent ones. Unfortunately there is no easy or quick way to perform this task. To judge whether the article talks about a recent or a coming event, the editor has to read most of the article, since the day of writing is not a reliable indicator, when given. Even if most of the articles may be short, this activity is too time-expensive to be executed in the short delay before going to press. The editor's

* Some of the work in this article was done in collaboration with Béatrice Arnulphy during her internship at Sinequa in the summer of 2006 as a graduate student of the French university *Université de Provence.*

© A. Gelbukh (Ed.)
Advances in Computational Linguistics.
Research in Computing Science 41, 2009, pp. 277-288

Received 09/11/08
Accepted 15/12/08
Final version 05/02/09

efficiency relies on his or hers capability of rapidly dating the event an article is talking about.

Sinequa[†] is a French software editor of search technology, and several of its press-related clients expressed the need to facilitate this process. Therefore, we have enriched its search technology with a tool that performs automatic date and period detection and calculates the difference with the publication date as to give a *freshness* score to each article. That score can then be indexed as meta-data.

The paper is organized as follows. After a section on related work, we present a typology of temporal markers that appear in newspaper articles and their corresponding patterns that can be used to date the article's contents. The technology used to tackle the date extraction problem is detailed in section 4. We then dedicate a section on the conversion of the relative dates and the calculation of the difference with the publication date. The final section presents an evaluation of our tool on several local newspaper corpora.

2 Related Work and Working Hypothesis

Well performing Named Entities Recognition (NER) systems are around for quite some time. Back in 1997, the best system at the English NER task in the MUC-7 evaluation showed recognition results close to those of the human annotators: the best system performed a 93.39 score where the "worst" human annotator did 96.95 [1]. NER remains a domain of high interest in the community, as shows e.g. the existence of the second HAREM contest for Portuguese [2] and the 2007 and 2008 ACE (Automatic Content Extraction) evaluations organized by the NIST [3]. Part of the sustaining interest is due to the desire of building robust NER systems for more than one language. The systems that participated at ACE 2007 EDR (Entity Detection and Recognition) task could run their system on evaluation corpora of broadcast news, newswires and weblogs in English, Chinese and Arabic. Three other corpora were available for English: broadcast conversations, telephone and Usenet. The results show that an overall progress is still to be made.

Learning-based probabilistic systems are expected to perform badly when confronted to an unknown text type. But rule-based systems, as report [4], show the same behavior when confronted to other text types: a recognition rate of 90% on written documents of newspaper genre drops to 50% when executed on more informal text. The authors' conclusion is that if we want to get near to human-level scores, rule-based systems have to take into account specific characteristics of the corpus. Our system being rule-based, its results on texts of the same genre are quite interesting in this perspective. [5] shows that open system architecture is a way to go to tackle the problem of the different text types.

ACE 2007 presented a task on the detection of temporal expressions. The standard used for the annotation is TIMEX2 [6]. Currently, our time extractions comply with an internal standard in XML, but could be converted into the TIMEX2 format. Aside some exceptions, they cover a subset of the TIMEX2 expressions. We did e.g. not

[†] See http://www.sinequa.com/ for more information.

include any proper nouns (like *New Year's Eve*, *All Saint's Day*) although we probably should have, nor hour expressions (*8:00*) as lexical triggers. Some adjectives (*biannual*, *daytime*) and adverbs (*currently*, *monthly*) of the standard are clearly not taken into account, while others are (*next*), but only in co-occurrence with other triggers.

Temporal processing of news has been the subject of [7] who detects temporal expressions in print and broadcast news for event ordering. They use a basic set of handcrafted rules refined by machine-learning results. [8] extracts temporal information of any document in order to obtain the temporal coverage of its topics, called event-time.

Finally, TimeML [9] provides an annotation scheme for identified events in a text document to be oriented on a timeline. It is the recognized standard for inclusion of all temporal references in order to build a general model of text semantics. Parent et al. (2008) have reported work done on French, but this kind of complete temporal analysis is too rich for our purpose. It is certainly very important for language modeling and has a direct application in Question Answering [10].

Our approach is engineering-driven, rule-based and corpus-oriented. The framework is clearly determined by the application and the extraction rules are motivated by the corpus study. These restrictions led us to the following working hypothesis.

1. A newspaper article is self-sufficient: it usually talks about one or more events that will be dated within the same article.

2. An article can contain more than one date. The one closest to publication date is probably closely related to the main event and therefore the only one of interest for the calculation of the article's freshness score.

3. An article mainly focuses on one event. All dates within one article are somehow related to this event. If this is not the case, the article is out of our scope.

4. Simple regular expression date detection is of no use as we need to situate all dates, eg. *Thursday*, in respect to the writer's temporal perspective: for the freshness score, we need to know whether it happened last Thursday or will happen next Thursday. A complete temporal analysis would in the mean time be too ambitious for our purpose.

3 Typology of the Extracted Temporal References

Of all temporal references discourse is made of, we will only detect those that give us a precise time indication about the article's event. We will call them *direct* in opposition to *indirect* temporal references. Indirect references give a circumstantial description and their conversion into an absolute temporal expression would ask for a deep syntactic and semantic understanding combined with some ontology-based calculation: e.g. *during the last presidential elections*. As indicated before, this is far too ambitious for the aimed application. Instead, we extract all direct references and use verbal tense as a temporal marker to situate the reference future or past to the writer's perspective.

The extracted temporal references are illustrated in Table 1.

Table 1. Typology of extracted temporal references

Type	Subtype	Example	Verb tense	Translation
Explicit relative references	TODAY	aujourd'hui; ce matin	-	today, this morning
	YESTERDAY	hier	-	Yesterday
	DAY_BEFORE_YESTERDAY	avant-hier	-	the day before yesterday
	TOMORROW	demain	-	Tomorrow
	DAY_AFTER_TOMORROW	après-demain	-	the day after tomorrow
Isolated days	DAY	dimanche	-	Sunday
	DAY_BEFORE	dimanche dernier	-	last Sunday
		dimanche	past	Sunday
	DAY_AFTER	dimanche prochain	-	next Sunday
		dimanche	present, future	Sunday
Weekends	WEEKEND_BEFORE	le week-end dernier	-	last weekend
		ce week-end	past	this weekend
	WEEKEND_AFTER	le week-end prochain	-	next weekend
		ce week-end	present, future	this weekend
Explicit date references	COMPLETE_DATE	21 juin 2007	-	21 June 2007
	DATE_BEFORE	lundi 3 mars; jusqu'au 12 janvier; depuis le 26 juin	past	Monday 3 March; until the 1st of April; since the 26th of June
	DATE_AFTER	lundi 3 mars; jusqu'au 12 janvier; depuis le 26 juin	present, future	Monday 3 March; until the 1st of April; since the 26th of June
General references	GEN_DATE	lundi 3 mars; 2004	-	Monday 3 March; 2004

The alphanumeric temporal references in table 1 are recognized by the patterns described in table 2. Some special restrictions concern the extraction of the months: they are solely extracted when preceded by the French preposition *en* (in), itself not preceded by a past participle.

Table 2. Alphanumeric temporal reference patterns

Day of the week	Day of the month	Month	Year	Example
x	x	x	x	lundi 3 mars 1981
	x	x	x	3 mars 1981
x	x	x		lundi 3 mars
	x	x		3 mars
		x	x	mars 1981
x				lundi
			x	1981

The following two temporal reference patterns are deliberately not extracted:

1. day of the week + day of the month (with no indication of the month) : e.g. *mardi 3* (= on Tuesday, the 3rd);

2. *le* + day of the week, which is used in French for events recurring every week: e.g. *le lundi* (= every Monday);

Special restrictions were introduced on the recognition of *aujourd'hui* (= today) and *hier* (= yesterday) because they can sometimes occur in idiomatic expressions like in *d'hier et d'aujourd'hui*, which can signify "of yesterday and today", as well as "of all times". Contextual recognition of these two references requires them to co-occur with a small and incomplete list of event verbs (eg. *débuter* = to start; *organiser* = to organize) and expressions (eg. *avoir lieu = to take place*).

Durations are only partially taken into account: only the last reference is extracted. Eg. *du vendredi 6 au 8 juin* (from the 6th to the 8th Juin) + past tense : *8 juin* is extracted as DATE_BEFORE ; *du 6 au 8 juin* + present/future tense : *8 juin* is extracted as DATE_AFTER.

Sequences of month days are treated in the same way as the durations, eg. *le 6, 7 et 8 février* (the 6th, 7th and 8th February): *8 février* is either extracted as preceding or following the publication date.

Depending on the verb tense, sequences of separate weekdays are not treated in the same way. If the tense is past, the last one is extracted, if it is present or future, the first one is. This way, only the weekday closest to the publication day is extracted. E.g. *La conférence aura lieu lundi, mardi, mercredi* ... (The conference takes place on Monday, Thursday, Wednesday): *lundi* is extracted.

4 Extraction Technology

We use the standalone version of the information extraction suite of Sinequa's search technology. It provides a rule- and lexicon-based morpho-syntactic analysis as well as disambiguation based on a general language model. Entity extraction is performed on the tagger's output with an in-house proprietary transducer technology. Patterns can be defined using word forms, lemmas and morpho-syntactic categories. These can be negated and combined. Word forms can be expressed by regular expressions and categories are dictionary-based (e.g. *adj* for the grammatical category adjective, *s* for singular) or computed (e.g. *has_vowel* for words with at least one vowel). For performance reasons, verb tense is provided by morpho-syntactic analysis rather than syntactic parsing.

Subtypes can be defined for the different paths. They are heavily exploited to provide the relative to absolute date conversion module with the necessary information. They also indicate whether the extracted date is situated before or after the writing date. Fig. 1 presents a screenshot of the transducer containing all of the extraction patterns. It contains 388 states and 668 arcs. The initial state is at the center and eleven final states are dispersed on the border of the image. The number of match-paths totals the impressive number of 6174.

The transducer used for the baseline extraction in the evaluation counts 67 nodes and 138 arcs. It represents 736 match-paths.

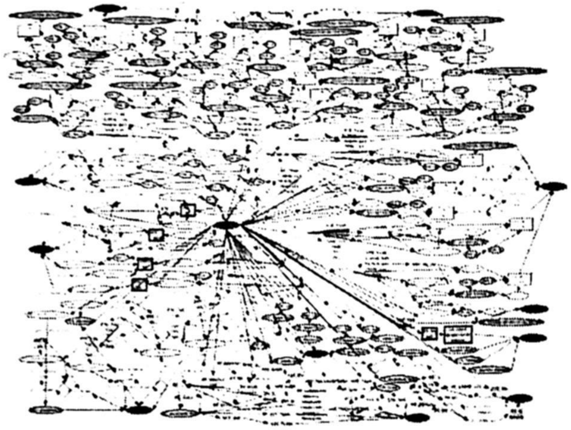


Fig. 1. Transducer for temporal reference extraction

5 Converting Extracted Dates into Absolute Temporal Expressions

The various temporal references extracted by the automaton can be absolute or relative. The global objective of the work is to give an absolute date to each newspaper article. Absolute dates are dates independent from any relative marker and can be listed into a calendar. Typical pattern for absolute dates is the following sequence: day of the month + month + year. All references should thus be transformed to absolute dates. The techniques used to compute absolute dates differ between types of relative references. All are presented in the next sections.

5.1 Explicit Relative Dates

Some relative dates are extracted with sufficient precision so as to easily deduce an absolute date. Those references belong to the extracted subtypes TODAY, YESTERDAY, DAY_BEFORE_YESTERDAY, TOMORROW and DAY_AFTER_TOMORROW. The absolute date is just the result of the addition or subtraction of 0 to 2 days to the publication date.

5.2 Isolated Days

The case of isolated days like *le mercredi dernier* (= last Wednesday) or *il arrivera lundi* (= he will arrive on Monday) that respectively refer to the subtypes DAY_BEFORE and DAY_AFTER implies a computation of the number of days between extracted reference and publication day.

The articles' metadata provide the publication date as an absolute date. It is formatted as following: YYYY-MM-DD, where Y=year, M=month and D=day of the month. To be able to compare the extracted day of the week with the given day of the month, we must transform the least into a day of the week. The technique used to achieve this transformation is the use of the famous Zeller formula. This formula computes the day of the week corresponding to a given date:

$$\text{Day} = (j+[2.6m-0.2] + a + [a/4] + [c/4] - 2c - (1+b)[m/11] \pmod 7) \tag{1}$$

- j is the day of the month
- m is the month number (march being the first month)
- c is the hundreds of the year
- a is the year in the century
- $b=1$ for bissextile years

The return value is a number from 0 to 6, 0 corresponding to Sunday.

The extracted day is then mapped to the same numbering system and the absolute day is given after calculation of the distance between the two days. The computation is different whether the extracted day is situated before or after the publication day.

$$\text{DAY\_AFTER} : d = (\text{extracted\_day} - \text{publication\_day}) \pmod 7 \tag{2}$$

$$\text{DAY\_BEFORE} : d = -((\text{publication\_day} - \text{extracted\_day}) \pmod 7))$$

5.3 Weekends

Dates referring to weekends (subtypes WEEKEND_BEFORE and WEEKEND_AFTER) are problematic as they correspond to 2 days. Given that we want to give a unique absolute date to each extracted reference, we have to decide between the 2 weekend days. Our decision was to take the day nearest to the publication day in order to get the minimum distance between the two dates. As a consequence, the extracted references "last weekend" and "next weekend" will respectively refer to "last Sunday" and "next Saturday". The exact distance to the publication date is computed using again the Zeller formula on publication date and a simple difference between the two normalized dates.

5.4 Explicit and Absolute Dates

Some temporal references are already given in the article in their absolute date form: (day of the week +) day + month + year, e.g. "January the 5th 2008". These do not require any specific transformation.

Some others are formed without an explicit mention of the year (subtypes DATE_BEFORE and DATE_AFTER). We made the hypothesis that the absence of the year implies a time distance of less than a year to the publication date. As a consequence, the absolute date is computed by considering the difference between days and month for both dates.

6 Giving an Absolute Date to Newspaper Articles

Given that we have computed an absolute date for each extracted temporal reference, the objective is now to give a unique date to the corresponding article. The case where only one date has been extracted is trivial as the extracted date is assigned to the article. The problem is harder when for an article comes several or no dates.

6.1 Choosing between Multiple Dates

When several dates have been extracted from the article, we decided to assign the date nearest to the publication date. We assume here that previous dates are used to put the context and later ones to give a deadline. The nearest date is usually the reference that triggers off the recounted event.

Another problem persists: two different dates can be extracted and have the same time distance to the publication date (a previous one and a later one). To follow the hypothesis previously put forward, we assume that the previous date refers to the context of the event while the later one is likely to represent the event to come. We thus chose to assign to the article the later one.

6.2 Absence of Extracted Dates

The absence of extracted temporal references is a definitive obstacle to article dating. This does not necessarily mean that the article is not dated. Some or more cases could have not been predicted in our study. We only can say that we did not found any explicit reference in order to date the article with precision.

The exact dating is thus excluded but there are some key elements that could provide some information about the article. Verb tenses are precious temporal references. Present and future tenses refer to an up-to-date event. The past tense is trickier. If the first sentence uses the present perfect, the following not present perfect verb will determine if the article should be located in the past or in the future.

There is no perfect solution in the absence of exact dating to decide whether the article is up-to-date or not.

7 Evaluation on Four French Local Newspapers

The evaluation of our approach is done on a large corpus of four French local newspapers. The objective of the evaluation is two-fold: First, we intend to give a quantitative evaluation about the performances of our extraction technology. The highlight is put on the impact on article dating results with a special focus on error analysis. We then propose a task-oriented comparison to our standard date extraction technology.

7.1 Corpus Presentation

The test corpus is made of 4 subcorpora totalizing 71942 articles. Each subcorpus corresponds to a local newspaper from a different French region. The articles are in the XML format. All articles are well structured and the textual content of the news is well delimited.

Table 3. Corpus of 4 French local newspapers

	Number of documents	Total size (in Mo)
Newspaper A	6202	14.2
Newspaper B	12809	21.5
Newspaper C	19601	38.8
Newspaper D	33330	83
Total	**71942**	**157.5**

Only the smallest of the subcorpora, newspaper A, served as development corpus for the establishment of the extraction patterns.

7.2 Extraction Results

Table 4 lists the distribution of subtype extractions in the different newspapers. In spite of some local disparities in the distribution of subtype extractions, our results show certain regularity.

The subtype GEN_DATE is the most represented. As this subtype corresponds to a date not precise enough to give an absolute date to the extracted reference, this means that, on average, one third of extracted references do not allow to date the article.

The other extracted references correspond to precise dates and are shared out evenly between dates referring to the past and dates referring to the future, with a short lead for previous dates.

Table 4. Distribution of subtypes extractions in the corpus (in %)

	News A	News B	News C	News D
GEN_DATE	30.7	31.1	40.8	43.6
COMPLETE_DATE	1.9	2.4	2.6	3.5
TODAY	1.8	2.2	5.3	3.7
YESTERDAY	10.4	4.7	9.9	13.3
DAY_BEFORE_YESTERDAY	0.1	0.1	0.1	0.1
TOMORROW	4.7	1.4	5.8	3
DAY_AFTER_TOMORROW	0.1	0.1	0.1	0.1
DAY	0.4	1.5	1.2	0.5
DAY_BEFORE	25.2	21.8	8.4	10
DAY_AFTER	12.8	8.8	5.8	5.4
WEEKEND_BEFORE	0.9	1.3	0.5	0.6
WEEKEND_AFTER	0.5	0.8	0.6	0.5
DATE_BEFORE	5	10.9	8.2	7.4
DATE_AFTER	5.7	13	11.1	8.2

7.3 Dating of the Articles

If we exclude newspaper A that was used to develop the transducer, we observe that 30% to 40% of the articles are undated (see Table 5). These results are not surprising as approximately one third of the extracted dates do not refer to absolute dates.

Nevertheless, results show that undated articles can be divided into two categories: articles without extracted dates and articles with imprecise extracted dates. The case of the absence of dates has previously been developed (see 4.2), so we will say no more about it.

Table 5. Distribution of undated articles

	Articles with no dates	Articles with no precise dates	Total of undated articles
News A	2.9 %	2.4 %	5.3 %
News B	23.4 %	7.6 %	31 %
News C	22.1 %	20.7 %	42.8 %
News D	18.4 %	18.3 %	36.7 %

Imprecise dates. When we took a closer look into the results, we noticed that there are roughly two major patterns where the system was unable to compute an absolute date. The first corresponds to a sequence of four figures and is meant to match any year. The second matches months introduced by "in". These references are not intended to give an absolute date but are key elements to temporally situate the date in the present or in the past

7.4 Analysis of Extraction Errors

Some extraction patterns have a tendency to be error-prone. The first unreliable pattern tries to match all occurrences of years i.e. every sequence of four figures (e.g. "1324"). The problem detected for this pattern is that we have not set any time window and almost all sequences of 4 figures are extracted. Tests on the corpus show that on average 50% of these matches do not refer to a plausible year (before year 1950 and after year 2015).

The use of verbal tenses is another source of errors that is much more difficult to comprehend. The extraction technique does not use syntactic analysis. As a consequence, the verb chosen to temporally locate the event is not always the right one. The extraction of subtypes DATE_BEFORE and DATE_AFTER may thus be erroneous. Analysis on the corpus reveals that about 15% of those extractions give the wrong position to the publication date.

Even if those problems have to be corrected, we have noticed no consequence on article dating. In fact, extractions of years do not refer to absolute dates and are thus not used for article dating. The errors on verbal tenses may be more harmful but our distance calculation makes up for it and always gives the exact difference between extracted date and publication date when complete dates are extracted, which is the case here.

7.5 Comparison with our Standard Date Extraction

A comparison of our approach to date articles and the use of standard date extraction has been achieved to highlight the differences between the two techniques and explain the necessity to develop our article dating methodology.

The number of extractions resulting from both approaches is shown in Table 6. The total amount of extractions is equivalent but the nature of the extracted references is significantly different as shown by the number of common extractions.

Table 6. Comparison of extractions between standard date and article dating extractions

	Number of extractions with standard date automaton	Number of extractions with article dating automaton	Number of common extractions
News A	28811	25448	16825
News B	44847	27378	18653
News C	58043	49210	26785
News D	108994	107015	59050

Extraction differences. The differences between the two extraction techniques are of two natures. On the one hand, some subtypes are not recognized by the "standard" approach, e.g. DAY and WEEKEND_BEFORE. The non extracted subtypes generally correspond to complex structures. On the other hand, the "standard dates" not extracted by the article dating transducer are mainly references that cannot be converted into absolute dates and have therefore been deliberately excluded from recognition.

Dating usability. The major difference between the two approaches lies in the capacity to locate the extracted references. The article dating approach can not only extract temporal references but also situate them in respect to the publication date. This information is not provided by the standard date extraction technique which makes it unusable for our purpose.

8 Conclusion and Perspectives

In this paper, we have studied the impact of temporal reference extraction on an article dating task. The results show that, when present, any extracted reference can assign an absolute date to the article corresponding to the moment when the related event takes place. In spite of the good precision of the technique, the amount of undated articles is still a problem. An interesting lead, not available in our corpora, would have been to select articles only written by local correspondents to get better results.

For genericity concerns, one would expect the use of handcrafted rules to give somehow more corpus independent results than with a learning-based method. Our results show that even handcrafted rules are corpus-dependant, since the difference between the test corpus and the other corpora was too significant to be a coincidence.

It would be most interesting to compare the results of both methods on several corpora.

To enhance the precision of our approach, the use of syntactic analysis instead of the morpho-syntactic one has to be tested. It would be of great interested to select the exact dates referring to the related event. Finally, not all references should be considered at the same level. An in-depth study of the type of speech containing a temporal reference is needed to get a better analysis of what reference to extract.

References

1. Proceedings of the Message Understanding Conference 7, MUC-7, http://www-nlpir.nist.gov/related_projects/muc/proceedings/ne_english_score_report.html
2. Mota, C., Santos, D.: Desafios na avaliação conjunta do reconhecimento de entidades mencionadas: Actas do Encontro do Segundo HAREM (Aveiro, 7 de Setembro de 2008). Linguateca (2008).
3. ACE, http://www.nist.gov/speech/tests/ace/
4. Poibeau, T., Kosseim, L.: Proper Name Extraction from Non-Journalistic Texts. In W. Daelemans, K. Sima'an, J. Veenstra and J. Zavrel (eds.) Computational Linguistics in the Netherlands. Selected Papers from the Eleventh CLIN Meeting, p. 144-157, Amsterdam/New York (2001)
5. Maynard, D., Tablan, V., Ursu, C., Cunningham, H., Wilks, Y.: Named entity recognition from diverse text types. In Recent Advances in Natural Language Processing 2001 Conference. Tzigov Chark, Bulgaria (2001)
6. Ferro, L., Gerber, L., Mani, I., Sundheim, B. and Wilson G. *TIDES 2005 Standard for the Annotation of Temporal Expressions.* April 2005, Updated September 2005 (2005)
7. Mani, I., Wilson, G.: Robust temporal processing of news. In Proceedings of the 38th Annual Meeting on Association for Computational Linguistics, pp. 69-76. Morristown, NJ (2000)
8. Llidó, D., Llavori, R. B., and Cabo, M. J.: Extracting Temporal References to Assign Document Event-Time Periods. In *Proceedings of the 12th international Conference on Database and Expert Systems Applications* (September 03 - 05, 2001). H. C. Mayr, J. Lazanský, G. Quirchmayr, and P. Vogel, Eds. Lecture Notes In Computer Science, vol. 2113, pp. 62-71. Springer-Verlag, London (2001)
9. Pustejovsky J., Ingria R., Sauri R., Castano J., Littman J., Gaizauskas R., Setzer A., Katz G. & Mani I: The specification language TimeML. In I. Mani, J. Pustejovsky & R. Gaizauskas, Eds., The Language of Time: A Reader. Oxford University Press (2005).
10. Pustejovsky, J., Knippen, R., Littman, J. Saurí, R. Temporal and Event Information in Natural Language Text. Computers and the Humanities, Volume 39, Numbers 2-3, May 2005, pp. 123-164(42). Springer (2007)

PtiClic: A Game for Vocabulary Assessment combining JeuxDeMots and LSA

Mathieu Lafourcade[1] and Virginie Zampa[2]

[1] LIRMM-TAL, Univ. Montpellier 2 France
mathieu.lafourcade@lirmm.fr
[2] LIDILEM-DIP Univ. Grenoble 3 France
virginie.zampa@u-grenoble3.fr

Abstract. One interesting path for designing software for vocabulary acquisition and assessment could be games. In this article we present PtiClic, a lexical game based on the principles behind JeuxDeMots and combined with LSA. Such a game can foster the interest of young people in developing lexical skills in either a tutored or an open environment.

Keywords: Lexical acquisition, lexical network, serious games, LSA, JeuxDeMots

1 Introduction

Developing software for vocabulary acquisition and/or assessment in general, and for young people in particular, is a risky business. There are several difficulties. First, we have to be able to design an activity that can foster an interest in learning which is not easy in the case of children. Then, the underlying dictionaries or lexical databases can prove tremendously hard to develop, especially if we want to go beyond just a list of words with parts-of-speech and venture into the realm of lexical functions and the relations intertwining terms.

Automated acquisition of lexical or functional relations between terms is necessary in a large number of tasks in Natural Language Processing (NLP) outscoping largely Technology-Enhanced Learning of language. These relations that we find generally in thesauruses or ontologies can of course be revealed in a manual way; for example, one of the oldest thesauruses is Roget's, its current version being (Kipfer, 2001), or the most famous lexical network is Wordnet (Miller, 1990). Such relations can be also determined computationally from corpora of texts, for example (Robertson and Spark Jones, 1976), (Lapata and Keller, 2005), or (Landauer et al 1998) in which statistical studies on the distributions of words are made. Moreover, many applications of NLP require information of various natures, like synonymy or antonymy, but also relations of hyperonymy / hyponymy, holonymy / meronymy etc. The building of such relations, when done manually by experts, requires resources

© A. Gelbukh (Ed.)
Advances in Computational Linguistics.
Research in Computing Science 41, 2009, pp. 289-298

Received 10/11/08
Accepted 15/12/08
Final version 06/02/09

which can be prohibitive, while their automatic extraction from a corpus can be biased by the chosen texts.

The method developed here relies on a contributory system, where the users feed the relation database through a game (JeuxDeMots). Furthermore, contrary to conventional methods which aim generally at static lexical information, the prototype introduced here is able to acquire evolving lexical information over time. From this game, we have designed a sequel, named PtiClic, which is a simpler version of JeuxDeMots aimed at vocabulary acquisition and assessment.

In this article, we first present the principles of JeuxDeMots[1] a game for building a lexical database containing various lexical functions. Secondly, we explain briefly how we produced a training corpus for LSA from the lexical network. Finally, the proposal of the PtiClic game is described with some insight into the rationale behind the design.

2 How to Build a Lexical Network through a Game

2.1 Principle of JeuxDeMot

To ensure a system leading to (quality and consistency in/the quality and consistency of) the lexical database, it has been decided that the relations anonymously given by a player should be validated by other players, also anonymously. In practical terms, this means a relation is considered valid if it is given by at least one pair of players. This process of validation is similar to the one used by (von Ahn and Dabbish, 2004) for the indexation of images or more recently by (Lieberman and al., 2007) to collect common sense knowledge. As far as we know, this has never been done in the field of lexical networks.

A game takes place between two players, in an asynchronous way, based on the concordance of their propositions. When a first player (A) begins a game, an instruction concerning a type of competence (synonyms, antonyms, domains …) is displayed, as well as a term T randomly picked from a database of terms. This player A has then a limited timeframe in which to answer by giving propositions which, in his view, correspond to the instruction applied to the term T. The number of propositions which can be made is limited so that players do not just type anything as fast as possible, as we want the players to take time to think. The same term, along with the same instruction, is later proposed to another player (B); the process is then identical. To increase the playful aspect, for any common answer between the two players, they each receive a given number of points. The calculation of this number of points (explained in section 2.2) is crafted to induce both precision and recall in the feeding of the database.

[1] JeuxDeMots is available at http://jeuxdemots.org. An English version has recently been added, as well as a Thai version and a Japanese version (both in development), at http://www.lirmm.fr/jeuxdemots/world-of-jeuxdemots.php

For the target term T, we do not record answers given only by one of the two players but record the common answers given by player pairs. This allows the construction of a lexical network connecting the terms by typed and balanced relations, validated by pairs of players. These relations are labeled by the instruction given to players and they are weighted according to the number of pairs of players who proposed them. The structure of the lexical network relies on the notions of nodes and relations between nodes, as reminded by (Polguère, 2006). Every node of the network is constituted by a lexical item (term or expression) grouping together all its lexical items and the relations between nodes refer to lexical functions, as presented by (Mel'cuk and al., 1995). Nodes are constituted by an initial set of terms, but if both players in the same game suggest a term hitherto unknown (i.e not in the database), it is then added to the lexical database. Figure 1 presents the relations acquired for the French term *aile* (*wing*).

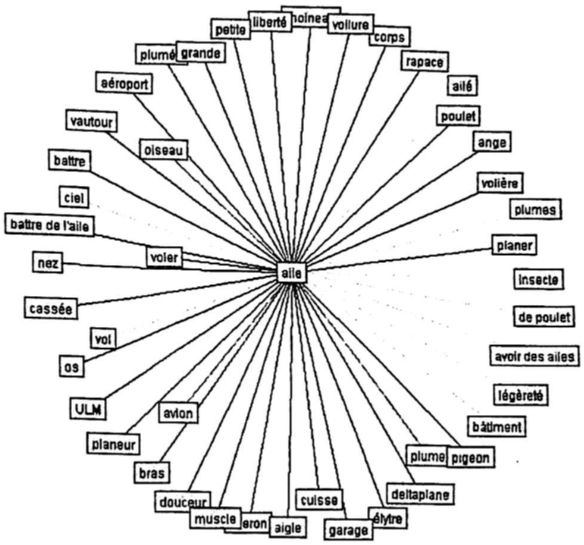


Fig. 1 Partial example of the lexical network for *aile*.

aile61 relations ==>
aile --r_assoc:370-> oiseau
aile --r_assoc:370-> voler
aile --r_assoc:340-> avion
aile --r_assoc:260-> plume
aile --r_assoc:140-> poulet
aile --r_assoc:130-> vol
aile --r_loc:130-> oiseau
aile --r_assoc:110-> cuisse
aile --r_assoc:100-> ange

aile --r_assoc:90-> planer
aile --r_loc:90-> avion
aile --r_holo:80-> avion
aile --r_holo:80-> oiseau
aile --r_assoc:70-> deltaplane
aile --r_assoc:60-> pigeon
aile --r_syn:60-> aileron
aile --r_syn:60-> bras
aile --r_has_part:60-> os
aile --r_has_part:60-> plume
aile --r_holo:60-> aigle

```
aile --r_loc:60-> ange
aile --r_loc:60-> bâtiment
aile --r_loc:60-> volière
aile --r_assoc:50-> ailé
aile --r_assoc:50-> battre
aile --r_assoc:50-> bras
aile --r_assoc:50-> bâtiment
aile --r_assoc:50-> insecte
aile --r_assoc:50-> moineau
aile --r_assoc:50-> planeur
aile --r_assoc:50-> plumes
aile --r_assoc:50-> voiture
aile --r_syn:50-> voilure
aile --r_syn:50-> élytre
aile --r_hypo:50-> plume
aile --r_has_part:50-> muscle
aile --r_holo:50-> ULM
aile --r_holo:50-> pigeon
aile --r_holo:50-> voiture
aile --r_loc:50-> aigle
aile --r_loc:50-> aéroport
aile --r_loc:50-> ciel
aile --r_loc:50-> garage
aile --r_loc:50-> pigeon
aile --r_loc:50-> poulet
aile --r_loc:50-> vautour
aile --r_loc:50-> voiture
aile --r_carac:50-> cassée
aile --r_carac:50-> grande
aile --r_carac:50-> petite

aile        31 relations <==

mouche --r_assoc:260-> aile
plume --r_assoc:240-> aile
oiseau --r_has_part:230-> aile
oiseau --r_assoc:200-> aile
insecte --r_has_part:150-> aile
poulet --r_assoc:150-> aile
rapace --r_has_part:140-> aile
volaille --r_has_part:140-> aile
papillon --r_assoc:130-> aile
avion --r_has_part:120-> aile
cuisse --r_assoc:100-> aile
nez --r_has_part:90-> aile
coq --r_has_part:80-> aile
avion --r_assoc:70-> aile
voler --r_instr:70-> aile
Ailette --r_assoc:60-> aile
faucon --r_has_part:60-> aile
frégate/oiseau --r_assoc:60-> aile
plume --r_holo:60-> aile
Icare --r_assoc:50-> aile
battement --r_assoc:50-> aile
carrosserie --r_has_part:50-> aile
deltaplane --r_has_part:50-> aile
fée --r_assoc:50-> aile
huîtrier-pie --r_assoc:50-> aile
poule --r_has_part:50-> aile
poule --r_assoc:50-> aile
poulet --r_hypo:50-> aile
toucan --r_assoc:50-> aile
voilure --r_syn:50-> aile
voler --r_assoc:50-> aile
```

The JeuxDeMots software was mainly developed in PHP / MySQL; some secondary programs were written in the Java and C ++ languages. The user interface, the computation of scores, but also the notions of levels and points of honor, winning terms, trials between players etc., as well as the display of players' rankings, were implemented to make the game more attractive. The purpose is to incite players to return regularly to the site, and thus to increase the number of acquired relations accordingly: it is the major interest of this game compared to some other software programs which would merely ask users to provide relations making them certainly more aware of their role as experts, but probably leading them to spend less time on the game.

2.2 How is the Game Played?

Every time a player connects to the site and starts a game, an instruction is displayed for a few seconds (for example: "give ideas associated with ..."), before the term to which this instruction applies appears on the screen. This term is randomly picked from a base of about 150,000 terms. The player then has one minute to give their answers. If the player is (B), the result of the game, the suggestions made by player (A) and the number of points won are immediately displayed. If he is a player (A), the equivalent information will be sent to him by e-mail after (B) has played (this last point also serving as a reminder to return to the site and continue playing the game). The games proposed to the player are either "starter games" where he is a player (A), or "games to be completed" in which he is a player (B). Thus, there are a certain number of games to be finished.

For a given term and a given instruction, if a player has no idea, he can "pass", then ending the game prematurely. There are two main reasons for the player not to give any answer: either the term is not a common term (for example: *"gnomon"*, which is an old Greek sundial), or the instruction applied to this term does not have any clear meaning (for example: *"contraires de pigeon ?"* (*"Opposites for pigeon ?"*). The system then records the fact that this term is may not have many lexical relations, in particular with regard to this instruction; consequently, the term coupled with this instruction will come up less often.

Any game created with a player (A) generates two games to be finished. Indeed, if it were not the case, player (B) would just have to pass without making any suggestion which may induce a feeling of frustration for player (A) who might lose interest in the game. This is why more games to be finished than beginning games are offered to players as it is indeed more rewarding to get the immediate result of their propositions.

In order to allow each player to compare themselves with the others, it is possible to display a summary table of the recorded players, with their performances. The member list is ordered according to their points of honor, as well as their best scores obtained in a game.

2.3 What are the Results?

The current version of JeuxDeMots is relatively recent: it was released in July 2007. In approximately twelve months, more than 1200 players have been registered and most of them connect several times a week. More than 120,000 games have been played: they have brought to the foreground more than 180,000 relations, among which 80,000 of the "associated ideas" type. At present, more than 2000 relations are taboo, i.e. they have been validated by a sufficient number of player pairs, and therefore to encourage the production of new links between words, players are informed that these words are "taboo" and will not win any points.). There is a fast emergence of the relations and we also note that statistically the strongest are created first (i.e. they are the most spontaneous ideas the player comes up with). The evolution of the base of terms is inevitably slower: to date, it amounts to approximately 163,000 terms; players have already added more than 8000 new terms to it, mainly related to current events.

3 From a Lexical Network to a LSA Corpus

We use LSA to produce a cloud of words from a target word. From a given word, with LSA we can obtain any number of similar terms using a knn (K nearest neighbors) algorithm. The issue raised here is how to produce a corpus for training LSA from a lexical network? First, we have the following assumptions about using LSA:

- The context window is limited to a paragraph ;
- All words in a paragraph are equal in building the context of each of them.

3.1 From a Lexical Network to a Training Corpus for LSA

The solution we adopted, although imperfect, is quite straightforward. We enumerated all the associations contained in the database of JeuxDeMots and produced for each of them a duplicated number of paragraphs roughly proportional to the weight of the relation divided by 10. This approximation is motivated by the desire not to inflate the corpus too much. We did not, at this stage, take into account the type of relation. For example, for the relations presented above for *aile*, we obtain:

aile	oiseau	(duplicated 37 times)
aile	voler	(duplicated 37 times)
aile	avion	(duplicated 34 times)
...		
aile	vol	(duplicated 13 times)
...		
aile	muscle	(duplicated 5 times)
aile	ULM	(duplicated 5 times)
aile	pigeon	(duplicated 5 times)

In this way, we produce a corpus of 1.3 million paragraphs, each associating two terms. This corpus is given to LSA to compute a similarity matrix between all terms.

3.2 From a List of Similar Words to a Cloud

We intend to produce a cloud of *n* words (in the prototype of PtiClic we display between 20 to 30 terms). To do so, we ask LSA to produce a knn (k nearest neighbors) list from the similarity matrix with *k* equal 5*n. We select randomly those terms among the *k* terms, in such a way as to produce different clouds for different games with the same target word.

3.3 Why use LSA?

At this point, one question may arise: why use LSA rather than directly producing the cloud of words from the lexical network? Using LSA on a corpus extracted from the lexical network has the following advantages:

- Once compiled, obtaining all distance neighbors and not only immediate neighbors is much more computationally efficient than using a graph walking. In this respect, the application of LSA can be viewed as a direct projection of the lexical network to a space;
- Using LSA we are both making symmetric the relations of the network and reducing the noise;
- The transitivity property of LSA allows catching terms that are not immediately related, or co-hyponyms, which is definitively interesting in a vocabulary assessment task.

3.4 Why use the JeuxDeMots Lexical Network?

Why did we not use a normal corpus of texts for training LSA? A first answer is that building a text corpus that is balanced between text genres is difficult. Secondly, although it is outside the precise topic of this paper, it is interesting to assess the transformation of the lexical network through LSA concerning term similarity. Some terms in the graph may be implicitly related, and LSA can "discover" those hidden relations. The PtiClic games can allow them to be validated.

5 Principle of PtiClic

The game PtiClic (http://pticlic.org) is just a derivation and simplification of JeuxDeMots. Here, we will firstly outline how it is played, and then explain the reasoning behind the choices made.

To begin, a first player (we called the Tutor) selects a term (the target) and the system proposes a cloud of words as produced by LSA. The Tutor then, selects some of these terms by clicking on them and, from a set of menus, builds the definition of the task to completed by the other players (known as Learners). The description of the task is basically to select a given number of terms related to the target term by a lexical function. For example, we can have the following tasks:

- Select up to 5 synonyms
- Select up to 3 most closely related terms
- Select up to 4 terms opposites (antonyms)
- Select up to 6 terms which are not part of the <target>
- Select 4 terms that are kinds of/a kind of <target>

The tutor can also produce some comments (as free text) that will be displayed at the end of the game.

When playing, the learners get the target word, the description of the task and the cloud of words. The player then can click on the words of the cloud to select or unselect them. When finished he can validate his/her answers. The result is then displayed, comparing answers proposed by the Tutor and the learners. Points are given to the learners on the basis of 2 for each proper answer and -1 for each wrong ones.

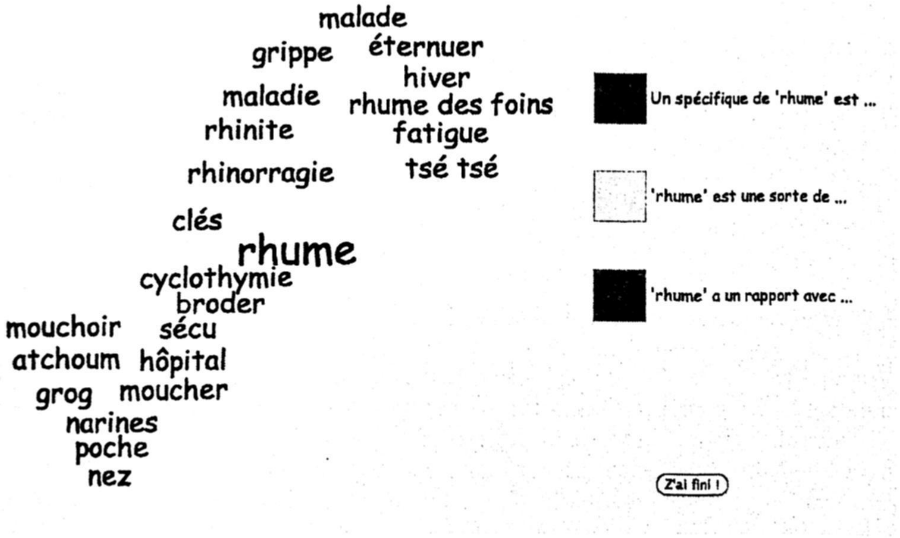


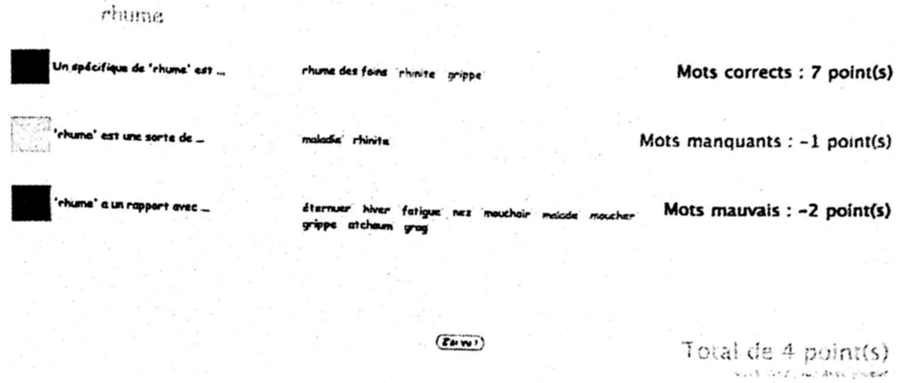
Fig. 2 Example of word cloud around the term "*rhume*" (cold N). Three drop-zones are present and correspond to "specific", "kind-of" and "freely associated ideas".

Although, for a given game built by the Tutor, the cloud of words remains the same, words are displayed in a random order to the Learners. Any given order (for example alphabetical) would produce a bias to the players. By randomly presenting words, we statistically reduce this bias.

PtiClic is a closed world contrary to JeuxDeMots which is an open game, as players can freely propose any terms. This design seems more adapted to new learners of the language, either young people or recent second-language learners. JeuxDeMots, as an open world, presents some obstacles like proper spelling for instance, and forces players to activate otherwise passive vocabulary which may be difficult for new learners, especially if they are indeed trying to acquire this vocabulary.

PtiClic, on the other hand, is simpler in its execution and contributes more to vocabulary acquisition and assessment. The game can be played during a strict time frame or without time limit. In the case of a session aimed more at vocabulary assessment, a time limit for each game is generally set.

A prototype version of PtiClic has been developed in a non-tutored environment. Anyone can play, not specifically people learning vocabulary. In this case, the creator is rewarded with the same amount of points as a player for a given game. This aspect is quite significant, as the creator is induced to select proper lexical items according to the task.



rhume

Un spécifique de 'rhume' est ... rhume des foins rhinite grippe Mots corrects : 7 point(s)

'rhume' est une sorte de ... maladie rhinite Mots manquants : -1 point(s)

'rhume' a un rapport avec ... éternuer hiver fatigue nez mouchoir malade moucher grippe atchoum grog Mots mauvais : -2 point(s)

Total de 4 point(s)

Fig. 3 Example of result of the previous game. The players got 4 points. 7 words of the cloud were dropped on the proper zone, 2 words were not given (in grey) at all, and 4 words were wrong. We can notice that some words could have been dropped in several places. The score is computed in the more favorable way to the player.

At the time of writing of this paper, we do not have more than first impressions made on some users of PtiClic (mainly children form 8 to 14). Overall, the game was found to be very enjoyable and we discovered than it could lead to vocabulary acquisition (although at first the project was essentially aimed at assessment). Indeed, in games without a strict time frame (of around one minute), players went online for dictionaries and encyclopedia to gather information about words in the cloud they didn't know or were not sure about.

The game aspect - in particular the point system - made children try to score as much as possible, forgetting completely the learning aspect while doing so.

6 Conclusion

The JeuxDeMots prototype is an on-line game on the Web the objective of which is the construction of a lexical network. Making a game was justified by the assumption that it would attract a lot of people from various horizons. The emergence of labeled and weighted relations between terms is made through the gaming activity of a large number of users. These users are certainlynot linguists, but we strongly believe that both their number and variety will allow obtaining a lexical network with a satisfactory coverage and precision for general knowledge. Our purpose is not the constitution of an experts' database, but rather the representation of common general knowledge.

From this lexical network, and thanks to LSA, we were able to produce the lexical data that serves as the foundation for another game: PtiClic. This game is a closed world, i.e. players have to make selections among proposals according to a given task, instead of proposing terms by themselves. This game seems to be more appropriate for people still acquiring vocabulary (either young players, or second-language learners). The first experiments seem to confirm the idea that presenting a learning activity through a game with scores, involving emulation between players, is an interesting path of research.

Acknowledgments. We are very grateful to Lucy Garnier for proofreading this paper and asking many not so naïve questions helping to improve its contents.

References

1. vonAhn L. et Dabbish L. (2004) Labelling Images with a Computer Game. In *ACM Conference on Human Factors in Computing Systems (CHI)*, pp. 319-326.
2. Kipfer B.A. (2001). *Roget's International Thesaurus*, sixth edition, Harper Resource (First Edition : 1852)
3. Landauer, T. K., Foltz, P. W., Laham, D. (1998). An introduction to Latent Semantic Analysis. Discourse Processes, 25, 259-284.
4. Lapata M. et Keller F. (2005) Web-based Models for Natural Language Processing. In *ACM Transactions on Speech and Language Processing*, vol.2, n°1, pp. 1-30.
5. Lieberman H., Smith D.A. and Teeters A. (2007) Common Consensus: a web-based game for collecting commonsense goals, *International Conference on Intelligent User Interfaces (IUI'07)*, Hawaii, USA
6. Mel'čuk I.A., Clas A., Polguère A. (1995) *Introduction à la lexicologie explicative et combinatoire*, Editions Duculot AUPELF-UREF
7. Miller G.A., Beckwith R., Fellbaum C., Gross D. and Miller K.J. (1990) Introduction to WordNet: an on-line lexical database. In: *International Journal of Lexicography* 3 (4), pp. 235-244.
8. Polguère A. (2006) Structural properties of Lexical Systems: Monolingual and Multilingual Perspectives. *Proceedings of the Workshop on Multilingual Language Resources and Interoperability* (COLING/ACL 2006), Sydney, pp. 50-59.
9. Robertson S. et Spark Jones K. (1976) Relevance weighting of search terms, *Journal of the American Society for Information Science*, n° 27, pp. 129-146.
10. Salton G. (1968) *Automatic Information Organization and Retrieval*, Mac Graw Hill, NY.
11. Véronis J. (2001) Sense tagging: does it make sense? *Corpus linguistics' 2001 Conference*, Lancaster, U.K.

Author Index

Editorial Board of the Volume

Computational linguistics is an interdisciplinary research area that combines the ideas and methods of linguistics, computer science, and artificial intelligence and has two-fold goal: on the one hand, to study human language by means of modern computational methods, and on the other hand, to develop computer programs capable of human-like activities related to understanding or producing texts or speech in human language, such as English or Chinese.

This volume includes 25 original research papers by authors from 23 different countries on the following areas of theory and applications of computational linguistics:

- Computational lexicography and lexical resources
- Morphology and syntax
- Semantics
- Anaphora and co-reference
- Text classification
- Text summarization
- Speech generation
- Applications

The volume is oriented to researchers and students working in computational linguistics, natural language

ISSN: 1870
www.ipn.mx
www.cic.ipn.m

SEP

INSTITUTO POLITÉCNICO NACIONAL
"La Técnica al Servicio de la Patria"