Andrés Montoyo
Rafael Muñoz
Elisabeth Métais (Eds.)

# Natural Language Processing and Information Systems

LNCS 3513

10th International Conference on Applications of Natural Language to Information Systems, NLDB 2005 Alicante, Spain, June 2005, Proceedings

Springer

Lecture Notes in Computer Science 3513

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

## Editorial Board

David Hutchison
*Lancaster University, UK*

Takeo Kanade
*Carnegie Mellon University, Pittsburgh, PA, USA*

Josef Kittler
*University of Surrey, Guildford, UK*

Jon M. Kleinberg
*Cornell University, Ithaca, NY, USA*

Friedemann Mattern
*ETH Zurich, Switzerland*

John C. Mitchell
*Stanford University, CA, USA*

Moni Naor
*Weizmann Institute of Science, Rehovot, Israel*

Oscar Nierstrasz
*University of Bern, Switzerland*

C. Pandu Rangan
*Indian Institute of Technology, Madras, India*

Bernhard Steffen
*University of Dortmund, Germany*

Madhu Sudan
*Massachusetts Institute of Technology, MA, USA*

Demetri Terzopoulos
*New York University, NY, USA*

Doug Tygar
*University of California, Berkeley, CA, USA*

Moshe Y. Vardi
*Rice University, Houston, TX, USA*

Gerhard Weikum
*Max-Planck Institute of Computer Science, Saarbruecken, Germany*

Andrés Montoyo  Rafael Muñoz
Elisabeth Métais (Eds.)

# Natural Language Processing and Information Systems

10th International Conference on Applications
of Natural Language to Information Systems, NLDB 2005
Alicante, Spain, June 15-17, 2005
Proceedings

Springer

Volume Editors

Andrés Montoyo
Universidad de Alicante
Departamento de Lenguajes y Sistemas Informáticos
Apartado de correos, 99, 03080 Alicante, Spain
E-mail: montoyo@dlsi.ua.es

Rafael Muñoz
Universidad de Alicante
Departamento de Lenguajes y Sistemas Informáticos
Apartado de correos, 99, 03080 Alicante, Spain
E-mail: rafael@dlsi.ua.es

Elisabeth Métais
Cedric Laboratory, CNAM
292 rue Saint-Martin, 75003 Paris, France
E-mail: elsa@cnam.fr

Library of Congress Control Number: Applied for

CR Subject Classification (1998): H.2, H.3, I.2, F.3-4, C.2

ISSN 0302-9743
ISBN-10 3-540-26031-5 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-26031-8 Springer Berlin Heidelberg New York

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

Springer is a part of Springer Science+Business Media

springeronline.com

© Springer-Verlag Berlin Heidelberg 2005
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Olgun Computergrafik
Printed on acid-free paper SPIN: 11428817 06/3142 5 4 3 2 1 0

# Preface

NLDB 2005, the 10th International Conference on Applications of Natural Language to Information Systems, was held on June 15–17, 2005 at the University of Alicante, Spain. Since the first NLDB conference in 1995 the main goal has been to provide a forum to discuss and disseminate research on the integration of natural language resources in information system engineering.

The development and convergence of computing, telecommunications and information systems has already led to a revolution in the way that we work, communicate with each other, buy goods and use services, and even in the way that we entertain and educate ourselves. The revolution continues, and one of its results is that large volumes of information will increasingly be held in a form which is more natural for users than the data presentation formats typical of computer systems of the past. Natural language processing (NLP) is crucial in solving these problems, and language technologies will make an indispensable contribution to the success of information systems. We hope that NLDB 2005 was a modest contribution to this goal.

NLDB 2005 contributed to advancing the goals and the high international standing of these conferences, largely due to its Program Committee, composed of renowned researchers in the field of natural language processing and information system engineering. Papers were reviewed by three reviewers from the Program Committee. This clearly contributed to the significant number of papers submitted (95). Twenty-nine were accepted as regular papers, while 18 were accepted as short papers.

We would like to express here our thanks to all the reviewers for their quick and excellent work. We extend these thanks to our invited speakers, Ruslan Mitkov and Branimir Boguraev, for their valuable contribution, which undoubtedly increased the interest in the conference. We are also indebted to a number of individuals for taking care of specific parts of the conference program, specially to Miguel Angel Varó who built and maintained all Web services for the conference.

March 2005

Andres Montoyo
Rafael Muñoz
Elisabeth Métais

# Organization

## Conference Chairs

Rafael Muñoz (University of Alicante, Spain)
Elisabeth Métais (CEDRIC/CNAM, France)

## Program Chair

Andres Montoyo (University of Alicante, Spain)

## Organization Committee

Patricio Martínez-Barco (University of Alicante, Spain)
Andres Montoyo (University of Alicante, Spain)
Paloma Moreda (University of Alicante, Spain)
Rafael Muñoz (University of Alicante, Spain)
Maximiliano Saiz (University of Alicante, Spain)
Armando Suárez (University of Alicante, Spain)
Elisa Noguera (University of Alicante, Spain)

## Program Committee

Kenji Araki (Hokkaido University, Japan)
Mokrane Bouzeghoub (PRiSM, Université de Versailles, France)
Gary A. Coen (Boeing, USA)
Isabelle Comyn-Wattiau (CEDRIC/CNAM, France)
Günther Fliedl (Universität Klagenfurt, Austria)
Alexander Gelbukh (Instituto Politécnico Nacional, Mexico)
Jon Atle Gulla (Norwegian University of Science and Technology, Norway)
Harmain Harmain (United Arab Emirates University, United Arab Emirates)
Helmut Horacek (Universität des Saarlandes, Germany)
Paul Johannesson (Stockholm University, Sweden)
Zoubida Kedad (PRiSM, Université de Versailles, France)
Nadira Lammari (CEDRIC/CNAM, France)
Jana Lewerenz (sd&m Düsseldorf, Germany)
Robert Luk (Hong Kong Polytechnic University, Hong Kong, China)
Bernardo Magnini (IRST, Italy)
Paul McFetridge (Simon Frazer University, Canada)
Elisabeth Métais (CEDRIC/CNAM, France)
Farid Meziane (Salford University, UK)

Luisa Mich (University of Trento, Italy)
Ruslan Mitkov (University of Wolverhampton, UK)
Ana Maria Moreno (Universidad Politécnica de Madrid, Spain)
Diego Mollá Aliod (Macquarie University, Australia)
Andrés Montoyo (Universidad de Alicante, Spain)
Rafael Muñoz (Universidad de Alicante, Spain)
Jian-Yun Nie (Université de Montréal, Canada)
Manuel Palomar (Universidad de Alicante, Spain)
Odile Piton (Université Paris I Panthéon-Sorbonne, France)
Reind van de Riet (Vrije Universiteit Amsterdam, The Netherlands)
Hae-Chang Rim (Korea University, Korea)
Vijay Sugumaran (Oakland University Rochester, USA)
Veda Storey (Georgia State University, USA)
Bernhard Thalheim (Kiel University, Germany)
Juan Carlos Trujillo (Universidad de Alicante, Spain)
Luis Alfonso Ureña (Universidad de Jaén, Spain)
Sunil Vadera (University of Salford, UK)
Panos Vassiliadis (University of Ioannina, Greece)
Hans Weigand (Tilburg University, The Netherlands)
Werner Winiwarter (University of Vienna, Austria)
Christian Winkler (Universität Klagenfurt, Austria)
Stanislaw Wrycza (University of Gdansk, Poland)

## Additional Reviewers

Andrea, Mulloni
Bergholtz, Maria
Isabelle, Comyn-Wattiau
Nadira, Lammari
Del Jesus Diaz, Maria Jose
Echizen-ya, Hiroshi
Evans, Richard
Ferrández, Antonio
García Cumbreras, Miguel Ángel
Ha, Le An
Haddad, Hatem
Kabilan, Vandana
Kozareva, Zortnisa
Llopis, Fernando
Maria, Bergholtz
Martínez-Barco, Patricio
Martin-Valdivia, M.Teresa
McFetridge, Paul
Montejo Ráez, Arturo
Moreda, Paloma
Navarro, Borja
Pekar, Viktor
Peral, Jesús
Prost, Jean-Philippe
Puscasu, Georgiana
Roger, Sandra
Rzepka, Rafal
Sasaoka, Hisayuki
Selima, Besbes
Storey, Veda
Vadera, Sunil
Vazquez, Sonia
Xue, Xiaohui

# Table of Contents

## Regular Papers

# Extracting Semantic Taxonomies of Nouns from a Korean MRD Using a Small Bootstrapping Thesaurus and a Machine Learning Approach*

SeonHwa Choi and HyukRo Park

Dept. of Computer Science, Chonnam National University,
300 Youngbong-Dong, Puk-Ku Gwangju, 500-757, Korea
csh123@dreamwiz.com, hyukro@chonnam.ac.kr

**Abstract.** Most approaches for extracting hypernyms of a noun from the definition in an MRD rely on the lexico-syntactic patterns compiled by human experts. Not only these methods require high cost for compiling lexico-syntatic patterns but also it is very difficult for human experts to compile a set of lexical-syntactic patterns with a broad-coverage, because in natural languages there are various different expressions which represent the same concept. To alleviate these problems, this paper proposes a new method for extracting hypernyms of a noun from an MRD. In proposed approach, we use only syntactic(part-of-speech) patterns instead of lexico-syntactic patterns in identifying hypernyms to reduce the number of patterns while keeping their coverage broad. Our experiment shows that the classification accuracy of the proposed method is 92.37% which is significantly much better than those of previous approaches.

## 1 Introduction

The importance of broad-coverage lexical/semantic knowledge-bases has been stressed more than ever before as the natural language processing (NLP) systems became large and applied to wide variety of application domains. These lexical/semantic knowledge-bases include such as lexicons, thesauri and ontologies and machine-readable dictionaries. A lexical/semantic knowledge-base contains a list of terms, their semantic definitions and some of the relationships that exist between terms such as synonym, antonym and hypernym. Among the various relationships between terms, many researchers believe that the taxonomy relationship is especially useful because it can be utilized in various inference processes found in machine translation, information retrieval, word sense disambiguation and so on.

The taxonomy relationship is usually represented in thesauri as the broad-term (BT) narrow-term (NT) relations. However, because building broad-coverage thesauri is a very costly and time-consuming job, they are not readily available and often too general to be applied to a specific domain.

The work presented here is an attempt to alleviate this problem by devising a method for constructing taxonomy relations automatically from a machine readable dictionary (MRD). We use semantic hierarchies of nouns in a small thesaurus and a definition of a noun in a Korean MRD.

* This study was financially supported by special research fund of Chonnam National University in 2004.

A. Montoyo et al. (Eds.): NLDB 2005, LNCS 3513, pp. 1–9, 2005.
© Springer-Verlag Berlin Heidelberg 2005

Most of the previous approaches for extracting hypernyms of a noun from the definition in an MRD rely on the lexico-syntactic patterns compiled by human experts. Not only these methods require high cost for compiling lexico-syntactic patterns but also it is very difficult for human experts to compile a set of lexical-syntactic patterns with a broad-coverage because, in natural languages, there are various different expressions which represent the same concept. Accordingly the applicable scope of a set of lexico-syntactic patterns compiled by human is very limited.

To overcome the drawbacks of human-compiled lexico-syntactic patterns, we use part-of-speech (POS) patterns only and try to induce these patterns automatically using a small bootstrapping thesaurus and machine learning methods.

The rest of the paper is organized as follows. We introduce the related works to in section 2. Section 3 deals with the problem of features selection. In section 4, our problem is formally defined as a machine learning method and discuss implementation details. Section 5 is devoted to experimenal result. Finally, we come to the conclusion of this paper in section 6.

## 2 Related Work

[3] introduced a method for the automatic acquisition of the hyponymy lexical relation from unrestricted text, and gives several examples of lexico-syntactic patterns for hyponymy that can be used to detect these relationships including those used here, along with an algorithm for identifying new patterns. Her approach is complementary to statistically based approaches that find semantic relations between terms, in that hers requires a single specially expressed instance of a relation while the others require a statistically significant number of generally expressed relations. The hyponym-hypernym pairs found by Hearst's algorithm include some that she describes as "context and point-of-view dependent", such as "Washington/nationalist" and "aircraft/target". [4] was somewhat less sensitive to this kind of problem since only the most common hypernym of an entire cluster of nouns is reported, so much of the noise is filtered. [3] tried to discover new patterns for hyponymy by hand, nevertheless it is a costly and time-consuming job. In the case of [3] and [4], since the hierarchy is learned from text, it get to be domain-specific different from a general-purpose resource such as WordNet.

[2] proposed a method that combines a set of unsupervised algorithms in order to accurately build large taxonomies from any MRD, and a system that 1)performs fully automatic extraction of taxonomic links from MRD entries and 2) ranks the extracted relations in a way that selective manual refinement is allowed. In this project, he introduced the idea of the hyponym-hypernym relationship appears between the entry word and the genus term. Thus, usually a dictionary definition is written to employ a genus term combined with differentia which distinguishes the word being defined from other words with the same genus term. He finds the genus term by simple heuristic defined using several examples of lexico-syntactic patterns for hyponymy.

[1] presented the method to extract semantic information from standard dictionary definitions. His automated mechanism for finding the genus terms is based on the observation that the genus term from verb and noun definitions is typically the head of the defining phrase. The syntax of the verb phrase used in verb definitions makes it possible to locate its head with a simple heuristic: the head is the single verb follow-

ing the word *to*. He asserted that heads are bounded on the left and right by specific lexical defined by human intuition, and the substring after eliminating boundary words from definitions is regarded as a head.

By the similar idea to [2], [10] introduced six kinds of rule extracting a hypernym from Korean MRD according to a structure of a dictionary definition. In this work, she proposed that only a subset of the possible instances of the hypernym relation will appear in a particular form, and she divides a definition sentence into a head term combined with differentia and a functional term. For extracting a hypernym, she analyzes a definition of a noun by word list and the position of words, and then searches a pattern coinciding with the lexico-syntactic patterns made by human intuition in the definition of any noun, and then extracts a hypernym using an appropriate rule among 6 rules. For example, rule 2 states that if a word X occurs in front of a lexical pattern "*leul bu-leu-deon i-leum* ( *the name to call* )",then X is extracted as a hypernym of the entry word.

Several approaches[11][12][13] have been researched for building a semantic hierarchy of Korean nouns adopting the method of [2].

## 3 Features for Hypernym Identification

Machine learning approaches require an example to be represented as a feature vector. How an example is represented or what features are used to represent the example has profound impact on the performance of the machine learning algorithms. This section deals with the problems of feature selection with respect to characteristics of Korean for successful identification of hypernyms.

**Location of a Word.** In Korean, a head word usually appears after its modifying words. Therefore a head word has tendency to be located at the end of a sentence. In the definition sentences in a Korean MRD, this tendency becomes much stronger. In the training examples, we found that 11% of the hypernyms appeared at the start, 81% of them appeared at the end and 7% appeared at the middle of a definition sentence. Thus, the location of a noun in a definition sentences is an important feature for determining whether the word is a hypernym or not.

**POS of a Function Word Attached to a Noun.** Korean is an agglutinative language in which a word-phrase is generally a composition of a content word and some number of function words. A function word denotes the grammatical relationship between word-phrases, while a content word contains the central meaning of the word-phrase.

In the definition sentences, the function words which attached to hypernyms are confined to a small number of POSs. For example, nominalization endings, objective case postpositions come frequently after hypernyms but dative postpositions or locative postpositions never appear after hypernyms. A functional word is appropriate feature for identifying hypernyms.

**Context of a Noun.** The context in which a word appears is valuable information and a wide variety of applications such as word clustering or word sense disambiguation make use of it. Like in many other applications, context of a noun is important in deciding hyperhyms too because hypernyms mainly appear in some limited context.

Although lexico-syntactic patterns can represent more specific contexts, building set of lexco-syntactic patterns requires enormous training data. So we confined ourselves only to syntactic patterns in which hypernyms appear.

We limited the context of a noun to be 4 word-phrases appearing around the noun. Because the relations between word-phrases are represented by the function words of these word-phrases, the context of a noun includes only POSs of the function words of the its neighboring word-phrases. When a word-phrase has more than a functional morpheme, a representative functional morpheme is selected by an algorithm proposed by [8].

When a noun appears at the start or at the end of a sentence, it does not have right or left context respectively. In this case, two treatments are possible. The simplest approach is to treat the missing context as don't care terms. On the other hand, we could extend the range of available context to compensate the missing context. For example, the context of a noun at the start of a sentence includes 4 POSs of function words in its right-side neighboring word-phrases.

## 4 Learning Classification Rules

Decision tree learning is one of the most widely used and a practical methods for inductive inference such as ID3, ASSISTANT, and C4.5[14]. Because decision tree learning is a method for approximating discrete-valued functions that is robust to noisy data, it has therefore been applied to various classification problems successfully.

Our problem is to determine for each noun in definition sentences of a word whether it is a hypernym of the word or not. Thus our problem can be modeled as two-category classification problem. This observation leads us to use a decision tree learning algorithm C4.5.

Our learning problem can be formally defined as followings:

- Task T: determining whether a noun is a hypernym of an entry word or not .
- Performance measure P: percentage of nouns correctly classified.
- Training examples E: a set of nouns appearing in the definition sentences of the MRD with their feature vectors and target values.

To collect training examples, we used a Korean MRD provided by Korean TermBank Project[15] and a Korean thesaurus compiled by Electronic Communication Research Institute. The dictionary contains approximately 220,000 nouns with their definition sentences while the thesaurus has approximately 120,000 nouns and taxonomy relations between them. The fact that 46% of nouns in the dictionary are missing from the thesaurus illustrates the necessity of this research i.e. to extend a thesaurus using an MRD.

Using the thesaurus and the MRD, we found that 107,000 nouns in the thesaurus have their hypernyms in the definition sentences in the MRD. We used 70% of these nouns as training data and the remaining 30% of them as evaluation data. For each training pair of hypernym/hyponym nouns, we build a triple in the form of (hyponym definition-sentences hypernym) as follows.

| ga-gyeong | [a-leum-da-un gyeong-chi *(a beautiful scene)*] | gyeong-chi |
|---|---|---|
| hyponym | definition sentence | hypernym |

Morphological analysis and Part-Of-Speech tagging are applied to the definition sentences. After that, each noun appearing in the definition sentences is converted into a feature vector using features mentioned in section 3 along with a target value (i.e. whether this noun is a hypernym of the entry word or not).

Table 1 shows some of the training examples. In this table, the attribute *IsHypernym* which can have a value either Y or N is a target value for given noun. Hence the purpose of learning is to build a classifier which will predict this value for a noun unseen from the training examples.

In Table 1, *Location* denotes the location of a noun in a definition sentence. 0 indicates that the noun appears at the start of the sentence, 1 denotes at the middle of the sentence, and 2 denotes at the end of a sentence respectively. *FW of a hypernym* is the POS of a function word attachted to the noun and *context1,...,context4* denote the POSs of function words appearing to the right/left of the noun. "*" denotes a don't care condition. The meanings of POS tags are list in Appendix A.

**Table 1.** Some of training examples.

| Noun | *Location* | *FW of a hypernym* | *context1* | *context2* | *context3* | *context4* | *IsHypernym* |
|---|---|---|---|---|---|---|---|
| N1 | 1 | jc | ecx | exm | nq | * | Y |
| N2 | 2 | * | exm | ecx | jc | nq | Y |
| N3 | 2 | * | exm | jc | nca | exm | Y |
| N4 | 1 | exm | jc | jc | ecx | m | N |
| N5 | 1 | jc | jc | ecx | m | jca | N |
| N6 | 1 | jc | ecx | m | jca | exm | Y |
| N7 | 2 | * | exm | exm | jca | exm | Y |
| N8 | 1 | * | nc | jca | exm | jc | N |
| N9 | 1 | jca | nc | nc | nc | jc | Y |
| N10 | 2 | exn | a | nca | jc | nca | Y |
| .. | .. | .. | .. | .. | .. | .. | .. |

Fig. 1 shows a part of decision tree learned by C4.5 algorithm. From this tree, we can easily find that the most discriminating attribute is *Location* while the least one is *Context*.

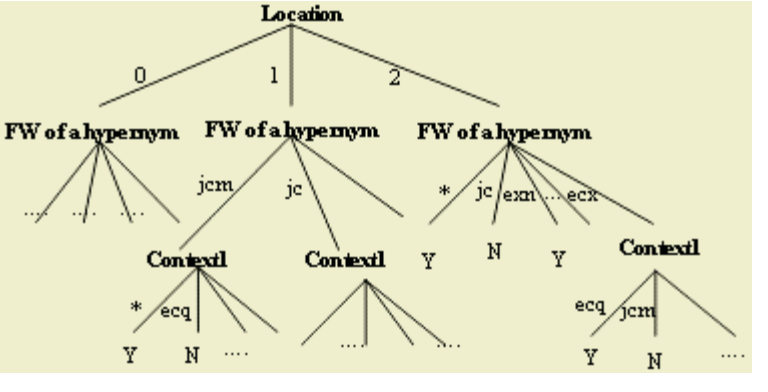



**Fig. 1.** A learned decision tree for task T.

## 5 Experiment

To evaluate the proposed method, we measure classification accuracy as well as precision, recall, and F-measure which are defined as followings respectively.

$$classification\ accuracy = \frac{a+d}{a+b+c+d}$$

$$precision = \frac{a}{a+b}$$

$$recall = \frac{a}{a+c}$$

$$F-Measure = \frac{2 * precision * recall}{precision + recall}$$

**Table 2.** Contingency table for evaluating a binary classifier.

| | Yes is correct | No is correct |
|---|---|---|
| Yes was assigned | a | b |
| No was assigned | c | d |

Table 3 shows the performance of the proposed approach. We conducted two experiments using 2 different definitions for the context of a word as stated in section 3. In the experiment denoted A in table 3, the context of a word is defined as 4 POSs of the function words, 2 of them immediately proceeding and 2 of them immediately following the word. In the experiment denoted B, when the word appears at the beginning of a sentence or at the end of a sentence, we used only right or left context of the word respectively. Our experiement shows that the performance of B is slightly better than that of A.

**Table 3.** Evaluation result.

| | Classification accuracy | Precesion | Recall | F-Measure |
|---|---|---|---|---|
| A | 91.91% | 95.62% | 92.55% | 94.06% |
| B | 92.37% | 93.67% | 95.23% | 94.44% |

Table 4 compares the classification accuracy of the proposed method with those of the previous works. Our method outperforms the performance of the previous works reported in the literature[10] by 3.51%.

Because the the performance of the previous works are measured with small data in a restricted domain, we reimplemented one of the those previous works[10] to compare the performances using same data. The result is shown in Table 4 under the column marked D. Column C is the performance of the [10] reported in the literature. This result shows that as the heuristic rules in [10] are dependent on lexical information, if the document collection is changed or the application domain is changed, the performance of the method degrades seriously.

**Table 4.** Evaluation result.

| | Proposed | | M.S.Kim 95[11] | Y.J.Moon 96[10] | | Y.M.Choi 98[13] |
|---|---|---|---|---|---|---|
| | A | B | | C | D | |
| classification accuracy | 91.91% | 92.37% | 88.40% | 88.40% | 68.81% | 89.40% |

## 6 Conclusion

There have been several works to build a noun taxonomy from an MRD. However, most of them relied on the lexico-syntactic patterns compiled by human experts. Not only these methods require high cost for compiling lexico-syntactic patterns but also it is very difficult for human experts to compile a set of lexical-syntactic patterns with a broad-coverage, because in natural languages there are various different expressions which represent the same concept. Accordingly the applicable scope of a set of lexico-syntactic patterns compiled by human is very limited.

This paper has proposed a new method for extracting hypernyms of a noun from an MRD. In proposed approach, we use only syntactic patterns instead of lexico-syntactic patterns in identifying hypernyms to reduce the number of patterns while keeping their coverage broad. We also adopted a machine learning method to collect the patterns automatically.

Our experiment shows that the classification accuracy of the proposed method is 92.37% which is significantly much better than those of previous approaches. Throughout our research, we have found that machine learning approaches to the problems of identifying hypernyms from an MRD could be a competitive alternative to the methods using human-compiled lexico-syntactic patterns.

## References

1. Martin S. Chodorow, Roy J. Byrd, George E. Heidorn.: Extracting Semantic Hierarchies From A Large On-Line Dictionary. In Proceedings of the 23rd Conference of the Association for Computational Linguistics (1985)
2. Rigau G., Rodriguez H., Agirre E.: Building Accurate Semantic Taxonomies from Mololingual MRDs. In Proceedings of the 36th Conference of the Association for Computational Linguistics (1998)
3. Marti A. Hearst.: Automatic acquisition of hyonyms from large text corpora. In Proceedings of the Fourteenth International Conference on Computational Linguistics (1992)
4. Sharon A. Caraballo.: Automatic construction of a hypernym-labled noun hierarchy from text. In Proceedings of the 37th Conference of the Association for Computational Linguistics (1999).
5. Fernando Pereira, Naftali Thishby, Lillian Lee.: Distributional clustering of English words. In Proceedings of the 31th Conference of the Association for Computational Linguistics (1993)
6. Brian Roark, Eugen Charniak.: Noun-phrase co-occurrence statistics for semi-automatic semantic lexicon construction. In Proceedings of the 36th Conference of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (1998)
7. Tom M. Mitchell.: Machine Learning. Carnegie Mellon University. McGraw-Hill (1997).
8. SeonHwa Choi, HyukRo Park.: A New Method for Inducing Korean Dependency Grammars reflecting the Characteristics of Korean Dependency Relations. In Proceedings of the 3rd Conterence on East-Asian Language Processing and Internet Information Technology (2003)
9. YooJin Moon, YeongTak Kim.:The Automatic Extraction of Hypernym in Korean. In Preceedings of Korea Information Science Society Vol. 21, NO. 2 (1994) 613-616
10. YooJin Mon.: The Design and Implementation of WordNet for Korean Nouns. In Proceedings of Korea Information Science Society (1996)

11. MinSoo Kim, TaeYeon Kim, BongNam Noh.: The Automatic Extraction of Hypernyms and the Development of WordNet Prototype for Korean Nouns using Koran MRD. In Proceedings of Korea Information Processing Society (1995)
12. PyongOk Jo, MiJeong An, CheolYung Ock, SooDong Lee.: A Semantic Hierarchy of Korean Nouns using the Definitions of Words in a Dictionary. In Proceedings of Korea Cognition Society (1999)
13. YuMi Choi and SaKong Chul.: Development of the Algorithm for the Automatic Extraction of Broad Term. In Proceedings of Korea Information Management Society (1998) 227-230
14. Quinlan J. R.: C4.5: Programs for Machine Learning. San Mateo, CA: Morgan Kaufman (1993) http://www.rulequest.com/Personal/
15. KORTERM.: KAIST language resources http://www.korterm.or.kr/

## Appendix A: POS Tag Set

**Table 5.** POS tag set.

| CATEGORY | | TAG | DESCRIPTION |
|---|---|---|---|
| noun | common | nn | common noun |
| | | nca | active common noun |
| | | ncs | statove common noun |
| | | nct | time common noun |
| | proper | nq | proper noun |
| | bound | nb | bound noun |
| | | nbu | unit bound noun |
| | numeral | nn | numeral |
| | pronoun | npp | personal pronoun |
| | | npd | demonstrative pronoun |
| predicate | verb | pv | verb |
| | adjective | pa | adjective |
| | | pad | demonstrative adjective |
| | auxiliary | px | auxiliary verb |
| modification | adnoun | m | adnoun |
| | | md | demonstrative adnoun |
| | | mn | numeral adnoun |
| | adverb | a | general adverb |
| | | ajs | sentence conjunctive adverb |
| | | ajw | word conjunctive adverb |
| | | ad | demonstrative adverb |
| independence | interjection | ii | interjection |
| particle | case | jc | case |
| | | jca | adverbial case particle |
| | | jcm | adnominal case particle |
| | | jj | conjunctive case particle |
| | | jcv | vocative case particle |
| | auxiliary | jx | auxiliary |
| | predicative | jcp | predicative particle |

**Table 5.** (continued).

| CATEGORY | | TAG | DESCRIPTION |
|---|---|---|---|
| ending | Prefinal | efp | prefinal ending |
| | conjunctive | ecq | coordinate conjunctive ending |
| | | ecs | subordinate conjunctive ending |
| | | ecx | auxiliary conjunctive ending |
| | transform | exn | nominalizing ending |
| | | exm | adnominalizing ending |
| | | exa | adverbalizing ending |
| | final | ef | final ending |
| affix | prefix | xf | prefix |
| | suffix | xn | suffix |
| | | xpv | verb-derivational suffix |
| | | xpa | adjective-derivational suffix |

# On the Transformation of Sentences with Genitive Relations to SQL Queries

Zsolt T. Kardkovács

Budapest University of Technology,
Department of Telecommunications and Mediainformatics
kardkovacs@tmit.bme.hu

**Abstract.** In our ongoing project called "In the Web of Words" (WoW) we aimed to create a complex search interface that incorporates a deep web search engine module based on a Hungarian question processor. One of the most crucial part of the system was the transformation of genitive relations to adequate SQL queries, since e.g. questions begin with "Who" and "What" mostly contain such a relation. The genitive relation is one of the most complex semantic structures, since it could express wide range of different connection types between entities, even in a single language. Thus, transformation of its syntactic form to a formal computer language is far from clear. In the last decade, several natural language database interfaces (NLIDBs) have been proposed, however, a detailed or a general description of this problem is still missing in the literature. In this paper, we describe how to translate genitive phrases into SQL queries in general, i.e. we omit Hungarian-dependent optimizations.

## 1 Introduction

Deep web sites (e.g. library and news portals, book stores, theatre and movie guides) are usually based on data stores. Contents of these sites are rarely visible by search engines despite of the fact that they are publicly accessible on the internet since they are often dynamically generated by forms or by dynamic parameters in the address line. Information stored in these databases contain about four hundred times more data than are searchable nowadays by search engines[1]. In addition, databases are yet structured and categorized and so they could provide a much more accurate answer to any user request. The main question is how to query these sites, in general?

In the project WoW, we aimed at creating a natural language interface, mainly focus on Hungarian, to query the deep web. To achieve this goal one has to transform interrogative sentences into a formalized language, for example into XQuery, XPath or simply into SQL. Later in this paper, we focus only on the SQL as a target language, however, our results can be easily adopted for both XQuery and XPath according to [2].

One may ask why we need natural language (interface) to query the deep web. Are not keyword based engines and form or menu driven interfaces enough?

A. Montoyo et al. (Eds.): NLDB 2005, LNCS 3513, pp. 10–20, 2005.
© Springer-Verlag Berlin Heidelberg 2005

Creating database queries from keywords (a set of words) instead of sentences would result in a remarkably worse precision and recall rate. For example, keywords *"president"*, *"Russia"*, *"visit"* and *"US"* could mean

- *"When did the president of Russia visit the US?"*,
- *"Which president visited Russia and the US?"*,
- *"Which president of the US visited Russia?"*,
- *"Who can visit the president of Russia from the US?"*, etc.

In these examples, all alternative sentences affect different database queries and thus many irrelevant answers, too, since the user asked only one of them. Solutions that do not deal with the given morphosyntax of the sentence, e.g. AnswerBus[3], Askjeeves[4], BrightPlanet[5], Ionaut[6], also suffer from this problem, even if input question contains a genitive phrase. (To test whether the solution is keyword based or not – even if the input is a natural language question – try: *"What is the name of the son of Juan Carlos I?"*. The results usually contain a wide variety of person names whose sons' name is Juan Carlos or details of the life on the present King of Spain which includes or not his son's name.)

Form based or other graphical interfaces are the best choices to retrieve information from a single database or to query web sites on a concrete topic. On the one hand, there are differences in structural design and in granularity of data between databases in a multi-database environment. That is why a form based search application needs to semantically restructuring user input according to the target databases or it needs to reduce differences in a way. Both of them are hard tasks if the number of databases is not strongly limited or if there is no agreement between the site owners. On the other hand, the number of attributes to be queried is not bounded in search engines. Without a hard limitation on the number of the topics, form based applications become unusable or impractical. In addition, natural questions as above can not be displayed by forms easily. For a more detailed analysis on the differences between NLIDBs, keyword and menu based search engines, see [5, 7, 8].

From our point of view deep web can be treated as a most universal database which has no definite data structure (or it is not known) and thus our project produces a natural language interface to databases (NLIDBs). On the other hand, deep web could serve as a knowledge base or as a source of knowledge for question answering systems (QAS). The only difference between deep web engines and QAS is that deep web engines retrieve sites with appropriate knowledge to answer questions instead of finding answers for them. That is, deep web querying incorporates some aspect of both NLIDBs and QAS.

There has been a lot of work in recent years on both natural language processing and web-based queries. A non-exhaustive list of projects: Practice[9, 10], START[11–13], SQ-HAL[14], NL for Cindi[15], Masque/SQL[16], Chat-80[17] or Team[18] provide solutions for English, Spanish NLQ[19], Sylvia-NLQ[20] for Spanish, Phoenix[21] for Swedish, Edite[22], LIL/SQL[23] for Portuguese, NChiql[24] for Chinese and KID[25, 26] for Korean languages. Although several ideas and techniques have been adapted from these systems to WoW, none of

them contains a general and clear description on how to transform correctly genitive relations to their formal languages. In this paper, we focus on this problem and propose a possible solution.

Our work differs from others in several ways. Firstly, we state new theoretical foundations to deal with non-trivial genitive phrases. (We call a genitive phrase non-trivial if its SQL equivalent contains at least one embedded query.) Our solution is not limited to giving a formal semantics for genitive relations but includes transformation methods from syntactically analyzed sentences to SQL statements. Last but not least, our approach can also handle compound (or multiple) genitive phrases which is beyond the capabilities of above mentioned projects. The solution is designed for our native language, however, it is language independent in its own, and we omit language-dependent optimizations.

The paper is organized as follows: in the next section we introduce basic notions. We state main problems and a new algorithm based on the design of targeted databases in Section 3. Section 4 shows how our algorithm works. We summarize our results in the last section.

## 2 Preliminaries

Genitive relations could express a wide range of different things in natural languages, see e.g. [27–29], which emerges problems on natural language processing as a new source of ambiguity. For example,

- *"head **of** the firm"* (affiliation or group membership),
- *"the author **of** Hamlet"* (attribution),
- *"Edith**'s** TV"* (ownership),
- *"Bizet**'s** Carmen"* (authorship),

etc. refer different relationships between entities, however, their syntactical form (and thus their syntactical decompositions) is the same both in Hungarian and in English. This also means that after the syntactical analysis one can determine just semantically which aggregation or reference in databases corresponds to the relationship encoded in the given expression.

Identification and syntactical decomposition of genitive phrase elements are language dependent tasks which were widely discussed earlier for English in [27, 29, 30] and for Hungarian [31]. After identifying possessors and possessums of a sentence one needs to determine what genitive relations express, "mean" or refer to, and how to transform them to query databases. Obviously, the meaning of an expression depends on the terms used in it. The Table 1 illustrates by simple and similar examples that the transformation is far from trivial.

The term *"Mecca of movies"* needs some further explanations. In the one hand, it refers in Hungarian to a city visualized usually in movies which can be a bit different from the reality. On the other hand, it also could mean (as an idiomatic expression) a location where movie shooters return to over and over like Islamites return to Mecca. Unfortunately, neither of these senses is fully captured by our methods since it assumes a common human knowledge and associative (metaphoric) connections between entities.

**Table 1.** Genitive phrases and one of their equivalent SQL statements

| | |
|---|---|
| Bizet**'s** Carmen | `SELECT title FROM Operas`<br>`WHERE author = 'Bizet' AND title = 'Carmen'` |
| dramas **of** Shakespeare | `SELECT title FROM Dramas`<br>`WHERE author = 'Shakespeare'` |
| Tom**'s** name | `SELECT name FROM Persons`<br>`WHERE name = 'Tom'` |
| Mecca **of** movies | `SELECT shootingsite FROM Movies`<br>`GROUP BY shootingsite`<br>`HAVING COUNT(*) >= ALL (SELECT COUNT(*)`<br>`FROM Movies GROUP BY shootingsite)` |
| book**'s** characters | `SELECT character FROM Roles`<br>`WHERE play IN ( SELECT title FROM Books )` |
| head **of** department | `SELECT head FROM Departments` |
| king **of** Spain's name | `SELECT name FROM Kings WHERE name IN`<br>`(SELECT king FROM Reigns WHERE kingdom = 'Spain')` |

On translating interrogative sentences into SQL queries one has to find a mapping between the elements of database design and the ontological concepts or else SQL queries will not fit the database structure. This implies that a transformation of the ontological taxonomy needs to be represented in the database design and conversely the elements of the database structure are the basis of the ontological taxonomy.

One may ask then: does the knowledge stored in the database design suffice to achieve the decomposition of genitive relations?

Let $X \Rightarrow Y$ stand for a genitive relation in which $X$ is the possessor part and $Y$ is the possessum part. According to relational database terms, genitive expressions can consist of values (of attributes), individuals (tuples, entities and instances), schemas (sets, classes) and attributes (properties). Let us denote them by $\mathbf{V}$, $\mathbf{I}$, $\mathbf{S}$ and $\mathbf{A}$, respectively.

Working implementations, e.g. [9, 10, 12, 21, 22], say *yes* to that question, hence their knowledge for semantic analysis is originated from the database design. If data are modeled by <object, property, value> triples (was proposed by [13, 32]) then they can only query for elements of these triples, which is a quite restricted form of the genitive relations. That is, these solutions *only* handle genitive relations of the form $\mathbf{S} \Rightarrow \mathbf{A}$ and $\mathbf{I} \Rightarrow \mathbf{S}$, but they can not deal with multiple or compound genitive relations, e.g. *"king of Spain's name"*.

Before introducing a more general semantics for natural genitives we need some clarification in notions and introduction of some new terms.

**Definition 1 (Natural keys).** *Let* $\mathcal{DB} = <\mathbf{V}, \mathbf{I}, \mathbf{S}, \mathbf{A}>$ *be a database, where* $\mathbf{V}$, $\mathbf{I}$, $\mathbf{S}$ *and* $\mathbf{A}$ *are non-empty sets of stored attribute values, individuals, schemas and attributes, respectively. Let* $\kappa : \mathbf{S} \rightarrow \mathbf{A}$ *be a function that maps every schema* $s \in \mathbf{S}$ *to an attribute* $\alpha \in \mathbf{A}$ *in* $s$ *such that any individual* $\omega \in \mathbf{I}$ *of* $s$ *is named in natural languages by the value* $\mathbf{v} \in \mathbf{V}$ *of* $\alpha$ *in* $\omega$*. We say* $\kappa(s)$ *is the natural key of* $s$*.*

For example, $\kappa(book) = title$, $\kappa(person) = name$, etc. Natural keys and well-known (primary) keys differ in that the former ones could have the same value for distinct entities in a schema, while the latter ones are unique by definition.

**Definition 2 (Reference function).** *Let* $\mathcal{DB} = <\mathbf{V}, \mathbf{I}, \mathbf{S}, \mathbf{A}>$ *be a database and* $\alpha \in \mathbf{A}$ *be an attribute of a schema* $\mathbf{s} \in \mathbf{S}$*. Assume that attribute names appear in at most one schema (unique property name assumption). The function* $\varphi : \mathbf{A} \rightarrow \mathbf{S}$ *is called reference function and it is defined as follows:*

$$\varphi(\alpha) = \begin{cases} \mathbf{s} & \text{if } \alpha = \kappa(\mathbf{s}) \\ \mathbf{s'} & \text{if } \alpha \neq \kappa(\mathbf{s}), \text{ for some } \mathbf{s'} \in \mathbf{S} \end{cases}$$

*The function* $\varphi$ *is well-founded iff for any value of* $\alpha$ *is also value of* $\varphi(\alpha)$*. We also define the inverse function* $\varphi^{-1}(\mathbf{s}) = \{\alpha \mid \varphi(\alpha) = \mathbf{s}\}$*.*

Values of natural keys are treated as constants, i.e. they are references to themselves. Individuals are usually distinguished from each other in databases by a unique id. If there is a reference function in a database then unique ids have to be references, since they are not natural at any sense. As a consequence, databases with reference function must have a schema for unique ids or else $\mathbf{V} = \mathbf{I}$. These options are equivalent with respected to the genitive relations and make no real distinctions between values, references and individuals. Later on this paper, we call this construction **(V)ISA**-model of genitive phrases, or shortly **(V)ISA**-model, that is, any database with a reference function is a **(V)ISA**-model.

**Proposition 1.** *Let* $\mathcal{DB} = <\mathbf{V}, \mathbf{I}, \mathbf{S}, \mathbf{A}>$ *be a* **(V)ISA***-model with a reference function* $\varphi$*.* $\varphi$ *defines an equivalence relation on* $\mathbf{A}$*. Further on this paper,* $\|\alpha\|$ *stands for the equivalence class of an attribute* $\alpha$*.*

The proof is pretty straightforward. It is also easy to see that every equivalence class contains only one natural key. Note that, reference function can be used for navigation between elements of relations, since $\varphi$ is also a constraint that determines which attributes can be joined (in a semantically valid way).

**Definition 3 (Weak well-foundedness of genitive relations).** *Let* $\mathcal{DB} = <\mathbf{V}, \mathbf{I}, \mathbf{S}, \mathbf{A}>$ *be a* **(V)ISA***-model with a reference function* $\varphi$*. The genitive relation* $X \Rightarrow Y$ *is called weakly well-founded iff*

- $X \subseteq \mathbf{I}$ *and* $Y \in \mathbf{I}$ *(e.g. "Bizet's Carmen")*
- $X \subseteq \mathbf{I}$ *and* $Y \in \mathbf{S}$ *(e.g. "dramas of Shakespeare")*
- $X \subseteq \mathbf{I}$ *and* $Y \in \mathbf{A}$ *(e.g. "Tom's name")*
- $X \subseteq \mathbf{S}$ *and* $Y \in \mathbf{I}$ *(e.g. "Mecca of movies")*
- $X \subseteq \mathbf{S}$ *and* $Y \in \mathbf{S}$ *(e.g. "book's characters")*
- $X \subseteq \mathbf{S}$ *and* $Y \in \mathbf{A}$ *(e.g. "head of department")*

Weakly well-founded genitive relations are the most common genitive phrases in natural languages. Usually, attributes or attributive descriptions can not stand

for possessors, hence their functionality is quite different – they can not have ownership, partitive, constitutional, membership, etc. relations[27] with some other element. Pragmatically speaking, to be a schema means to be described by parameters, measures, qualities or values; to be an attribute means to belong to some schema and to have some value; and at last to be an individual is to be a named element of a more abstract category or schema.

In a genitive phrase $X \Rightarrow Y$, if $X$ is a schema (schema name) then usually it has a general meaning. Formally, schema (as a set) is equivalent to the set of entities (values of natural keys) belong to it, however, in natural languages it not necessarily means the same. For example, *"head of department"* is not the head of all but one of known departments (department names); while in the case of *"Mecca of movies"* are not the Mecca of all and not any of known movies (movie titles). That is, schemas can not be modeled as sets in general. Actually, schemas can be individuals from the designer's (or the speaker) point of view and vice versa. In other words, we treat schema as a more abstract individual from some others, and thus, we can create a hierarchy between individuals based on abstraction. As a consequence, we have $\mathbf{S} \subseteq \mathbf{I}$.

**Definition 4 (Strong well-foundedness of genitive relations).** *A weakly well-founded genitive relation $X \Rightarrow Y$ is called strongly well-founded or well-founded iff $X \subseteq \mathbf{I}$.*

## 3 On the Semantics of Genitive Relations

It is easy to see according to the previous contexture that weak and strong well-foundedness are not necessarily different terms, it is not a real limitation on the expressive power. While well-foundation determines what kind of element can be a possessor or a possessum, it does not deal with the real semantics. We need to introduce what valid "ownership" could mean based on the works of [33, 34].

**Definition 5 (Valid genitive relations).** *Let a relation $\Pi : \mathbf{A} \times \mathbf{A}$ be defined on a $\mathcal{DB} = <\mathbf{V}, \mathbf{I}, \mathbf{S}, \mathbf{A}>$ (V)ISA-model. For $\alpha, \beta \in \mathbf{A}$ and there exists a $\mathbf{s} \in \mathbf{S}$ they are both belong to then $\Pi(\alpha, \beta)$ is true if and only if values of $\alpha$ can be possessors in well-founded genitive phrases for which $\beta$ or values of $\beta$ are the possessum.*

For example, $\Pi$(*"person's name"*, *"date of birth"*) is a valid possessive relation but in reverse is not. Also note that, $\Pi$ is a language-dependent relation, hence it raises semantic issues and as such it is the semantic constraint proposed in [27, 34]. There are some more thematic roles, e.g. in English than are in Hungarian[27, 28]:

- *"ring **of** gold"* (materialization or constitution)
- *"Mr. Jones **of** Suffolk County"* (origination or affiliation)
- *"the herd **of** cattle"* (natural measure)
- *"man **of** yesterday"* (temporal predication)

Their special meanings are not fully captured by our proposal. Notwithstanding, one can represent these connections in a modified database model extending $\Pi$ (e.g. in the case of *"ring of gold"*) or one needs to treat them as idioms (e.g. *"man of yesterday"*) and processed in an alternative way.

In databases, $\Pi$ can be stored as a schema with two attributes and $\Pi$ is true for all tuples which are defined in the relation.

**Definition 6 (Semantics of genitive relations in (V)ISA-model).** *Let $\mathcal{DB} = <\mathbf{V}, \mathbf{I}, \mathbf{S}, \mathbf{A}>$ be a* **(V)ISA***-model with a reference function $\varphi$. We introduce the following functions and notions:*

- *We denote by $\alpha \in s$ for some $s \in \mathbf{S}$ the fact that $s$ has an attribute named $\alpha$.*
- *Let $\Sigma : 2^{\mathbf{A}} \to 2^{\mathbf{S}}$ be a function that maps a set of attributes onto a set of such schemas that contain at least one element of the attribute set.*
- *The $\sigma : \mathbf{I} \to 2^{\mathbf{A}}$ is a function that maps any individual $i \in \mathbf{I}$ onto a set of attributes with values $i$. That is, $\sigma$ defines the set of attributes where an individual $i$ may appear as a value.*

1. *if $Y \in \mathbf{I}$ then let $\gamma := \sigma(Y)$*
2. *if $Y \in \mathbf{S}$ then let $\gamma := \varphi(\kappa(Y))$*
3. *if $Y \in \mathbf{A}$ then let $\gamma := \|Y\|$*

*If there exist attributes $\alpha \in \sigma(X)$, $\beta \in \gamma$ such that $\exists s \in \Sigma(X) \cap \Sigma(\gamma)$, $\alpha, \beta \in s$ and $\Pi(\alpha, \beta)$ hold then $X \Rightarrow Y$ is the set of values (individuals) of $\beta$ assuming that $X$ is a single individual. If $X$ stands for a set of individuals then*

$$X \Rightarrow Y = \bigcup_{\chi \in X} \chi \Rightarrow Y.$$

Note that the semantics does not depend explicitly on the database design; hence there is no mention of requirements for world modeling. In other words, attributes $\alpha$, $\beta$ and the container schema $s$ are free variables; their concrete names or values depend on the site we are trying to query only. As a consequence, the algorithm is quite effective in the sense it has to find a proper triple for a quite limited number of attributes and schemas. Such a constraint resolution can be very effective by constraint logic programming (Prolog). On the other hand, it needs transformations for each site we want to search in, however, every translation can be done and be supervised by these sites.

The novelty of this approach based on this fact: it takes into account that data can be separately stored but there can be a well-defined path or a path expression to navigate from an entity to another from a natural point of view. If this last condition does not hold it can not be resolved the association even for a human. Obviously, if natural keys served for navigating are replaced by commonly used primary keys then most of our algorithm still works. Notwithstanding, that would be a less precise solution since it would produce a large amount of false positive results (consider e.g. dates or numbers).

**Algorithm 1 ((V)ISA-algorithm)** *The following algorithm transforms a genitive phrase of the form $X \Rightarrow Y$ into an equivalent database (site) specific SQL code. Let $\mathcal{D}$ denote the given database. We use notions of the Definition 6.*

1. *determine $\gamma$ according to Definition 6*
2. *find appropriate $\alpha$, $\beta$ and $s$ due to Definition 6 in $\mathcal{D}$*
3. *if $Y$ is an individual then* `$insert := "`$\beta$ `= Y AND "`
4. *else* `$insert := ""`
5. *if $X$ is a single individual then*
   `SELECT` $\beta$ `FROM` $s$ `WHERE $insert` $\alpha$ `= X`
6. *else* {
7. *if $X$ is a schema then replace it by* `SELECT` $\kappa(X)$ `FROM` $X$
8. *if $X$ is a genitive phrase then* {
9. *apply* ***(V)ISA****-algorithm for it*
10. *and replace $X$ by the result of the algorithm*
    }
11. `SELECT` $\beta$ `FROM` $s$ `WHERE $insert` $\alpha$ `IN ( X )`
    }

## 4 (V)ISA-Algorithm by Examples

Consider e.g. the expression *"Bizet's Carmen"*. There exists an attribute $\alpha =$ *"author"* and $\beta =$ *"title"* in *"Operas"* $\in \Sigma(\sigma($*"Bizet"*$)) \cap \Sigma(\|$*"Carmen"*$\|)$ for which $\Pi($*"author"*, *"title"*$)$ also holds. Since *"Carmen"* is an individual, `$insert` contains `"title = 'Carmen'"`. That is, the solution is:

```
SELECT title FROM Operas
WHERE title = 'Carmen' AND author = 'Bizet'
```

which is exactly the same as in Table 1.

It is easy to see, that for the expression *"head of department"* **(V)ISA**-algorithm produces

```
SELECT name FROM Departments
WHERE name IN ( SELECT name FROM Departments )
```

which is an equivalent to the solution seen in Table 1. Unfortunately, this solution is not equivalent to the original genitive expression as we pointed out earlier. Notwithstanding, there can be no better solution, thus this defectiveness is rather originated from the database formalism or data model than from the algorithm.

The algorithm also works for multiple or compound genitives. For example, let us examine the phrase *"king of Spain's name"* ((*"Spain"* $\Rightarrow$ *"king"*) $\Rightarrow$ *"name"*). For *"Spain"* $\Rightarrow$ *"king"* $\alpha =$ *"kingdom"*, $\beta =$ *"king"*, $s =$ *"Reigns"*. After that one has to consider all possible individuals of *"Reigns"* by definition. As a consequence, algorithm generate identical solution to which was seen in Table 1.

The algorithm generates appropriate SQL statements for schema-reflexive multiple genitive relations due to the definition of $\Pi$. Consider, e.g. the question

*"Who is the husband of Tom's wife?"*). The genitive phrase *"Tom"* ⇒ *"wife"* ⇒ *"husband"* is transformed by **(V)ISA**-algorithm to:

```
SELECT husband FROM Consorts
WHERE wife IN ( SELECT wife FROM Consorts
                WHERE husband = 'Tom' )
```

since $\Pi($*"husband"*, *"wife"*$)$ and $\Pi($*"wife"*, *"husband"*$)$ are valid in the schema *"Consorts"* while e.g. $\Pi($*"husband"*, *"husband"*$)$ is not.

Our algorithm is not universal (or fully semantic) in the sense that it neither handles metaphoric or common human knowledge based expression (e.g. *"Mecca **of** movies"*) nor idiomatic expressions (e.g. *"man **of** yesterday"*). Moreover, it also fails for derivatives of complex verbs, e.g. for *"winner"*. The problem is such derivatives could have different representations based on the local context, e.g. win a match could mean scoring more goals, getting less error points, being faster than others or win three games in a row. These notions can be captured by ontologies but integration of them into this model is far from trivial, we are still working on it.

## 5 Conclusions

Algorithms or foundations on the decomposition of possessives are missing from the literature. In this paper, we stated problems and we proposed solutions which deal with genitive phrases in natural language processing and translate them properly into SQL queries.

Our approach is universal in the sense that it works for all kinds of genitive phrases and also handles compound structures which is one of the most important novelties of this approach. Obviously, it has limitations. Our algorithm can not resolve expressions with wider or metaphoric sense, concetti, idiomatic expressions, terms which assume deeper human knowledge and derivatives of predicate verbs.

Our proposal does not require ontology but with additional information on reference functions, natural naming conventions and valid possessive relations one can determine unambiguously the possessor of genitive phrases assuming that the predicate verb has no special meaning in the sentence. We also demonstrated by examples how our proposition works.

Unfortunately, there still not exists a query corpus with genitive phrases for Hungarian and are rarely available even for English. That is, we could not present a detailed comparison to other solutions, however, our corpus with 87 genitive phrases has been properly processed in 87,3% (11 wrong solutions). We are working on building a reference corpus with both language dependent and independent parts.

## Acknowledgement

The research was partially funded by the Hungarian Research and Development Foundation under the name "Szavak hálójában" (NKFP-2002/19) and "Magyar

egységes ontológia" (NKFP-042/04). The project was also funded by Axelero Internet Ltd. (T–Mobile) the largest Hungarian Web Service Provider.

## References

1. (Brightplanet – deep web white paper) http://www.brightplanet.com/pdf/.
2. (Iso/iec 9075-14:2003 standard) http://www.iso.org.
3. (Answerbus) http://www.answerbus.com/.
4. (Askjeeves) http://www.askjeeves.com/.
5. (White paper on deep content retrieval) http://www.brightplanet.com/pdf/.
6. (Ionaut) http://www.ionaut.com:8400/.
7. Androutsopoulos, I., Ritchie, G.D., Thanisch, P.: Natural language interfaces to databases – an introduction. Journal of Natural Language Engineering **1** (1995) 29–81
8. Winkler, H. In: Suchmaschinen – Metamedien im Internet? Campus Verlag, Frankfurt/New York (1997) 185–202
   In English: http://www.unipadernborn.de/~timwinkler/schum_e.html.
9. Popescu, A.M., Etzioni, O., Kautz, H.: Towards a theory of natural language interfaces to databases. In Johnson, W., André, E., Domingue, J., eds.: Proc. of the $8^{th}$ IUI 2003. Internation Conferences on Intelligent User Interfaces, Miami, ACM Press (2003) 149–157
10. Popescu, A.M., Armanasu, A., Etzioni, O., Ko, D., Yates, A.: Modern natural language interfaces to databases: Composing statistical parsing with semantic tractability. In: Proc. of COLING 2004. International Conference on Computational Linguistics, Geneva, Switzerland (2004)
11. Katz, B. In: Using English for Indexing and Retrieving. Volume 1 of Artificial Intelligence at MIT: Expanding Frontiers. MIT Press (1990) 134–165
12. (Start) http://www.ai.mit.edu/projects/infolab/.
13. Katz, B., Lin, J.L.: Start and beyond. In: Proc. of SCI 2002. Volume XVI of World Multiconference on Systemics, Cybernetics and Informatics. (2002)
14. (Sqhal)
    http://www.csse.monash.edu.au/hons/projects/2000/Supun.Ruwanpura.
15. Stratica, N., Kosseim, L., Desai, B.C.: A natural language processor for querying cindi. In Milutinovic, V., ed.: Proc. of SSGRR 2002. International Conference on Advances in Infrastructure for e-Business, e-Education, e-Science, and e-Medicine on the Internet, L'Aquila, Italy, Electronically: http://www.ssgrr.it/ (2002)
16. Androutsopoulos, I., Ritchie, G.D., Thanisch, P.: Masque/sql – an efficient and portable natural language query interface for relational databases. In Chung, P., Lovegrove, G., Ali, M., eds.: Proc. of IEA/AIE 93 Conference. IEA/AIE Conferences, Edinburgh, Gordon and Breach Publishing (1993) 327–330
17. Warren, D.H., Pereira, F.C.: An efficient easily adaptable system for interpreting natural language queries. Computational Linguistics **8** (1982) 110–122
18. Grosz, B.J., Appelt, D.E., Martin, P.A., Pereira, F.C.: Team: An experiment in the design of transportable natural-language interfaces. Artifical Intelligence **32** (1987) 173–243
19. Rangel, R.A.P., Gelbukh, A.F., Barbosa, J.J.G., Ruiz, E.A., Mejía, A.M., Sánchez, A.P.D.: Spanish natural language interface for a relational database querying system. In Sojka, P., Kopecek, I., Pala, K., eds.: Proc. of TSD 2002. Volume 2448 of Lecture Notes in Computer Science., Brno, Czech Republic, Springer-Verlag (2002) 123–130

20. (Sylvia) `http://www.lllf.uam.es/proyectos/sylvia.html`.
21. Cedermark, P.: Swedish noun and adjective morphology in a natural language interface to databases. Master thesis, Uppsala University, Department of Linguistics (2003)
22. Reis, P., Matias, J., Mamede, N.: Edite: A natural language interface to databases – a new perspective for an old approach. In: Proc. of ENTER'97. Information and Communication Technologies in Tourism, Edinburgh, Springer-Verlag (1997) 317–326
23. Filipe, P.P., Mamede, N.J.: Databases and natural language interfaces. In Delgado, C., Marcos, E., Corral, J.M.M., eds.: Proc. of $5^{th}$ JISBD. Jornada de Engenharia de Software e Bases de Dados, Valladolid, Universidad de Valladolid (2000) 321–332
24. Meng, X., Wang, S.: Overview of a chinese natural language interface to databases: Nchiql. International Journal of Computer Processing of Oriental Languages **14** (2001) 213–232
25. Chae, J., Lee, S.: Frame-based decomposition method for korean natural language query processing. International Journal of Computer Processing of Oriental Languages **11** (1998) 213–232
26. Lee, H., Park, J.C.: Interpretation of natural language queries for relational database access with combinatory categorial grammar. International Journal of Computer Processing of Oriental Languages **15** (2002) 281–303
27. Barker, C., Dowty, D.R.: Non-verbal thematic proto-roles. In Schafer, A., ed.: Proc. of NELS 23 Conference. North-Eastern Linguistics Conferences, Amherst, Massachusetts, GLSA Publications (1992) 49–62
28. Chisarik, E., Payne, J.: Modelling possessor constructions in lfg: English and hungarian. In Butt, M., King, T., eds.: Proc. of the LFG01 Conference. International Lexical-Functional Grammar Conferences, Hongkong, Stanford CSLI Publications (2001) 49–62
29. Jensen, P.A., Vikner, C.: The english prenominal genitive and lexical semantics. Workshop on the Semantics/Syntax of Possessive Constructions, Amherst, Massachusetts (2002) Invited paper.
30. Storto, G.: Possessives in context. Possessives and Beyond: Semantics and Syntax, Amherst, Massachusetts, GLSA Publications (2004) 59–86
31. Tikk, D., Kardkovács, Z.T., Andriska, Z., Magyar, G., Babarczy, A., Szakadát, I.: Natural language question processing for hungarian deep web searcher. In Elmenreich, W., Haidinger, W., Machado, J.T., eds.: Proc. of ICCC 2004. IEEE Internation Conference on Computational Cybernetics, Wien, Austria (2004) 303–309
32. Katz, B., Felshin, S., Yuret, D., Ibrahim, A., Lin, J.J., Marton, G., McFarland, A.J., Temelkuran, B.: Omnibase: A uniform access to heterogeneous data for question answering. In Andersson, B., Bergholtz, M., Johannesson, P., eds.: Proc. of NLDB 2002. Volume 2553 of Lecture Notes in Computer Science., Stockholm, Springer-Verlag (2002) 230–234
33. Barker, C.: Possessive Descriptions. PhD thesis, University of Carolina, Santa Cruz, Department of Linguistics (1995)
34. Storto, G.: Possessives in Context – Issues in the Semantics of Possessive Constructions. PhD thesis, University of California, Los Angeles, Linguistics (2003)

# Binary Lexical Relations for Text Representation in Information Retrieval

Marco Gonzalez[1,2], Vera Lúcia Strube de Lima[1], and José Valdeni de Lima[2]

[1] PUCRS - Faculdade de Informática, Av. Ipiranga, 6681 – Prédio 16 - PPGCC
90619-900 Porto Alegre, Brazil
{gonzalez,vera}@inf.pucrs.br
[2] UFRGS – Instituto de Informática, Av. Bento Gonçalves, 9500
91501-970 Porto Alegre, Brazil
valdeni@inf.ufrgs.br

**Abstract.** Text representation is crucial for many natural language processing applications. This paper presents an approach to extraction of binary lexical relations (*BLR*) from Portuguese texts for representing phrasal cohesion mechanisms. We demonstrate how this automatic strategy may be incorporated to information retrieval systems. Our approach is compared to those using bigrams and noun phrases for text retrieval. *BLR* strategy is shown to improve on the best performance in an experimental information retrieval system.

## 1 Introduction

Many applications of natural language processing deal with text representation. Information retrieval (IR) is an example. The crucial question regarding indexing and searching in IR concerns how the query and document representation should be built to produce more effective retrieval. There are two research directions: unigram representation and term dependence representation.

Unigram representation follows a geometric (vector space retrieval model [16]) or mathematical (probabilistic retrieval model [18]) approach. Both approaches rely on the independence of the terms. The probabilistic model expresses the well known term independence assumption [19]. However, there are regularities provided by term dependences that need to be considered [11].

Term dependence representation is based on text cohesion features. Works that adopt this direction (e.g., [4, 12]) may apply mainly two approaches for term dependence extraction, with three different alternative models.

The main methods for term dependence extraction are based on statistical or syntactic approaches. In the former, term dependences are usually identified as statistical term co-occurrences (e.g., [17]). For a n-gram strategy, according to the statistical approach, the weight of a new term depends on the weight of the previous n-1 terms. In the syntactic approach, in addition to single terms, noun phrases (e.g., [10]), head-modifier relations (e.g., [12]), or other relationships may only be used to indicate (e.g., [8]) term dependences, or may be indeed considered as separate concept descriptors (e.g., [21]).

A. Montoyo et al. (Eds.): NLDB 2005, LNCS 3513, pp. 21–31, 2005.
© Springer-Verlag Berlin Heidelberg 2005

In this paper, we use the expression "relationship" for the representation of any complex concept which is constituted by two or more atomic concepts. Usually, a complex concept may be represented by bigrams, noun phrases, or other relationships. A noun phrase is a special case of these representations because it may represent a complex concept as well as an atomic concept in a term dependence model.

The alternative term dependence models investigated are: term-term (TT), term-relationship (TR), and relationship-term (RT) models. In the TT model (e.g., [15, 17, 20]), the weight of a term depends on the weight of another term if there is a relationship between them. Relationships are not considered as separate concept descriptors. In the TR model (e.g., [4, 8, 12, 21]), the relationship weight depends on the weights of their component terms, but also depends on their own presence in the document text. Relationships may be considered separate concept descriptors or not. In the RT model (e.g., [10, 24]), relationships are the main concept descriptors. Component terms are identified within these descriptors. The relationship weight may be used to evaluate the weights of the component terms, and these terms may not be used as separate concept descriptors.

Our model uses binary lexical relations (*BLR*s) as separate concept descriptors as well as terms. We combine TR and RT models. The evidence (weight) of a term depends on its presence in phrasal cohesion mechanisms [14] which are represented by *BLR*s. The *BLR* evidence depends on the evidence of its component terms. The purpose of this paper is to explain our model and to discuss its evaluation in an experimental IR system.

The rest of this paper is organized as follows: Section 2 presents works related to term dependence; Section 3 introduces the *BLR* definition; Section 4 explains our *BLR* extraction strategy; Section 5 presents our evidence-based weighting schema for terms and *BLR*s; Section 6 describes data and methods adopted to evaluate our model and shows experimental results; and Section 7 presents final considerations and future works.

## 2 Related Works

Related works are briefly discussed here concerning the term dependence models.

Fagan [2] analyses, in a TT model, both statistical and syntactic relationships representation. He points out advantages from the second approach, like, for instance, the capability of capturing the relationship between "parallel" and "algorithms" from "parallel and sequential algorithms".

The use of bigrams is a typical strategy in the TT model. Song and Croft [17] investigate how to incorporate term dependence by this strategy. Miller, Leek, and Schwartz [13] use a Hidden Markov Model to implement term dependence also through bigrams. Srikanth and Srihari [20] use biterms to represent term dependence. According to these authors, a biterm is similar to a bigram except that the constraint of order in terms is relaxed: biterms are unordered term-pairs. The weight of a biterm may be given by the Eq. (5) presented in Section 6. Biterm strategy is included in our experimental evaluation.

Some experiments discuss how to include linguistic knowledge in the text representation process using mainly the TR model. Changki Lee and Gary Lee [8] use a dependency parse tree with grammatical connections. Relationship between two nodes (parent and son) determines that the son node is dependent (or modifier) of the parent node. Matsumura, Takasu, and Adachi [12] create a structured index which is represented by a binary tree. A dependence relationship has one "relation word", which identifies the relationship, and two "concept words", which are the arguments of the relation. Weights of these relationships depend on their importance in a document part, and also on the weights of the component terms (concept words). Vilares, Barcala, and Alonso [21] present an approach for Spanish text indexing which uses an approximate grammar to conflate syntactic and morphosyntactic variants of a given multi-word term into a common base form. Gao, Nie, Wu, and Cao [4] present a method for capturing word dependences using a grammar and a graph. This structure limits the dependences to the most important relationships for retrieval.

In other works, noun phrases are used as main descriptors in RT model. Liu, Liu, Yu, and Meng [10] identify noun phrases in the query and search documents based on this identification. A document *d* has a query noun phrase *i* if all content terms of *i* are within a text window of a certain size in *d*. Zhai [24] uses a noun phrase parser for document indexing and combines three different kinds of descriptors from noun phrases: single terms, head-modifier relationships, and full noun phrases.

There are, according to TR or RT models, diverse methods proposed to identify term dependence. Bruza and Weide [1] specify relationships among terms using index expressions. These expressions are based on terms and connectors (prepositions and gerunds) in a structure named lithoid [23]. Katz and Lin [7, 9] use a system that extracts from text arbitrary textual patterns and relations. These relations (ternary expressions) may be viewed intuitively as subject-relation-object triples, or as typed binary relations from a syntactic point of view, or as two-place predicates from a semantic point of view. Kahane and Polguère [6] use lexical functions to represent collocations through oriented relations. A lexical function may denote a set of pairs of lexical items linked by lexical relations.

With respect to these works, our approach is not restricted to previously named identification term, and do not consider only collocations nor subject-relation-object triples. It generates a text representation through nominalized terms and *BLR*s motivated by phrasal cohesion mechanisms. The main contribution of our strategy is to combine TR and RT models by using a new evidence-based descriptor weighting.

## 3 Binary Lexical Relations

In our model, the text representation for indexing purposes derives from documents through a sequence of processing steps:

- tokenization: words and punctuations are detected;
- morphological tagging: morphological tags are assigned to each word and punctuation;
- lexical normalization: a nominalization process derives nouns from non stopwords of the text;

- binary lexical relation extraction: term dependences are extracted from phrasal cohesion mechanisms;
- descriptor weighting: a evidence-based calculation process evaluates the descriptor weights;

Through these steps, our strategy identifies two descriptor types:

- nominalized terms and
- binary lexical relations (*BLR*s).

We propose nominalization as an alternative for traditional lexical normalization processes, like stemming or lemmatizing. Nominalization is a transformation process which derives nouns from verbs, adjectives, and adverbs, like:

construct → constructor and construction

quick → quickness

easily → easiness

An automatic nominalization process was developed and integrated to our indexing strategy based on the fact that nouns are usually the most representative words of a document content [25].

Our nominalization tool reads an input word (a lemmatized verb, adjective, or adverb) and returns it in a concrete and/or an abstract noun form through finite automata. Each final state of the automata specifies nominalization operations in order to derive concrete and abstract nouns.

*BLRs* identify relationships between nominalized terms. These relationships capture phrasal cohesion mechanisms [14], like those which occur between subject and predicate, subject and object (direct or indirect), noun and adjective or verb and adverb. Some of those mechanisms reveal term dependences.

A *BLR* has the form *id*(*t*1,*t*2) where *id* is a relation identifier, and *t*1 and *t*2 are arguments (terms). There are three kinds of *BLRs*:

- classification: where *id* is the equal sign, *t*1 is a subclass or an instance of *t*2, and *t*2 is a class, e.g., =(rex,dog);
- restriction: where *id* is a preposition, *t*1 is a modifier and *t*2 is a head, e.g., of(quickness,decision); and
- association: where *id* is an event, *t*1 is a subject and *t*2 is a direct or indirect object, e.g., orientation(guide,tourist).

The examples above may respectively be derived from "rex is a dog", "they decide quickly" or "the quick decision", and "the guide orientates the tourist". Restriction *BLR*s are motivated by the mapping of syntactic dependencies onto semantic relations [3].

## 4 Extraction of *BLR*s

We developed a tool named RELLEX that automatically extracts *BLR*s from a Portuguese text with morphologic tags. Figure 1 shows our strategy adopted for *BLR* extraction.

Our *BLR* extraction strategy considers that a text is composed by sentences and each sentence is composed by clauses.

The sentence is the scope for the *BLR* extraction. A sentence has the following constituents: $C_1$ $C_2$ ... $C_n$ P, where $C_i$ is a clause i, and P is a punctuation mark, except comma, parentheses, and dash.

A clause has all or some the following constituents: *LS VS RS,* where:

- *LS* (left side) is constituted by a noun phrase (and/or relative pronouns or conjunctions);
- *RS* (right side) is constituted by a noun phrase, or by prepositional phrases, or by all of them; and
- *VS* (verbal set) is constituted by a verbal phrase without the right side.

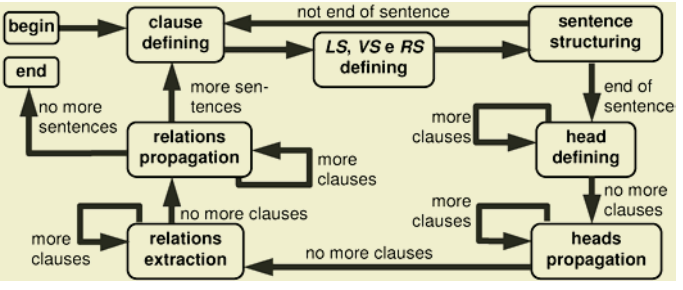



**Fig. 1.** *BLR* extraction strategy.

Each of these constituents may have a head. The heads are identified as follows:

- the *LS'* head is the noun phrase head;
- the *RS'* head is the direct object head, if it exists; and
- the *VS'* head is, in the following preferential order, the last main verb, or the last past participle or adjective.

There are two types of clause links: coordination and subordination (see Figure 2). They are useful in procedures of head and relation propagation which occur only in vertical direction, as shown in Figure 2.

Figure 2 shows the example of the structure of a sentence with three clauses. For instance, the sentence "The kind governess, which worked in the country house, and the butler escaped", has the following clauses:

- Clause 1 has only *LS* ("The kind governess").
- Clause 2 has three constituents: *LS* ("which"), *VS* ("worked"), and *RS* ("in the country house").
- Clause 3 has *LS* ("and the butler") and *VS* ("escaped").

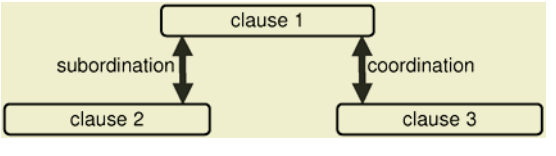



**Fig. 2.** Example of structure of a sentence with three clauses.

Relative pronouns and conjunctions provide clause links. They also make heads and relations propagation possible. In our example, "governess" is the *LS'* head of the clause 1, "worked" is the *VS'* head of the clause 2, "butler" is the *LS*' head of the clause 3, and "escaped" is the *VS'* head of the clause 3. The *LS'* head of the clause 1 is propagated (through the relative pronoun "which") to the *LS* of the clause 2. Therefore, "governess" is also the *LS'* head of the clause 2.

The *BLR*s "of(escape,*)", "by(escape,*)", and "=(*,fugitive)", for example, are propagated from clause 3 to clause 1. Then, "governess" and "butler" are valid values as the * argument of these *BLR*s.

The *BLR* arguments are original nouns of the text or nominalized terms. For example, in the *BLR* "=(butler,fugitive)", "butler" occurs in the text and "fugitive" is derived from "escaped" through nominalization.

**Table 1.** Some examples of *BLR* extraction rules. Notation: N = noun, AJ = adjective or participle, AV = adverb, PR = preposition, VB = verb, Vb = auxiliary verb, $(X)_C$ = concrete nominalization of X, and $(X)_A$ = abstract nominalization of X.

| *BLR* types | Extraction rules and examples |
|---|---|
| classification | N1 N2 → = ( N2, N1 ) |
| | the goalkeeper Dida → =(dida,goalkeeper) |
| | N1 'to be' N2 → = ( N1, N2 ) |
| | Dida is a goalkeeper → =(dida,goalkeeper) |
| | N VB → = ( N, $(VB)_C$ ) |
| | the citizen elected the ... → =(citizen,elector) |
| restriction | AJ N → of ( $(AJ)_A$, N ) ∨ of ( N, $(AJ)_C$ ) |
| | quick team → of(quickness,team) |
| | residential address → of(address,residence) |
| | N PR N → PR ( N, N ) |
| | lawyer with style → with(lawyer,style) |
| | AV VB → of ( $(AV)_A$, $(VB)_A$ ) ∨ of ( $(VB)_A$, $(AV)_C$ ) |
| | perfectly projected → of(perfection,project) |
| | mentally projected → of(project,mind) |
| association | N1 VB N2 → $(VB)_A$ ( N1, N2 ) |
| | the coach trained the athlete → training(coach,athlete) |
| | N1 Vb AJ 'by' N2 → $(AJ)_A$ ( N1, N2 ) |
| | the athlete was trained by the coach → training(coach,athlete) |
| | N1 VB PR N2 → Na(VB).PR ( N1, N2 ) |
| | the tourist traveled across Europe → travel.across(tourist,europe) |

Our approach for *BLR* extraction is based on rules. Some simplified rules for *BLR*s extraction and examples are presented in Table 1. For Portuguese language, some rules may accept word's inversions. For example, the first rule for restriction needs an alternative form "N AJ → …".

The distinction between abstract and concrete nominalizations is crucial to the correct identification of *BLR*s, and especially to the argument positions. For instance, see the first and the last examples for restriction *BLR*s in Table 1.

## 5 Descriptor Weighting

We use Okapi BM25 formula [19], according to the IR probabilistic model, for the weight ($W_{i,d}$) associated with a descriptor $i$ in a document $d$. This weight is given by:

$$W_{i,d} = \frac{evd_{i,d}\,(k_1 + 1)}{k_1((1-b) + b\,\frac{DL_d}{AVDL}) + evd_{i,d}} \log \frac{N}{df_i} \tag{1}$$

where $k_1$ and $b$ are parameters whose usual values are 1.2 and 0.75 respectively; $DL_d$ is the length of $d$ and $AVDL$ is the average document length in the collection; $N$ is the number of documents in the collection and $df_t$ is the number of documents where $t$ occurs.

Usually

$$evd_{i,d} = f_{i,d} \tag{2}$$

where $f_{i,d}$ is the frequency of the descriptor $i$ in a document $d$.

But in our evidence-based approach, *BLRs* are used to evaluate the evidence ($evd_{t,d}$) of a term $t$ in a document $d$:

$$evd_{t,d} = \frac{f_{t,d}}{2} + \sum_r f_{r,t,d} \tag{3}$$

where $f_{t,d}$ is the frequency of the term $t$ in a document $d$; and $f_{r,t,d}$ is the number of *BLRs* (in the document $d$) where $t$ is an argument.

For a *BLR* $r$, the evidence ($evd_{r,d}$) is:

$$evd_{r,d} = f_{r,d}\,(evd_{t1,d} + evd_{t2,d}) \tag{4}$$

where $f_{r,d}$ is the frequency of $r$ in document $d$, and $r$ is between terms $t1$ and $t2$, with their respective evidences ($evd_{t1,d}$ and $evd_{t2,d}$).

Eq. (4) is based on the following justification. If there is a *BLR* $r$ between terms $t1$ and $t2$ in a document $d$, there are 3 distributed evidence units concerning the 3 concepts involved, represented by $t1$, $t2$, and $r$. If a term of $r$ participates in other phrasal cohesion mechanism, this fact adds 1 evidence unit to the compound concept relating to $r$ in the context where it occurs, the document $d$. Eq. (3) is formulated to support this justification.

## 6 Experimental Evaluation

### 6.1 Data and Methods

The main purposes of this work are to present our *BLR* approach and to analyze the contribution of *BLR*s in results of an experimental IR system. With this second purpose, we have used the document collection Folha94 (adapted from [5]) constituted by 4,156 articles extracted from 229 editions from Folha de São Paulo newspaper of the year 1994.

Folha94 has 1,235,291 lexical items (words and punctuations): 62,077 adjectives, 47,848 adverbs, 334,682 nouns, 23,440 participles, 86,197 verbs, and 681,047 occur-

rences of punctuations and stopwords. To comparatively evaluate the examined approaches, we have used 50 topics for generation of test queries, following the strategy in use by the Text Retrieval Conferences [22].

We have examined three automatic indexing approaches in probabilistic IR model: BT, NP, and NOMBLR. BT uses biterms according to Srikanth and Srihari's work [20], NP uses noun phrases, and NOMBLR uses nominalized terms and *BLR* as descriptors.

Table 2 shows the size of the inverted files, the amount of descriptors (terms and relationships), and the time (in a 866 MHz Pentium III machine) spent for indexing and searching in each approach. Indexing times consider the time averages to index a document with 1,000 lexical items (words and punctuations). Searching times consider the time averages to process a query with 2 terms.

Note that, in Table 2, BT relationships are biterms, but 2.9% of these descriptors are single terms which occur alone in the text. These terms do not constitute relationships, but they are included in the same inverted file for relationships. The same happens in the case of NP: 13.3% of the noun phrases have only one term. Only NOMBLR has a specific inverted file for single terms.

**Table 2.** Memory space, terms, and processing time.

| indexing approaches | inverted files (Kb) | terms | relationships | indexing time (s) | searching time (s) |
|---|---|---|---|---|---|
| NOMBLR | 10,542 | 38,616 | 245,093 | 0.579 | 0.178 |
| BT | 8,938 | 0 | 296,243 | 0.155 | 0.437 |
| NP | 4,453 | 0 | 112,239 | 0.107 | 0.214 |

All of the searching strategies use Eq. (1) except that concerning BT. In this strategy the weight of a biterm $\{t1,t2\}$ is given by:

$$W_{\{t1t2\},d} = \alpha_1 \frac{f_{(t1t2),d} + f_{(t2t1),d}}{2\min\{f_{t1,d}, f_{t2,d}\}} + (1-\alpha_1)\left(\alpha_2 p_{t2,d} + (1-\alpha_2) p_{t2,C}\right) \quad (5)$$

where $(titj)$ is a bigram with terms $ti$ and $tj$; $f_{i,d}$ is the frequency of the term (or bigram) i in a document $d$; $\min\{f_{t1,d}, f_{t2,d}\}$ is the minimum of the frequencies of terms $t1$ and $t2$; $p_{t2,d} = f_{t2,d}$ / total of terms of $d$; $p_{t2,C} = f_{t2,C}$ / total of terms of the document collection; $f_{t2,C}$ is the $t2$ frequency in the document collection; and $\alpha_1$ and $\alpha_2$ are parameters whose values are 0.1 and 0.4 respectively in Srikanth and Srihari's work [20].

The NOMBLR1 and NP1 searching strategies consider only *BLR*s or noun phrases respectively. NOMBLR2 and NP2 use also single component terms. Note that, in both NP1 and NP2, a query noun phrase may be within a largest noun phrase in the document. NOMBLR1 and NOMBLR2 use Eqs. (3) and (4), while NP1 and NP2 use Eq. (2).

### 6.2 Results

Figure 3 presents the recall-precision curves for the three approaches examined here. It is possible to see that NOMBLR2 approach has the higher values for precision and

recall. The comparative analysis also indicates, in this experiment, the superiority of BT and NP2 over NOMBLR1 and NP1, in recall values.

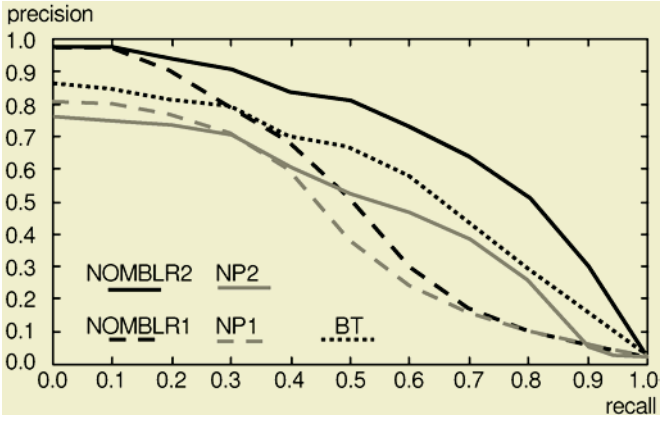



**Fig. 3.** Recall-precision curves.

Table 3 presents precision, recall, and F measure values for each approach after some specific ranking positions of the retrieved documents. The superiority of the NOMBLR2 strategy over all other ones is also observable here. The strategies which use both terms and relationships (NOMBLR2 and NP2) have higher values for all measures when compared to strategies which use only terms (NOMBLR1 and NP1), except the precision values of NOMBLR1 which are higher than NP2.

**Table 3.** Some values of precision, recall, and F measure. Notation: "at N" means that the value is for the first N documents in the ranking.

| searching | precision | | recall | | F |
|---|---|---|---|---|---|
| approaches | at 1 | at 10 | at 10 | at 100 | at 10 |
| NOMBLR2 | 0.980 | 0.716 | 0.646 | 0.952 | 0.679 |
| BT | 0.860 | 0.612 | 0.560 | 0.912 | 0.585 |
| NP2 | 0,760 | 0,548 | 0,499 | 0.869 | 0.522 |
| NOMBLR1 | 0,980 | 0.560 | 0.483 | 0.653 | 0.519 |
| NP1 | 0.800 | 0.506 | 0.448 | 0.579 | 0.475 |

## 7 Conclusion and Future Work

With respect to the amount of descriptors and inverted file size (see Table 2), the greatest economy belongs to the NP approach. The differences in processing time between NOMBLR and the other approaches are mainly due to the fact that BT and NP consider only bigrams and noun phrases respectively. The indexing time for NOMBLR is longer than the time spent by the other approaches because it needs to construct a more complex indexing structure. However, this structure improves on the best performance in searching phase.

A bigram (or biterm) strategy may fail because while some of the term dependences occur between adjacent terms, others are more distant [4]. The use of noun phrases, as a unique concept descriptor, means to discard verbs and their modifiers in text representation. Our strategy with nominalized terms and *BLR*s brings the following contributions for concept description:

- Extraction of the same representation for the same concept originated from distinct syntactic forms. The use of *BLR*s makes IR searching strategies independent from the specific query form.
  Example: "by(impediment,law)" is the unique *BLR* extracted from "... impeded by law" or from "the law impeded ...".
- Generation of distinct representations for distinct concepts' compositions, even when they are represented through the same terms. The use of *BLR*s invalidates the matching of improper term dependences in IR systems.
  Example: "engine in silent process" and "process in silent engine" are represented (concerning to the adjective) by distinct *BLR*s respectively: "of(silence,process)" and "of(silence,engine)".
- Use of prepositions which are not considered stopwords merely.
  Example: "the train from Madrid" and "the train to Madrid" have different representations: "from(train,madrid)" and "to(train,madrid)" respectively.
- Use of nominalization as a normalization process.

There are interesting future research directions based on this work. *BLR*s may be used to produce a graph where vertices are *BLR* arguments. In this structure, the evidence of both terms and relations plays an important role for text representation. Such structure may be useful for applications like text categorization and summarization. In IR, this graph assumes both index and thesaurus characteristics. A domain thesaurus, with classification, restriction and association links, can make possible automatic query expansion by including expanded terms associated with original ones through *BLR*s.

## References

1. Bruza, P. D.; van der Weide, Th. P. The Modeling and Retrieval of Documents using Index Expressions. ACM SIGIR Forum, Vol .25, N.2 (1991) 91-103.
2. Fagan, J. L. Automatic Phrase Indexing for Document Retrieval: An Examination of Syntactic and Non-Syntactic Methods. In Proceedings of 10th Annual International ACM SIGIR conference (1987) 91-101
3. Gamallo, Pablo; Gonzalez, M.; Agustini, A.; Lopes, G; Lima, Vera L. S. Mapping Syntactic Dependencies onto Semantic Relations. ECAI'02, Workshop on Natural Language Processing and Machine Learning for Ontology Engineering, Lyon, France (2002) 15-22
4. Gao, J.; Nie, J.; Wu, G.; Cao, G. Dependence language model for information retrieval. In Proceedings of 27th Annual International ACM SIGIR conference (2004) 170-177
5. http://www.nilc.icmc.usp.br/lacioweb
6. Kahane, Sylvain; Polguere, Alain. Formal Foundation of Lexical Functions. ACL'2000 – Workshop on Collocation, Toulouse (2001)

7. Katz, Boris; Lin, Jimmy. REXTOR: A System for Generating Relations from Natural Language. ACL'2000 – Workshop on Recent Advances in NLP and IR, Hong-Kong, University of Science and Technology (2000)
8. Lee, C.; Lee, G. G. Probabilistic information retrieval model for a dependency structured indexing system. Information. Processing and Management, Vol. 41 (2005) 161-175. Available online 19 December 2003.
9. Lin, Jimmy. Indexing and Retrieving Natural Language using Ternary Expressions. Master thesis, Massachusetts Institute of Technology, Cambridge (2001)
10. 10.Liu, S.; Liu, F.; Yu, C.; Meng, W. An effective approach to document retrieval via utilizing WordNet and recognizing phrases. In Proceedings of 27th Annual International ACM SIGIR conference (2004) 266-272
11. Losee, R. M. Term Dependence: a basis for Luhn and Zipf Models. Journal of the American Society for Information Science, Vol. 52, N. 12 (2001) 1019-1025.
12. Matsumura, A.; Takasu, A.; Adachi, J. The Effect of Information Retrieval Method Using Dependency Relationship Between Words. RIAO – Multimedia Information Representation and Retrieval (2000)
13. Miller, D. H., Leek, T.; Schwartz, R. 1999. A Hidden Markov Model information retrieval system. In Proceedings of 22th Annual International ACM SIGIR conference (1999) 214-221
14. Mira Mateus, M.H.; Brito, A. M.; Duarte, I.; Faria, I. H. Gramática da Língua Portuguesa. Lisboa: Ed. Caminho (2003)
15. Nallapati, R.; Allan, J. Capturing term dependencies using a language model based on sentence trees. In Proceedings of the 11th International Conference on Information and Knowledge Management, CIKM (2002) 383-390
16. Salton, G.; Buckley, C. Term-weighting approaches in automatic text retrieval. Information Processing and Management, Vol. 24 (1988) 513-523
17. Song, F.; Croft, B. A general language model for information retrieval. In CIKM (1999) 316–321
18. Sparck-Jones, K. Search Term relevance weighting given little relevance information. Journal of Documentation, Vol. 35 (1979) 30-48
19. Spark-Jones, K.; Walker, S.; Robertson, S. E. A Probabilistic Model of Information Retrieval: Development and Comparative Experiments – Part 1 and 2. Information Processing and Management, Vol. 36, N. 6 (2000) 779-840
20. Srikanth, M.; Srihari, R. 2002. Biterm language models for document retrieval. In Proceedings of 25th Annual International ACM SIGIR conference (2002) 425-426
21. Vilares, J., Barcala, F. M.; Alonso, M. A. Using Syntactic dependency-pairs conflation to improve retrieval performance in Spanish. In Computational Linguistics and Intelligent Text Processing, Springer-Verlag, Lectures Notes in Computer Science (2002)
22. Voorhees, E. M. Overview of TREC 2003. NIST Special Publication – SP500-255. The 12th Text Retrieval Conference, Gaithersburg (2003)
23. Wondergem, B.; van Bommel, P.; Weide, Th. P. Nesting and Defoliation of Index Expressions for Information Retrieval. Knowledge and Information Systems, Vol. 2, N. 1 (2000)
24. Zhai, C. Fast statistical parsing of noun phrases of document indexing. In Proceedings of the fifth conference on Applied natural language processing (1997) 312-319
25. Ziviani, N. Text Operations. In Baeza-Yates, R.; Ribeiro-Neto, B. Modern Information Retrieval. New York : ACM Press (1999)

# Application of Text Categorization to Astronomy Field

Huaizhong Kou*, Amedeo Napoli, and Yannick Toussaint

LORIA and INRIA-Lorraine
615, rue du jardin botanique, 54603 Villers-lès-Nancy, France
{huaizhong.kou,amedeo.napoli,yannick.toussaint}@loria.fr

**Abstract.** We introduce the application of text categorization techniques to the astronomy field to work out semantic ambiguities between table column's names. In the astronomy field, astronomers often assign different names to table columns at their will even if they are about the same attributes of sky objects. As a result, it produces a big problem for data analysis over different tables. To solve this problem, the standard vocabulary called "unified concept descriptors (UCD)" has been defined. The reported data about sky objects can be easily analyzed through assigning columns to the predefined UCDs. In this paper, the widely used Rocchio categorization algorithm is implemented to assign UCD. An algorithm is realized to extract domain-specific semantics for text indexing while the traditional cosine-based category score model is extended by combining domain knowledge. The experiments show that Rocchio algorithm together with the proposed category score model performs well.

## 1 Introduction

Text Categorization (TC) is the procedure of assigning one or multiple predefined domain-specific category labels to a free text document (category sometimes called "topic" or "theme"). Text categorization technologies have been widely employed to cope with various tasks that are based on the analysis of text content, such as categorical organization of news at Yahoo site.

In the astronomy research field, the volume of astronomy articles available in electronic forms grows increasingly over time and most of them contain one or more tables of observed data about sky objects, which consist of different columns. The tables contain observation data about various attributes of sky objects, such as temperature, luminary intensity, speed, position, rotation angle and so on. Also, they contain some data of astronomical instruments used to make observation. One column of a data table is about some attribute of sky objects. Actually, since there is not any standard about how to name table columns using a standard astronomy vocabulary, different astronomers often assign different names to the table columns of the same attributes of sky objects and consequently the semantics of data of table column are not clear. For example, there are 73 different column names in 3571 tables corresponding to the "Right Ascension"[1] attribute of observed sky objects. As a result, it is very hard to analyze and compare the data reported by different astronomers and many existing data mining technologies cannot be directly applied to discover astronomical knowledge. The ambiguity of table column names, which also covers semantics of column data, is one of the big problems for reusing and sharing of observed sky data.

---

* He is now with the Yellow River Conservancy Commission, China

[1] http://cdsweb.u-strasbg.fr/UCD/cgi-bin/ucd_stats?leaf=POS_EQ_RA_MAIN

A. Montoyo et al. (Eds.): NLDB 2005, LNCS 3513, pp. 32–43, 2005.
© Springer-Verlag Berlin Heidelberg 2005

To solve such problems and ease both reusing and sharing of the observed data, one has to normalize the semantics of data of table columns reported by different astronomy researchers. One solution is that one semantically structured list of standard concepts is firstly constructed to represent the attributes of sky objects and then the columns are associated with the standard concepts. For this, one hierarchy of relatively standard concepts called Unified Content Descriptors (UCD) 1 has been already defined at Strasbourg astronomical Data Center (CDS) 1.

The UCDs can provide unified and unambiguous descriptions about the attributes of sky objects. The semantics of data of table column become explicit through establishing map relationship from table columns to the UCD concepts. For example, "POS_EQ_RA _MAIN" is an UCD, which represents the "Right Ascension" attribute of sky objects. When the 73 different column names mentioned above are assigned to it, the semantics of these 73 columns are clear. The UCD assignment is aimed to associate table's columns to the predefined UCDs with the goal that data about sky objects reported by different astronomers can be easily shared and analyzed. For this, one system of UCD assignment has been manually built and is used to assign UCD to table columns at CDS 1. As yet almost all of the $10^5$ columns of the catalogues contained in VizieR have been associated with UCDs 1.

To support the UCD assignment, the *Readme* text files with specified format 6 are provided by astronomers when they upload their observation data documents. Among other things, the *Readme* files contain detail information about semantic of each column of table. Notice that there are already more than $10^5$ columns associated with UCDs. These motivate us to test standard text categorization algorithms with the hope that UCD assignment system could be built automatically and the UCD assignment could be probably improved by text categorization technologies, which is based on analysis of the contents of column's explanation texts. To do this, we have adapted Rocchio algorithm.

The contributions of this paper include: the application of standard text categorization technologies to UCD assignment in the astronomy field is implemented, an algorithm of extracting domain-specific semantics is developed and a model of calculating category score, which can increase performance by 6.7% according to Rocchio algorithm, is also defined; the obtained results can provide support for the definition of UCD ontology that is ongoing parallel work in the frame of ACI-MDA project[2].

The rest of this paper is organized as follows. Section 2 is about UCD assignment and related information, then Rocchio algorithm is presented in Section 3 and a category score model is also proposed. Section 4 describes semantic enrichment for text index. The results and analysis of our experiments are reported in Section 5 while Section 6 concludes this paper by indicating some future work.

## 2 UCD Assignment

### 2.1 Objective of UCD Assignment

Figure 2.1 illustrates the scenario of UCD assignment. The left part shows that often there are observation data tables along with the articles published by astronomers. The

[2] http://cdsweb.u-strasbg.fr/MDA/mda_en.html

authors of articles name the columns of data tables at their will. Since there does not exist any standard in the astronomy field about naming the columns of observation data tables, different names are often given to the columns about the same attributes of sky objects and at the same time same names maybe are linked to different columns about different attributes of sky objects. For example, there are 73 different column names for the “right ascension” attribute, including “RA”, ”Rao”, ”RAL”, ”RAG”, ”RA1984”, ”RAX”, ”RAK”, and so on. Such as, automatic analysis over different data tables is almost impossible.

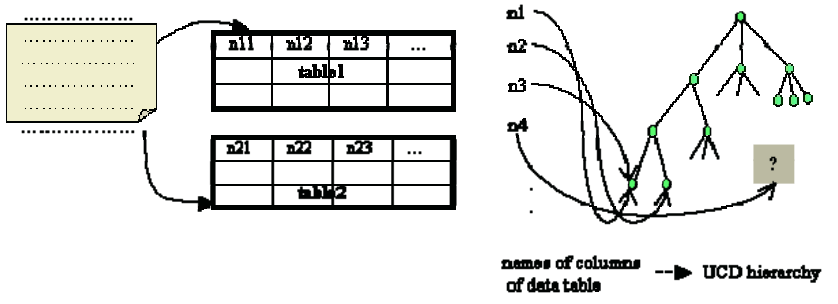



**Fig. 2.1.** Data table and UCDs.

The proposal of UCD by CDS is a great step toward standardizing the names of table columns and easing automatic data analysis. All UCDs together constitute one hierarchy tree of standard concepts in the astronomy field for naming table columns, like shown by the right part of Figure 2.1. The objective of UCD assignment is to match columns of data tables to a concept node in the UCD tree. The UCD assignment can be performed by analyzing the *ReadMe* file.

On the other hand, from the point of view of data integration, if we take the structures of data tables independently defined by different astronomers as local schemas of observation data and UCD hierarchy as global schemas accordingly, the problem of assignment of UCDs to table columns is just like one of mapping local data schema to one global schema, which is essential step toward transparent use of data. Once local schemas are mapped to the global schema, many analysis across data sources can be performed.

In our case, schema mapping is aimed to establish correspondences for table’s columns to UCDs in the UCD hierarchy. In the practice, description matching is one of approaches to schema mapping. By description matching, comment texts in natural language about the semantics of schema elements will be linguistically evaluated to map the elements of local schemas into the global schema. In the context of UCD assignment, the *ReadMe* files provide such comment texts about the semantics of table structures, see Section 2.3.

## 2.2 Schema of UCD

The UCD schema is a 4-level hierarchy tree that is firstly presented in 5. It contains 1417 nodes, including 1380 leaf nodes and 37 non-leaf nodes. Actually only 1183 UCDs are used in *VizieR*.

UCD is defined by the pairs (name, definition), where the name is identification of UCD used in *VizieR* and the definition defines semantic meaning of UCD. For example, the followings are three UCDs[3]:

- AT_COLL Atomic Collisional Quantities
- AT_COLL_EXCIT-RATE Collisional Excitation Rate
- AT_COLL_STRENGTH Collisional strength

AT_COLL is an internal node for atomic collision quantities while the second and the third UCD are for two different quantities of atomic collision, excitation rate and strength respectively.

UCD hierarchy tree provides an unified schema of concepts at the global level. On the top of such standard schema various correlation analysis of observation data parameters supplied by different astronomers can be performed. See 1 for the entire UCD tree.

### 2.3 ReadMe

Given a catalogue, its *ReadMe* 6 is a text file that contains all necessary information to interpreter and locate the contents of catalogues. It is composed of many sections, such as abstract, keywords, notes and etc.. Among them, the section *Description of data table* is most important for assigning UCDs.

The section *Description of data table* consists of many rows, each of which describes the semantics of one data table column in five fields: bytes, format, unit, label and explanation. The functions of the five fields are as follows:

- Bytes: the position of column data.
- Format: data format of column content.
- Unit: used for the data of column content.
- Label: column name.
- Explanation: a short text to explain the semantic information of column content.

We cite some rows of this section[4] as follows:

| | Bytes | format | unit | label | explanation |
|---|---|---|---|---|---|
| • | 31-32 | I2 | arcmin | DEm | Declination J2000 (minutes) |
| • | 1- 12 | A12 | --- | MACS | Designation |
| • | 14- 15 | I2 | h | RAh | Right Ascension J2000 , Epoch 1989.0 (hours) |
| • | 17- 18 | I2 | min | RAm | Right Ascension J2000 (minutes) |
| • | 20- 25 | F6.3 | s | RAs | Right Ascension J2000 (seconds) |
| • | 27 | A1 | --- | DE- | Declination J2000 (sign) |

Take the first row as an example. “31-32” is the position of column’s data in the data table file; “I2” is format of column’s data and it means two integers; “arcmin” is the unit that is associated with the table column named as “DEm”; the explanation “Declination J2000(minutes)” describes the semantic of the columns. Three fields: unit, label and explanation are actually used to assign UCDs to table columns 5. See 6 for detail of *ReadMe* file.

---

[3] http://cdsweb.u-strasbg.fr/viz-bin/UCDs

[4] http://vizier.u-strasbg.fr/doc/catstd-3.1.htx

## 2.4 Frame of UCD Assignment

Figure 2.2 shows the frame of UCD assignment. At large, it consists of two parts. The first one that is in dotted-line rectangle is aimed to learn the text classifier; the second one is to assign table's columns to UCDs. The corpus for learning is made up of example columns with their explanations contained in the corresponding *ReadMe* files and the UCDs that are already assigned to the example columns in the VizieR system at CDS. Learning the text classifier is a supervised process, which includes selection of vocabularies, text index, estimation of text classifier's parameters and etc..

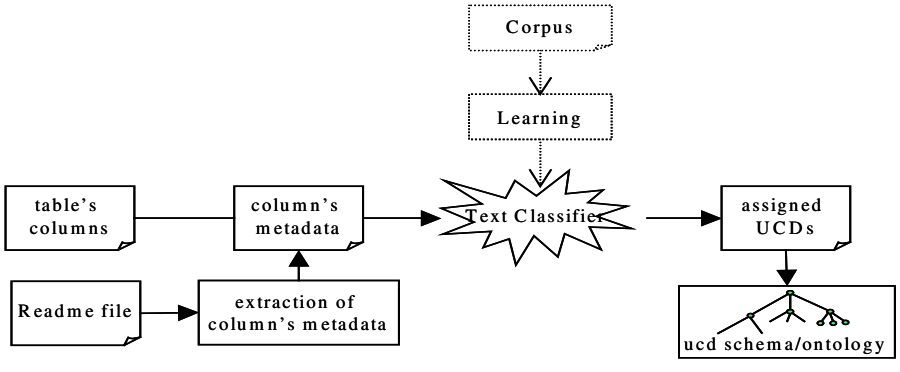



**Fig. 2.2.** Frame of UCD assignment.

Once the text classifier is built, it can be used to assign table's columns to UCDs. To assign table's columns to UCDs, the column's metadata are firstly extracted from the corresponding *ReadMe* file. Then they are fed to the learned text classifier. The text classifier will return one or more UCDs to the table's columns on analyzing the text contents of the metadata.

## 2.5 Related Work

At CDS, one system of UCD assignment has been manually built mainly using units, column's labels and explanations. As for any manually building system there is a very common agreement. That is, it is time consuming, heavily dependent of limit of knowledge of domain experts and it is very difficult to update the system over time. The CDS's system is certainly not an exception.

More than $10^5$ columns have been assigned to UCDs by CDS's system until now. For a given table column described by the metadata section of *ReadMe* file, UCD assignment is performed in two steps: firstly several most possible UCDs are returned by CDS's system on calculating relevance score for every UCD; then astronomers can select one UCD from the returned UCDs to assign it to the column. If neither of returned UCDs is suitable, astronomers have to choose one UCD from the UCD schema by themselves for the column. For this, sometimes astronomers have to study the content of article to finally decide one UCD. See 1 for more information about how to use CDS's assignment system.

# 3 Text Categorization Algorithm

## 3.1 Rocchio Algorithm

Rocchio algorithm is the relevance feedback-based optimal query creation algorithm, which has been successfully used in IR as of its early days. It has been adapted to categorize text 3. Being applied to document categorization, Rocchio firstly builds one "centroid vector", sometimes called "conceptual vector", for each category $c_i$ (i=1, 2,..., m) averaging the vectors of documents assigned to the category with the formula (3.2).

$$\vec{c}_i = \frac{1}{|c_i|}\sum_{d\in c_i}\vec{d} \tag{3.2}$$

$$simil(c_i, d) = \cos(\vec{c}_i, \vec{d}) \tag{3.3}$$

To categorize a given document $d$, the similarities between the vector of document $d$ and the centroid vectors of all category $c_i$ (i=1, 2,..., m) are calculated using cosine function-based similarity model (3.3). Such similarities are used as category scores. Then all categories are ranked in the decreasing order of category scores, and so a ranked list of categories is obtained. Finally, some strategy is taken to decide the categories to which the document d is assigned. Rocchio algorithm assumes that the centroid vector of category can present the concept model of category. It performances well if the centroid vectors can characterize the concepts contained in categories.

## 3.2 Model of Calculating Category Score

In our implementation of UCD assignment system, three factors are combined to calculate category score in the following formula (3.4).

$$\begin{aligned} score\ (\ column, UCD\ C) &= unitRevelance(unit, unitList, \beta) \times \\ &(\ \alpha \times Simil(vector\ of\ C\ definition, column's\ explanation\ vector) + \\ &(\ 1-\alpha) \times ScoreByAlgo(C, column's\ explanation\ vector)) \end{aligned} \tag{3.4}$$

$$unitRelevance(unit, unitList, \beta) = \begin{cases} 1, & if\ unit\ is\ in\ unitList \\ \beta, & if\ unit\ is\ not\ in\ unitList \end{cases}$$

where $\alpha \in [0,1]$ and $\beta \in [0,1]$, *ScoreByAlgo*(. , .) is the category score calculated by (3.3), *Simil*(vector of *C* definition, column's explanation vector) is calculated using cosine function and *unitRelevance*(.,.) measures the impact of units on relevance of UCD and column. The model in (3.4) is equal to *ScoreByAlgo*(. , .) if $\alpha$ is set to 0 and $\beta$ to 1.

In the case of UCD assignment, some UCDs are very similar and the explanation texts of columns belonging to them share very much terms. On the other hand, the definition texts of UCDs often contain significant terms for assigning UCD. Due to this fact, we introduce the similarity between UCD's definition and column's explanation text by *Simil*(vector of *C* definition, column's explanation vector). We hope that it can help to identify similar UCDs.

The introduction of the function *unitRelevance*(.,.) is based on the assumption that one UCD should be punished by reducing its category score if the list of accepted units of the UCD does not contain the unit of the column to be categorized. In other words, it takes 1 as its value if column's unit is in the list of accepted units of the UCD and $\beta \in [0,1[$ as its value otherwise. Here we assume that a relatively complete list of units used by table columns for a given UCD can be obtained if the corpus covers many enough cases of different columns. Setting $\beta$ as 0 means that a column is certainly not assigned to an UCD if the unit of the column is not in the list of accepted units of the UCD while $\beta$ with value as 1 means that units are not taken into account in calculating category score.

# 4 Semantic Enrichment for Index

## 4.1 Simple Words and Semantic Enrichment

Text index mainly consists of first identifying index words from texts of documents and of then measuring the importance of index word to represent the content of document. In the practice of traditional IR and text categorization, simple word-based text index is often used and documents are taken as a bag of words without distinguishing semantic information of words. Intuitively simple word-based index technologies have some limitations 7, including such as synonym, polysemy, local context and so on.

Assumed that more semantic information is used and higher precision performance can be achieved, some works on semantic enrichment-based index have been reported by different researchers 72. For example, context-specific phrases are extracted using information extraction technologies in 7, furthermore they are applied to text index and high precision of text categorization is achieved. Domain-specific concepts are extracted using probabilistic latent semantic analysis 2 and these domain-specific concepts are used to supplement simple index words. Their experimental results have shown that text categorization performance has been improved.

## 4.2 Astronomy Domain-Specific Semantic Enrichment

We also strongly believe that domain-specific semantic information can improve text categorization performance. It seems that semantic enrichment is more important in the case of UCD assignment mainly because the document texts are very short ( only about 4 words ) and information contained in document text is relatively poor. And often simple word-based index cannot discriminate certain close topics, for which semantic enrichment is indeed essential. For example, the following four UCDs are very similar:

- PHOT_FLUX_IR_12 : Flux density (IRAS) at 12 microns, or around 12 microns (ISO at 14.3)
- PHOT_FLUX_IR_25 : Flux density (IRAS) at 25 microns
- PHOT_FLUX_IR_60 : Flux density (IRAS) at 60 microns
- PHOT_FLUX_IR_100 : Flux density (IRAS) at 100 microns

**Table 4.1.** Example of UCD and texts of column's explanations.

| UCD | Text of column's explanations |
|---|---|
| PHOT_FLUX_IR_12 | Flux density at 12 micron |
| PHOT_FLUX_IR_12 | [0,] IRAS flux density at 12micron |
| PHOT_FLUX_IR_12 | Estimated IRAS 12 micron flux |
| PHOT_FLUX_IR_25 | IRAS flux at 25 micron |
| PHOT_FLUX_IR_25 | 25 micron IRAS flux |
| PHOT_FLUX_IR_100 | Flux density at 100 micron |
| PHOT_FLUX_IR_100 | IRAS flux density at 100micron |

```
1. Input : doc_tokens
2. String  regX1 = "[0-9]*[\\.]?[0-9]*\\s*[\\-]?\\s*[0-
   9]*[\\.]?[0-9]+";
3. String  regX2 = "[0-9]*[\\.]?[0-9]*\\s*[\\-]?\\s*[0-
   9]*[\\.]?[0-9]+\\s*[a-zA-Z]*";
4. List indexTerms;
5. while (hasMoreTokens){
6.    String word  =  nextToken();
7.    indexTerms.add(word);
8.    if (matches(regX1, word)){
9.       String unit = nextToken();
10.      if (isValidUnit(unit)){
11.          indexTerms.add(word+unit);
12.      }
13.     indexTerms.add(unit);
14.   } else if (matches(regX2,word)){
15.     String  unit  =  extractUnit(word);
16.     if (unit!=null)){
17.       indexTerms.add(unit);
18.     }
19.  }
20.}
21.return indexTerms;
```

**Fig. 4.1.** Domain-specific information extraction algorithm.

The column's explanation texts labeled with them often share much of common simple words like shown by Table 4.1. If here the number "12", "25" and "100" are individually treated, they are just same as mathematic number as themselves. As a result, the local context semantic information associated with them is certainly lost and this may mislead UCD assignment.

One of solutions to such problems is to combine these numbers with special words immediately following them (such as "micron" in our example) and enrich index semantic information. We will use, for example, "12", "micron" and "12micron" instead of only "12" and "micron" as index terms to represent the explanation "Flux density at 12 micron".

In our system, we extract two types of piece of information to enrich index semantic information at present:

a) Domain-specific patterns will be identified from explanation's texts. For example, "12 micron" will be extracted from the text "Flux density at 12 micron" and "2-10kev" from "The 2-10 keV count rate (2)".

b) Domain-specific simple words will be identified from some composite words. For example, “micron” will be identified from the text “[0,] IRAS flux density at 12micron”. This allows represent the relationship between “micron” and the text using both “micron” and “12micron” instead of only “12micron”.

Our information extraction algorithm in Figure 4.1 actually is knowledge-based because an astronomy domain-specific unit dictionary is used to guide extraction of both domain-specific patterns and domain-specific simple words. During the preprocessing of text, the extraction algorithm will be activated each time that predefined trigger pattern is encountered in order to identify index words from texts of documents. The trigger patterns are defined using regular expressions for every type of piece of information to be extracted.

The input *doc_tokens* is tokenized document text with space as delimiter, regX1 and regX2 are two trigger patterns respectively corresponding to the types a) and b). If the current *word* matches regX1 (line 8), the method *isValidUnit*() will be triggered to check if the following word called *unit* is valid unit ( line 9-11 ); if it matches regX2 ( line 14 ), the method *extractUnit*() will be triggered to try to extract possible *unit* contained in *word* ( line 15-18 ). Both *isValidUnit*() and *extractUnit*() are based on the astronomy unit dictionary. The extracted words will be added into index term list to supplement simple words rather than substitute them.

## 5 Experiment Results and Discussions

### 5.1 Design of Experiment

We notice that the numbers of columns assigned to each UCD are very different. For example, 2481 columns are assigned to the UCD “ ERROR” but only 1 column is assigned to many UCDs. Among 874 UCDs, some moderate UCDs whose frequencies are between 30 and 100 inclusive are selected and the columns that are assigned to these moderate UCDs are picked up to make up the corpus. Finally, our corpus includes 93 UCDs and 4904 columns: 3371 for training and 1533 for test. Two approaches to text indexing are implemented. The first approach to index is trivial, shown as follows:

- Text documents are tokenized into individual words.
- 318 English stop words are removed, such as “the”, “of”, “on” and so on.
- Non-alphabetic characters are removed with the except that “-” and “_” are kept.
- Variants of words are transformed to their dictionary original form with the help of WordNet 2.0[5]. For example, navigating, navigated =>navigate; thought, thinking =>think.
- Total 3228 words are obtained and all of them are used to index text documents.

In addition to the first 4 above steps, the second index approach extracts some domain-specific information using the algorithms presented in the Section 0 to enrich semantic information. By the second approach, the total 3636 words are obtained and used to represent text documents.

The number of words used for text index is very lower than mostly reported cases in the literature. The average number of words of documents is only 4. RCut thresh-

[5] http://www.cogsci.princeton.edu/cgi-bin/webwn2.0

olding strategy 8 is used to decide which category/ies is/are assigned to text documents. To evaluate the performance, micro- and macro- average recall, precision and F1 measures 4 are used. We take column's label plus explanation as documents in our experiments. That is, document= "column's label" + "column's explanation".

### 5.2 Results

Table 5.1 shows the experiment results by Rocchio for RCut = 1 and 3. For each case of RCut, the results are divided into 3 rows: the first row is the results by the first approach to text index without semantic enrichment treatment and the normal score calculated by Rocchio, where $\alpha = 0$ and $\beta = 1$; the second row is the results by the second approach to text index with semantic enrichment treatment but not taking into account the impact of units on categorization, where $\alpha = 0.3$ and $\beta = 1$; the third row is the results by the second approach to text index with semantic enrichment treatment and taking into account the impact of units on categorization, where $\alpha = 0.3$ and $\beta = 0$. $\beta = 0$ means that the categories will be rejected if the lists of units accepted by them do not contain unit associated with the columns to be categorized. We found that the best results are reached by Rocchio if $\alpha$=0.3 and $\beta$=0.

**Table 5.1.** Performance by Rocchio for RCut =1 and 3.

| RCut | α | β | micro-average | | | macro-average | | |
|---|---|---|---|---|---|---|---|---|
| | | | r | p | f1 | r | p | f1 |
| 1 | 0 | 1 | 77.3 | 77.1 | 77.2 | 78.2 | 79.6 | 76.7 |
| | 0.3 | 1 | 78.4 | 78.1 | 78.2 | 79.1 | 80.1 | 77.2 |
| | 0.3 | 0 | 81.3 | 82.1 | 81.7 | 81.7 | 83.7 | 80.4 |
| 3 | 0 | 1 | 90.8 | 30.6 | 45.8 | 91.1 | 36.6 | 50.1 |
| | 0.3 | 1 | 92.1 | 30.8 | 46.2 | 92.1 | 37.1 | 50.5 |
| | 0.3 | 0 | 93.2 | 35.9 | 51.8 | 93.1 | 44.1 | 56.8 |

### 5.3 Discussion

The results in Table 5.1 show that Rocchio together with our category score model performs well. The main idea of Rocchio is to build a centroid vector for each category and then the centroid vectors are explored to represent the main concepts discussed by the categories. If such centroid vectors indeed characterize the concept patterns contained in the categories, the Rocchio algorithms can work well.

In the case of UCD assignment, the centroid vectors for the UCDs can really represent main concept patterns of UCDs. In fact, the top 20 representative terms of centroid vectors can often describe the main concepts discussed by UCDs. For example, the UCD "PHOT_FLUX_IR_100" is about " Flux density (IRAS)" at "100 microns"

and the terms related to it are “flux”, “density”, “iras” and “100microns”. They are all present in the top 20 terms of its centroid vector.

Let us have a look at the contribution of semantic enrichment and column’s units to UCD assignments. According to Table 5.1, the micro-average performances of UCD assignment by Rocchio algorithm with semantic enrichment index increases by 1% compared to one by the trivial text index approach. This is due to the fact that semantic enrichment can improve text index. The following example can also confirm this fact.

**Example.** For the text “F100um Flux density at 100 micron”, its document vector is “100micron 0.5897, f100um 0.5897, micron 0.4172, density 0.2837, flux 0.2234” if semantic enrichment is performed; otherwise “f100um 0.7135, micron 0.5478, density 0.3432, flux 0.2703”. Its correct UCD is “PHOT_ FLUX_IR_100”. In these two cases, the possible 3 top UCDs returned by our system are as follow:

- PHOT_FLUX_IR_100 0.4690 0.3775
- PHOT_FLUX_IR_25 0.2551 0.3883
- PHOT_FLUX_IR_12 0.2512 0.3864

Here, the first and second numbers are the category scores calculated with semantic enrichment and without semantic enrichment respectively. Obviously, the result by semantic enrichment index is more reasonable.

We also notice that the gain of performance with semantic enrichment index together with column’s units is of at least 4% in terms of micro-average measures. This implies that units of columns can play important rule in correctly assigning UCD. This also provides useful hints for constructing UCD ontology. That is, information about units must be closely studied when defining UCD ontology.

The following example shows how column’s units improve the performances of UCD assignment.

**Example.** For the column whose label is “(O-C)Rho” and its explanation is “(O-C)Rho (O-C) in separation, arcseconds” and the linked unit is “deg”. Its correct UCD is “FIT_RESIDUAL”. The followings are the top 6 UCDs returned by Rocchio that are in descending order of category scores:

- CLASS_STAR/GALAXY 0.0767
- POS_EQ_DEC_OFF 0.0734
- POS_EQ_RA_OFF 0.0648
- FIT_RESIDUAL 0.0510
- PHYS_DISTANCE_TRUE 0.0162
- PHOT_COLOR_EXCESS 0.0053

The list of units accepted by the UCD “CLASS_STAR/GALAXY” does not contain the unit “deg”, so “CLASS_STAR/GALAXY” is rejected if $\beta$ =0. In fact, the units accepted by “CLASS_STAR/GALAXY” is only “%”. In this case, the final top 3 UCDs returned by categorization system include “POS_EQ_DEC_OFF”, “POS_EQ_RA_OFF” and “FIT_RESIDUAL”, whose lists of units contain the unit “deg”. The correct UCD “FIT_RESIDUAL” is returned if RCut=3 and $\beta$=0, while the first 3 top UCDs would be returned and “FIT_RESIDUAL” not returned if $\beta$ =1. It means that the number of correct UCDs returned can be increased if rejecting the UCDs with the list of accepted units that does not contain the units of unknown col-

umns. This example also confirms our assumption made in the Section 3.2. That is, one UCD should be punished by reducing its category score if the list of accepted units of the UCD does not contain the unit of the column to be categorized.

For RCut =3, both micro- and macro-recall are higher than 90% while the precisions are very low, since 3 possible UCDs are returned. Actually, the probability that the returned 3 possible UCDs include one correct UCD is 95%. Compared to the UCD assignment system at CDS, the results show that our UCD assignment system reaches very good performance. In addition, our model of category score defined by (3.4) increases the performance by 4.5% in terms of micro-average F1 and 6.7% in terms of macro-average F1 for RCut =1 and 3 respectively.

## 6 Conclusion and Future Works

The ambiguities of data table column's names reported in astronomy articles is a big problem for sharing observation data about sky objects in the astronomy field. By introducing the application of text categorization techniques, we attempt to cope with this problem and automatically build up an UCD assignment system. The experiment results have shown that domain-specific semantic enriching and utilization of domain-specific knowledge can improve performance of categorization.

In the framework of ACI-MDA, the parallel work focused on constructing an ontology of UCD is ongoing. In the future work, we will collaborate the knowledge of ontology of UCD and the built text categorization system.

## Acknowledgements

The authors would like to thank our astronomy colleagues of CDS. The research is supported by ACI scientific research funds of France.

## Reference

1. CDS, http://cdsweb.u-strasbg.fr/; VizieR, http://cdsweb.u-strasbg.fr/viz-bin/VizieR; UCD assignment, http://cdsweb.u-strasbg.fr/UCD/assign/
2. L. Cai and T. Hofmann, Text categorization by boosting automatically extracted concepts, Proceedings of the 26th SIGIR conference, Canada, pp.182-189, 2003.
3. T. Joachims, A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization, In the 14th Int. Conf. Machine Learning, pp.143-151,1997.
4. H. KOU, Intelligent Web Wrapper Generation Using Text Mining Techniques, PhD thesis, University of Versailles, July 2003.
5. Ortiz P. F., Ochsenbein F., Wicenec A., & Albrecht M., ESO/CDS Data-mining Tool Development Project, in ASP Conf. Ser., Vol. 172, eds. D. M. Mehringer, R. L. Plante, & D. A. Roberts (San Francisco: ASP), 1999.
6. Readme: http://vizier.u-strasbg. fr/doc/catstd.htx
7. E. Riloff, W. Lehnert, Information extraction as a basis for high-precision text classification, ACM Transactions on Information Systems, Vol.12, Issue 3, pp. 296 – 333, 1994.
8. Y. Yang, A study on thresholding strategies for text categorization, Proceedings of the 24th ACM SIGIR Conference, pp.137-145, 2001

# Towards an XML Representation of Proper Names and Their Relationships

Béatrice Bouchou, Mickael Tran*, and Denis Maurel

Université François-Rabelais de Tours - Laboratoire d'Informatique
DI de l'EPU de Tours, 64 avenue Jean Portalis
37200 Tours, France
{beatrice.bouchou,denis.maurel}@univ-tours.fr,
mickael.tran@etu.univ-tours.fr

**Abstract.** The presented work is a part of the Prolex project, whose aim is the design and implementation of a multi-lingual dictionary of proper names and their relationships. It focuses on the design of a standard XML representation for this kind of information. We first present the main lines of the conceptual model for proper names (a classical Entities / Relationships model), then we report on our experiment in designing an XML schema from this conceptual model. We describe the current resulting schema and discuss its main features.

## 1 Introduction

Since 1996, the Prolex project concerns proper names processing, particularly toponyms and inhabitant names [13], and stresses the need to link proper names together, e.g. in Foreign Affairs [14]. We have recently extended our project to every kind of proper names in a multilingual context [15]. We are creating a multilingual database of proper names, the Prolexbase, with linguistic information for natural language processing.

In the Prolex project, the need for an XML representation of proper names and their relationships has appeared first for interface purposes: a standard XML schema could enhance other ways for importing and exporting data, leading to more flexible exchanges or integration of data.

Indeed, according to classical database design, we have built a conceptual model, which has been translated into a logical model in order to efficiently store, maintain and use the dictionary of proper names. This has been done for the relational model: the french table counts more than 323000 entries and 55000 links of relation (these data have been translated into English, Italian, German, Spanich, etc.). As relationships between proper names are stored in the database, we can check whether some proper names are related, we can query for translations, etc. These are typical needs for our target applications: semantic tagging of texts, classification, translation, etc.

Now, there are several motivations for translating the conceptual model *also* into an XML schema:

– In the last few years, XML has become a logical data model, integrated into database applications: it appears however that the process of translating a conceptual model into an XML schema is an open challenge in itself.

* Supported by the RNTL-Technolangue project financed by the French Ministry of Industry.

A. Montoyo et al. (Eds.): NLDB 2005, LNCS 3513, pp. 44–55, 2005.
© Springer-Verlag Berlin Heidelberg 2005

- We wish the linguistic resource we are building to be widely used and nowadays XML is the standard way to integrate and/or exchange data: thus, XML can be a convenient interface layer for our relational database.
- Our schema represents a specialized vocabulary for proper names and should be used to describe terminal nodes in tagging models.

Our main contributions in this paper are to present a concrete experiment of XML schema design on the one hand, using an abstract notation to specify both the structure (schema) and the integrity constraints, and on the other hand to report on the current status of the XML schema for proper names (and their relationships), designed mainly on the basis of case studies of French and Serbian, for the moment.

The paper is organised as follows: in section 2 we present the conceptual model of proper names and their relationships. In section 3 we define the notation that we use for our XML schema, we describe the schema (and integrity constraints) and we discuss some of its features. In section 4 we conclude and present future work.

## 2 The PROLEX Conceptual Model

We have built a conceptual model, shown in Figure 1, derived from our ontology of proper names [10] which results from studies on their typology and on their inflectional and derivational mechanisms in different languages. In Figure 1, ontological concepts and their links are represented by entities (rectangles) and relationships (ovals). This model is structured in four layers which can be grouped in two parts: a multilingual part (*conceptual* and *metaconceptual* layers) and a monolingual part (*instances* and *linguistic* layers). Notice that each layer contains one main entity, which represents words in the layer of *instances*, lemmas in the *linguistic* layer, pivots in the *conceptual* layer and types in the *metaconceptual* layer.

### 2.1 Multilingual Part

The general architecture has been designed to be flexible enough in order to be applied to different languages without changing the interlingual structure, represented by the *conceptual* layer. The major concept for multilingual aspects is the *pivot*, a conceptual proper name used to connect proper names that represent the same concept in different languages (via the relationship *concept*). Relationships between proper names that are common to every languages are defined on pivots.

This is the case for the *synonymy*, which links pivots with a similar meaning (in a specific context called the *register*: politic, stylistic, diachronic, etc.). For instance, the synonymy in the diachronic register links pivots which represent a concept whose lemma has changed for historical reasons, *e.g. Saint-Petersburg* and *Leningrad* are linked to two different conceptual proper names related by this relationship.

The *predication* links two pivots which are arguments of the same predicate. It has been inspired at first by the lexical function *Cap* of Mel'čuk [12]. But it also includes other relationships like *London* is the capital of *England*, *Jacques Chirac* is the president of *France*, *Aaron* is the brother of *Moses*, etc. Notice that the relation of predication corresponds to a predicate of at least one language (instances of predicates are *president*, *capital*, etc.). The *meronymy*, inspired by WordNet, represents the link between a

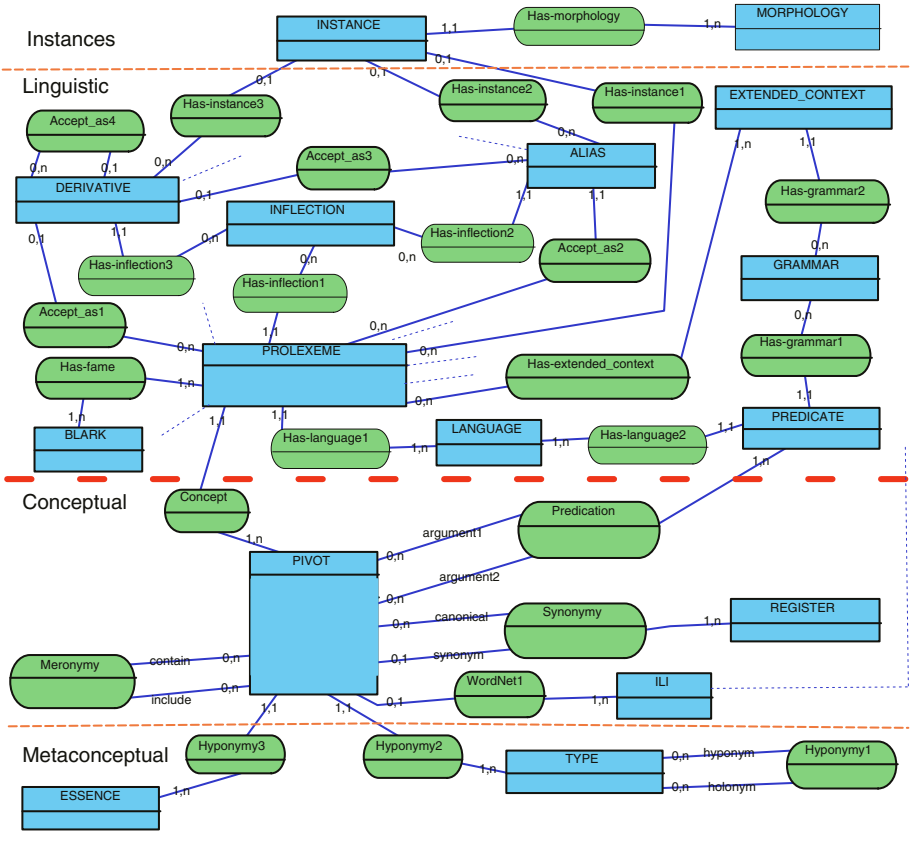



**Fig. 1.** The conceptual model of the Prolexbase.

whole and its parts. The *WordNet* relationship links a prolexeme and its EuroWordNet ILI (Inter-Lingual-Index) [17].

The metaconceptual layer contains metadata for pivots: *types*, which are hierarchically structured. There are four lexical classes of supertypes, *anthroponyms* (personal and collective names), *toponyms* (place names), *ergonyms* (artefacts and work names) and *pragmonyms* (event names). Simple types are restricted to a set of twenty-six lexical classes, that are determinated by close semantical characteristics. These classes include *organization*, *country*, *celebrity*, etc. The relationship *hyponymy1* is for type hierarchy, *hyponymy2* relates one pivot to its most specific *type*, and *hyponymy3* supports another kind of metadata for pivots: the *essence* specifies if the proper name belongs to a religious, historical or fictional domain.

## 2.2 Monolingual Part

The monolingual part is specific to each language. It consists in a *linguistic* description (there are big divergences between languages on morphological mechanisms applying

to proper names) and a set of words (the *instances*, which are all inflected forms of proper names) associated to their morphology.

The major concept in the *linguistic* layer of Figure 1 is the proper name: we use the term *prolexeme* to refer to the lemma of all the instances of a proper name. Proper names can have *aliases*, which are different variants of a prolexeme, uppercase or lowercase, diacritics, acronyms, abbreviations or transcriptions. Moreover proper names and their aliases may have *derivatives*: the lemma of the prolexeme allows to replace a word in a specific language by another one during translation. For example, in order to translate *It is the car of an inhabitant of Belgrade* in Serbian, we will have *To je Београђанинов ауто* where the proper name *Београђанинов* is in fact a derivative (more exactly a possessive adjective). The other concepts represented by entities associated to the prolexeme describe features of proper names. A *BLARK* (Basic Language Resources Kit) [6] is an indicator of fame which depends on different factors (the country, the period, etc.). An *extended context* points to a local grammar describing a context where the proper name can occur: it is useful in translation (as it varies from one language to another).

More information on proper names, not detailed in Figure 1, is supported: we allow to indicate if a proper name is linked to an *antonomasia*, a rhetoric device that indicates if we can substitute a phrase for a proper name or vice versa. For example, in English the proper name *biro* has become a common name for a ballpoint pen, whereas in French we use *bic*. We can store *Idiomatic expressions*: for example, *not for all the tea in China* in English will be translated into French by *pour rien au monde* (i.e. for nothing in the world). We have associated to every proper name information about its *sorting*. In most dictionaries, some multiword proper names are classified by permuting their units. For instance, in a French dictionary we will find *Mer d'Aral* under letter A. It is also sometimes useful to indicate whether a proper name may have an article (*determination*): *e.g.* the proper name *Spain* takes an article in French (*l'Espagne*). Finally, every prolexeme, alias or derivative is linked to an *inflection* paradigm.

## 3 The XML Representation

In the following, we first report on our experiment in translating the E/R (Entities/ Relationships) conceptual model into an XML schema (with constraints), then we present the resulting schema and discuss some of its features.

### 3.1 From Conceptual Model to XML Schema

There are surprisingly few works on methods of XML schema design, either from scratch or from conceptual models. Derivations from relational models [9] and from UML models [16] have been investigated, but compared to the vast amount of publications about the design of relational databases, this domain still lacks contributions. As we were dealing with an E/R model, we tried to follow steps described in [11] when it was possible. In particular, we use a grammatical notation of XML schemas similar to the one used in [11].

**Schema Notations.** There are several languages for describing schema of XML documents, and the choice between them is not obvious. DTDs are historically the first means to specify the structure of XML documents, and they are still widely used, even for specifying standards such as the Lexical Markup Framework (ISO standard [7]). But DTDs have shortcomings: in particular, in order to use XML as a logical model from a database point of view, it lacks means to define integrity constraints such as primary keys and foreign keys. In fact, dealing with these constraints in XML document is also a research area in itself ([5], [4]). The W3C consortium has proposed a formalism called XML Schema (or XSD) [3], which offers a variety of new constructs. But recent studies ([2]) tend to demonstrate that current schemas written in XSD only sparingly use these new features for structural specifications: most of them can be expressed by DTDs (the study does not address integrity constraints).

In this paper, we choose to use a high level schema notation which is coupled with a notation for integrity constraints: it is a tree grammar such as in [11]. Any schema written in any existing schema language can be easily translated into such a grammar.

**Definition 1. Grammar for schema:** The *grammar representing a schema* is denoted by a 6-tuple $\Gamma = (N, E, A, S, P, C)$, where

- $N$ is a finite set of non-terminal symbols (called *types*).
- $E$ is a finite set of element names.
- $A$ is a finite set of attribute names.
- $S$ is a set of start symbols, $S \subseteq N$.
- $P$ is a set of production rules of the form $X \rightarrow x(RE)$, where $X \in N$, $x \in E$ and $RE$ is a regular expression:
  $RE ::= \epsilon|\tau|@a|Y|(RE + RE)|(RE, RE)|(RE)?|(RE) * |(RE)+$
  where $\tau$ is an atomic data type, $\epsilon$ denotes the empty regular expression, $a \in A$, $Y \in N$.
- $C$ is a set of integrity constraints. □

Such a grammar offers a wide expressive power, but we will restrict ourselves to features that can be translated either into a DTD or into an XSD specification, for instance we consider only the atomic data type *string* (for $\tau$), we do not define regular expressions on attributes, etc. Attribute names are preceded by an @: in our schema (as in DTDs or XSD schemas) attributes are parts of element descriptions and contain only values.

From a database point of view, constraints are of fundamental importance, and specially primary and foreign keys: primary keys are a means of locating specific elements of the document and foreign keys allow to reference an element from another element (relationships). In particular, such information is used to maintain the connection from the concept in the real world to its representation when the system that is modeled evolves. As usual to define integrity constraints for XML, we use a subset of XPath expressions [8], precisely we use paths of the form $p ::= x|@a|p/p$, where $x \in E$, $a \in A$. Let $PE$ denote the set of such path expressions. We define the following notations for primary keys and foreign

keys: primary keys can be absolute or relative, and foreign keys are defined in the scope of the primary key they refer to.

**Definition 2. Integrity constraint specifications:**

- An *absolute primary key constraint* is specified as $pkey(X) = (p_1, ..., p_n)$, where $X \in N$ and $p_i \in PE$, $1 \leq i \leq n$. Paths must end with a data node, *ie* a node having a data value. The set $(p_1, ..., p_n)$ represents items composing the key for the type X. Notice that, as keys are specified for types, the schema must define an unambiguous type assignment.
- A *relative -primary- key constraint* is specified as $key(X)relative(Y) = (p_1, ..., p_n)$, where $X, Y \in N$ and $p_i \in PE$, $1 \leq i \leq n$. Such a specification indicates that *inside an element of type Y*, elements of type X are uniquely represented by the items in $(p_1, ..., p_n)$.
- A *foreign key constraint* is specified as $fkey(X, Y) = (p_1, ..., p_n)$, where $X, Y \in N$ and $p_i \in PE, 1 \leq i \leq n$. Such a specification indicates that items in $(p_1, ..., p_n)$, defined for type $X$, reference items in a key for type $Y$. □

In the schema for proper names and their relationships, we will use only *unary* (absolute and relative) primary keys (and thus unary foreign keys too).

**Design of the Target XML Schema.** Due to the lack of space, we could not analyze functional dependencies in section 2: therefore we can not detail here the translation steps. Although recommendations in [11] have been useful for first stages (to decide how to translate some relationships), it was not obvious to systematically derive an XML schema from the conceptual model. We departed from the method in [11] mainly in two points: we did not consider ID/IDREF(S) (special attribute types proposed in DTDs) as a useful way to express integrity constraints and we have had to strongly reorganize root's subelements.

Clearly, the design has been an iterative process: in fact, we even came back to the ontological level, refining the E/R conceptual model, in order to obtain a realistic XML schema to represent the *dictionary of proper names and their relationships*.

### 3.2 Schema for Proper Names

The schema grammar is $\Gamma = (N, E, A, S, P, C)$: we present it through its set of production rules $P$ (Figure 2) and its set of constraints $C$ (Figure 3). Items in $N$, $E$ and $A$ are introduced with production rules where they appear. The unique initial symbol in $S$ is *Root*. The first production rule **p1** specifies that a document containing proper names and their relationships is composed of two parts: *(i)* the paradigmatic *Relationships* part, shared by all natural languages (*i.e.* the *Conceptual* and *Metaconceptual* levels of E/R model in Figure 1), and *(ii)* the *Languages* description part, composed of one description for each language, containing proper names and their features (*i.e.* the *Linguistic* and *Instances* levels of E/R model in Figure 1). Notice that elements *relationships* and *languages* are compulsory but their content may be empty, in order to enhance partial descriptions.

**p 1** ***Root*** $\rightarrow root(Relationships, Languages)$

**p 2** ***Relationships*** $\rightarrow relationships(Pivot*, Predication*, Type*, WordNet)$

**p 3** ***Predication*** $\rightarrow predication(@pivot1, @pivot2, PReference+)$

**p 4** ***PReference*** $\rightarrow pReference(@language, @predicate)$

**p 5** ***Type*** $\rightarrow type(@name, Type*)$

**p 6** ***Pivot*** $\rightarrow pivot(@num, @essence, @type, @wordNet?, MeronymOf*, Canonical*,$
~ $Concept+)$

**p 7** ***MeronymOf*** $\rightarrow meronymOf(@pivot)$

**p 8** ***Canonical*** $\rightarrow canonical(@pivot, @register)$

**p 9** ***Concept*** $\rightarrow concept(@language, @prolexeme)$

**p 10** ***WordNet*** $\rightarrow wordNet(Ili*)$ ; ***Ili*** $\rightarrow ili(@num)$

**p 11** ***Languages*** $\rightarrow languages(Language+)$

**p 12** ***Language*** $\rightarrow language(@name, Prolexemes, ExtendedContexts, Predicates,$
~ $Idioms, Blarks, Statictics, Phonetics, Structures, Grammars, Inflections)$

**p 13** ***ExtendedContexts*** $\rightarrow extendedContexts(ExtendedContext*)$
~ ***ExtendedContext*** $\rightarrow extendedContext(@num, @name, @grammar)$

**p 14** ***Predicates*** $\rightarrow predicates(Predicate*)$
~ ***Predicate*** $\rightarrow predicate(@num, @name, @grammar)$

**p 15** ***Statistics*** $\rightarrow statistics(Stat*)$
~ ***Stat*** $\rightarrow stat(@num, @description, @weight)$

**p 16** ***Idioms*** $\rightarrow idioms(idiom*)$
~ ***Idiom*** $\rightarrow idiom(@num, @description)$
[...]

**p 17** ***Prolexemes*** $\rightarrow prolexemes(Prolexeme*)$
~ ***Prolexeme*** $\rightarrow prolexeme(@num, @name, @inflection, @pivot,$
~ $@determination?, @sorting?, @structure?, @IliAntonomasia?,$
~ $RIdiom*, RExtendedContext*, RBlark*, RStatistic*, RPhonetic*,$
~ $Aliases, Derivatives, Instances)$

**p 18** ***RIdiom*** $\rightarrow rIdiom(@idiom)$
~ ***RExtendedContext*** $\rightarrow rExtendedContext(@extendedContext)$
~ ***RBlark*** $\rightarrow rBlark(@blark)$
~ ***RStatistic*** $\rightarrow rStatistic(@statistic)$
~ ***RPhonetic*** $\rightarrow rPhonetic(@phonetic)$

**p 19** ***Aliases*** $\rightarrow aliases(Alias*)$
~ ***Alias*** $\rightarrow alias(@name, @category, @inflection, Instances, Derivatives?)$

**p 20** ***Derivatives*** $\rightarrow derivatives(Derivative*)$
~ ***Derivative*** $\rightarrow derivative(@name, @category, @inflection, Instances, Derivatives?)$

**p 21** ***Instances*** $\rightarrow instances(instance*)$
~ ***Instance*** $\rightarrow instance(@name, @morphology)$

**Fig. 2.** The production rules of the schema.

**c 1** $key(Pivot) =< @num >$

**c 2** $key(Predicate) =< @num >$

**c 3** $key(PReference)relative(Predication) =< @language >$

*// For a given predication element, there can not be two predicates in the same language.*

**c 4** $fkey(PReference) =< @predicate > REFERENCES(Predicate) < @num >$

**c 5** $fkey(Predication) =< @pivot1 > REFERENCES(Pivot) < @num >$

**c 6** $fkey(Predication) =< @pivot2 > REFERENCES(Pivot) < @num >$

**Fig. 3.** Examples of constraints.

**Conceptual and Metaconceptual Level.** The first part, rooted at element *relationships*, is specified by rules **p2** through **p10** in Figure 2. It is composed of (see rule **p2**):

- A list of elements of type *Pivot*: recall that the pivot is an abstract notion used to define general relationships between proper names.
- A list of elements of type *Predication*: such element links two pivots via a predicate of a given language.
- A list of elements of type *Type*: each type is the root of a hierarchically structured group of types. The hierarchy (recursive rule **p5**) reflects the relation of hyponymy.
- An element of type *WordNet*: it records links to WordNet ILIs.

Notice again that all lists can be empty (as well as the content of *wordNet* element), so *partial* views of the database can be valid with respect to the schema.

Before describing the type *Pivot*, we first give indications about the other sub-elements appearing in the content of an element tagged *relationships*. The relation of predication (rules **p3** and **p4**) links two pivots through several predicates, each predicate belonging to one language. In this way, having one pivot we can get the pivots it is linked with, and for each one the list of predicates (one predicate for one language, but one predication can exist in several languages, see for instance *brother of*, *frère de*, *hermano de*, etc.). In the same way, from a given predicate (in one language) we can obtain the two pivots and their related prolexemes (either in the same language or in other languages). We use keys and foreign keys (shown in Figure 3) in order to express these links (and to automatically verify them when updating documents, as usual in relational databases). Notice that key **c3** is relative: it is to ensure that *within one predication* there is at most one predicate for one language. The example in Figure 4 contains a predication indicating that *Paris* is the capital of *France*. Indeed, this instance of document contains two pivots in its *relationships* part which correspond to lemmas *Paris* and *France* in its *english language* part, and one predication corresponding to the predicate *capital*.

A *Pivot* (rules **p6** through **p10**) has a unique identifier, an *essence*, a *type* (notice that from that type we can get hyperonym types). An element *pivot* can refer to an entry in WordNet, it can be a meronym for a set of other pivots, it can reference canonical synonyms (in precise registers). Last, an element *pivot*

```
<root>
    <relationships>
        <pivot @num="400", @essence="historical", @type="city", @wordNet="05558236n">
            <canonical @pivot="410" @register="diachronic"/>
            <concept @language="english", @prolexeme="500" />
        </pivot>
        <pivot @num="600", @essence="historical", @type="country", @wordNet="05557178n">
            <concept @language="english", @prolexeme="800" />
        </pivot>
        <predication @pivot1="400", @pivot2="600"> <pReference @language="english", @predicate="500"/>
        </predication>
        <type @name="Toponym">      <type @name="Country"/>      <type @name="City"/> </type>
        ..........
        <wordNet >      <Ili @num="05558236n"/>      <Ili @num="05557178n"/>      </wordNet>
    </relationships>
    <languages>
        <language @name="english">
        <prolexemes>
            <prolexeme @num="500", @name="Paris", @determination="no", inflection="89", @pivot="400">
                <derivatives>
                    <derivative @name="Parisian", @category="3", @inflection="96" >
                        <instances>
                            <instance @name="Parisian", @morphology="S" />
                            <instance @name="Parisians", @morphology="P" />
                        </instances>
                    </derivative>
                    ..........
                </derivatives>
                <instances>      <instance @name="Paris", @morphology="S" />      </instances>
            </prolexeme>
            <prolexeme @num="800", @name="France", @determination="no", inflection="89", @pivot="600">
                <derivatives>
                    <derivative @name="French", @category="3", @inflection="96" >
                        <instances>      <instance @name="French", @morphology="S"/> ... </instances>
                    </derivative>
                    ..........
                </derivatives>
                <instances>      <instance @name="France", @morphology="S"/> </instances>
            </prolexeme>
        </prolexemes>
        <predicates>      <predicate @num="500", @name="capital", @grammar="12"/> ... </predicates>
        ..........
        </language>
    </languages>
</root>
```

**Fig. 4.** An example of proper names in XML: Paris and France.

represents a *concept* which exists in at least one language: for that reason, it is linked with one prolexeme of at least one language, and only one prolexeme per language. Obviously, it can be linked with several prolexemes, each one belonging to a different language. This is the same situation as for the predication relation (with predicates in languages). Therefore, we modelize it in the same way: *concept* elements are for *pivot* what *pReference* elements are for *predication*. Notice that the *language* in element *concept* as well as in element *pReference* is useful for translation applications. Indeed, in this way the access from one prolexeme (or one predicate) to corresponding prolexemes (or predicates) in other languages is immediate (via the pivot or via the predication).

**Linguistic and Instances Levels.** The second part of a document of proper names is rooted at element tagged *languages* (see rule **p11**). It contains information about at least one language, each language having a name (which is its key) and

containing a set of proper names and their descriptions (rule **p12**). A language can also have a list of *idioms*, useful for translation tasks.

Data about proper names (types in rule **p12**, except *Prolexemes*), are sets of information expressed in a standard $num - description$ shape: see rules **p13** through **p16**. The types *Blarks*, *Phonetics*, *Structures*, *Grammars* and *Inflections* are defined in the same way as *Idioms*. Notice that, as a *grammar* can be the same for several *predicate* elements, we chose to have a set of grammar descriptions and to reference one of these grammars from the *predicate* element: this reference is supported by a foreign key. The prolexemes themselves are under the element tagged *prolexemes*, whose type is described by rule **p17**: each *prolexeme* has a unique identifier *num*, a *name*, an $inflection$ code and a reference to its *pivot* (in order to address easily translation tasks for instance). Moreover, it can have information about its *determination* (in French it is *yes* or *no*) and about how to take its components into account in a *sorting* operation: for instance *2, 1* for *Jacques Chirac* indicates that the sorting must be done on *Chirac* first. The prolexeme can also have a reference to an internal *structure* (for compound proper names) and it can correspond to an *antonomasia*: in that case we allow to refer to the ILI WordNet of the corresponding common name, for translation purposes. The prolexeme can also refer to a set of *idioms* and a set of *extendedcontexts* in which it appears. It can be described by BLARKs, statistics and phonetics, too. For one prolexeme one can have sets of *aliases* and *derivatives* (these sets can be empty). Lastly, there is the set of *instances* (*values*) directly linked to the proper name.

Notice in rule **p18** that elements *rExtendedContext* encountered in a *prolexeme* contain just a reference to an *extendedContext* tagged element (which contains the description of the extended context). This is the same for BLARKs, statistics and phonetics, whereas *aliases*, *derivatives* and *instances* are fully described inside the prolexeme, as they are never shared by two distinct prolexemes (rules **p19** through **p21**).

Of course, all references are specified using keys and foreign keys: we do not describe every constraints for the sake of succinctness.

### 3.3 Discussion

The schema designed to represent proper names and their relationships takes advantage of XML nesting capabilities (*e.g.* defining recursive types), while avoiding much redundancies by following normalisation recommendations ([1]).

We have not found any need for union type (*i.e.* a type defined by a disjunctive regular expression), although it is a classical type of XML elements content, which denotes that the element is described either by some features or by other features. For instance, we can specify that a paper in a bibliography is either a presentation in a conference or an article in a journal. We have considered this capability in several places, *e.g.* when dealing with aliases and derivatives, but these two notions play different roles for a prolexeme, and one given prolexeme can have both aliases and derivatives... Hence, it seems that the target we modelize (proper names and their relationships) does not need union types.

The aim of modeling proper names and their relationships is to be as exhaustive as possible. Then, all details in the descriptions are present in the model. But we have carefully designed the schema in order that it can be usable even for partial descriptions (using optional contents every time it was possible).

The proper name description can be embedded in more general frameworks for modeling linguistic information. For instance the LMF (Lexical Markup Framework (ISO standard [7])) describes a high level model for representing data in lexical resources used in multilingual computer applications, including multilingual natural language processing lexicons. It is intended as a general framework, in which specialized vocabularies may be embedded without much difficulties. For that purpose, it provides a method for using *Feature Structures* and *Feature Values* to identify components of the lexical resource described. For instance, we could have: $< fname = numBlark+ > 000221 < /f >$ as an element part of description of a *prolexeme*.

It is clear that our approach is far more, let's say, normative: in fact we have design a schema in the classical spirit of database designers, specifying structures and constraints having in mind that there exist a (database) system to deal with these specifications in order to efficiently manage data, here the XML documents. By *managing* we mean classical tasks of a database system: storing, updating, querying, etc. On the contrary, a resource described in a framework such as ISO LMF could hardly take advantages of current and future XML generic tools, comprising database oriented tools.

## 4 Conclusions

We have presented a contribution to the Prolex project, recently developed within the RNTL-Technolangue project: the design of an XML schema for proper names and their relationships. XML schemas are useful for integration and/or exchange of data, in particular linguistic data.

During the design process, we tried to apply a method proposed in [11] which is to derive an XML schema directly from an (extended) E/R model. This E/R model is also briefly presented in this paper. Our conclusion was that such a derivation is not really straightforward. Nevertheless, following an iterative process we have obtained a structure (schema), together with integrity constraints, that accurately represent the concepts and relationships of the original E/R model.

Our XML schema is a basis for future work, in particular the specification of semantic tags for text markup. More generally, our aim is to use XML for developing new means of applying the dictionary of proper names in natural language processing tasks such as computer aided translation, information extraction, multilingual alignment text, etc.

## References

1. M. Arenas and L. Libkin. A normal form for XML documents. In *ACM Symposium on Principles of Database System*, 2002.

2. G. J. Bex, F. Neven, and J. Van den Bussche. DTDs versus XML schema: A practical study. In *Web and Databases (WebDB)*, 2004.
3. P. Biron and Eds. A. Malhotra. *XML Schema part 2*. http://www.w3.org/TR/xmlschema-2, 2001.
4. B. Bouchou, M. Halfeld Ferrari Alves, and M. Musicante. Tree automata to verify key constraints. In *Web and Databases (WebDB)*, 2003.
5. P. Buneman, S. Davidson, W. Fan, C. Hara, and W. Tan. Reasoning about keys for XML. In *Proceedings of Database and Programmming Languages*, 2001.
6. H. Strik C. Cucchiarini, W. Daelemans. Strengthening the dutch human language technology infrastructure. *http://www.elda.fr/article48.html*, 2000.
7. ISO/TC 37/SC 4 Committee. *Language resource management: Lexical Markup Framework*. ISO WD 24613, 2004(E), 2000.
8. M. Fernandez, A. Malhotra, J. Marsh, M. Nagy, and Eds. N. Walsh. *XQuery 1.0 and XPath 2.0 Data Model*. http://www.w3.org/TR/xpath-datamodel, 2004.
9. D. Lee, M. Mani, F. Chiu, and W. W. Chu. Net & Cot: Translating relational schemas to XML schemas. In *Australasian Database Conference*, 2002.
10. D. Maurel M. Tran, T. Grass. An ontology for multilingual treatment of proper names. In *OntoLex 2004, in Association with LREC2004*, pages 75–78, 2004.
11. M. Mani. Erex: a conceptual model for XML. In *Database and XML Technologies: XSym 2004, LNCS Volume 3186*, 2004.
12. I. Melćuk. Dictionnaire explicatif et combinatoire du français contemporain. *Les presses de l'Université de Montréal*, 1984-I, 1988-II, 1992-III.
13. C. Belleil O. Piton, D. Maurel. The prolex data base : Toponyms and gentiles for nlp. In *NLDB'99*, pages 233–237, 1999.
14. D. Maurel O. Piton. Beijing frowns and washington takes notice : Computer processing of relations between geographical proper names in foreign affairs. In *NLDB'2000*, pages 66–78, 2000.
15. D. Maurel O. Piton, T. Grass. Linguistic resource for nlp: Ask for *Die Drei Musketiere* and meet *Les Trois Mousquetaires*. In *NLDB'2003*, pages 200–213, 2003.
16. N. Routledge, L. Bird, and A. Goodchild. UML and XML schema. In *ACM Conference On Information and Knowledge Management (CIKM)*, 2002.
17. P. Vossen. EuroWordNet: A multilingual database with lexical semantic networks. *Kluwer Academic Publishers*, 1998.

# Empirical Textual Mining to Protein Entities Recognition from PubMed Corpus

Tyne Liang and Ping-Ke Shih

Department of Computer and Information Science
National Chiao Tung University, Hsinchu, Taiwan
tliang@cis.nctu.edu.tw

**Abstract.** Named Entity Recognition (NER) from biomedical literature is crucial in biomedical knowledge base automation. In this paper, both empirical rule and statistical approaches to protein entity recognition are presented and investigated on a general corpus GENIA 3.02p and a new domain-specific corpus SRC. Experimental results show the rules derived from SRC are useful though they are simpler and more general than the one used by other rule-based approaches. Meanwhile, a concise HMM-based model with rich set of features is presented and proved to be robust and competitive while comparing it to other successful hybrid models. Besides, the resolution of coordination variants common in entities recognition is addressed. By applying heuristic rules and clustering strategy, the presented resolver is proved to be feasible.

## 1 Introduction

Nowadays efficient automation of biomedical knowledge bases is urgently demanded to cope with the proliferation of biomedical researches. One crucial task involved in the automation is named entity recognition (NER) from biomedical literature. Similar to the recognition in general domains, the issues associated with biomedical entity recognition are open vocabulary, synonyms, boundaries and sense disambiguation. For example, the number of entries in SwissProt[1], a protein knowledge base, increases 277.36% in recent ten years. Each protein entity contains 2.54 synonyms in average, and each synonym contains 2.74 tokens in average.

Recent textual mining approaches useful to biomedical NER can be divided into rule-based, statistical and hybrid methods. Generally, rule-based approaches employ the information of terms and hand-craft rules to produce candidates which are then verified by using lexical analysis [1, 2, 5]. Yet rule-based methods require more domain knowledge and essentially lack of scalability. On the other hand, statistical models have been widely employed for their portability and scalability, such as Hidden Markov Model (HMM), Support Vector Model (SVM), Maximum Entropy (ME), and etc.. The recognition accuracy achieved by these models generally depends on a well-tagged training corpus and a well set

[1] SwissProt: http://us.expasy.org/sprot/

A. Montoyo et al. (Eds.): NLDB 2005, LNCS 3513, pp. 56–66, 2005.
© Springer-Verlag Berlin Heidelberg 2005

of features [3, 6, 7, 9, 10]. Recently, hybrid approaches are proposed by combining coded rules, statistical model and dictionaries [4, 9]. As pointed in [10], it is expected that systems on a specified evaluation corpus with help of dictionaries tend to perform better than the general ones without help of any dictionaries. For example, the recognition performance is significantly improved when dictionary and rules are applied at post-processing together with a ME-based recognition mechanism in [4].

In this paper, recognition for protein entities from PubMed[2] corpus is addressed so as to facilitate the automation of protein interaction databases construction. In order to mine more features relevant to protein entities, we assembled a domain-specific protein corpus SRC (SwissProt Reference Corpus) which were extracted from SwissProt reference articles and we tagged it by simply matching SwissProt entry collection. Experimental results show that this new domain corpus is indeed helpful in generating informative patterns used in both rule-based and statistical models. It is also found that though the derived rules are fewer and less complicated than the ones used in the rule-based systems Kex [1] or Yapex [5], the presented model outperforms these two systems in terms of higher F-scores on a general corpus like GENIA 3.02p [3] and the domain-specific SRC.

On the other hand, a concise HMM-based model is presented with a back-off strategy to overcome data sparseness. With a rich set of features, the presented approaches could achieve promising results, by showing 76-77% F-scores on both GENIA corpus and SRC. Compared to the results achieved by some successful systems (the best 78% F-score for protein instances in [9]) which employ dictionaries or semantic lexicon lists, our results are competitive for three reasons. First, the recognition is done without any help of dictionaries or predefined lexicon lists. Second, the presented concise HMM is easily implemented and robust for different corpora. Third, our results are evaluated with strict annotation and enetities with the longest annotation are adopted in case they are in the nested forms.

Besides, this paper addresses the issue of coordination variants while we tackle with NER problems in written texts. To resolve such term variants, a method based on heuristic rules and clustering strategy is presented. Experimental results on GENIA corpus 3.0 proved its feasibility by achieving 88.51% recall and 57.04% precision on a test of 1850 sentences, including 174 variants.

## 2 Corpus Preparation

In order to boost protein entities recognition by mining more relevant information, we assembled a domain-specific corpus 'SwissProt Ref Corpus' ('SRC' for short), other than the widely-used tagged corpus like GENIA 3.02p. The new corpus was processed by employing Sentence Splitter[4] and Penn Treebank

[2] PubMed: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=Pubmed
[3] http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/
[4] Sentence Splitter: http://l2r.cs.uiuc.edu/~cogcomp/

Tokenizer[5] for sentence segmentation and tokenization respectively. The POS-tagging is processed by a HMM-based POS tagger which was developed in our lab. By using GENIA 3.02p as training set, our POS-tagger could yield 95% F-score. For the sake of saving human efforts, annotating SRC with all the target entities was simply implemented with the following steps:

1. Tokens are split by space and hyphen.
2. Each token is converted to lower case except its initial character.
3. Entity is recognized if it matches an entity from SwissProt version 42.0.

The final specific SRC corpus is composed of 2,894 abstracts, which were particularly selected from SWISSPORT 82,740 reference articles in such a way that each of them contains at least six target entities. Table 1 lists the basic statistics for SRC and GENIA 3.02p.

**Table 1.** The statistics of SRC corpus and GENIA corpus 3.02p.

| | **SRC** | | **GENIA** | |
|---|---|---|---|---|
| | count | average | count | average |
| Abstract (a) | 2,894 | | 1,999 | |
| Sentence (s) | 28,154 | 9.73 (s/a) | 18,572 | 9.29 (s/a) |
| Token (t) | 740,001 | 255.70 (s/a) | 490.469 | 245.36 (t/a) |
| | | 26.28 (t/s) | | 26.41 (t/s) |
| Protein (p) | 31,977 | 11.05 (p/a) | 32,525 | 11.05 (p/a) |
| Entity | | 1.14 (p/s) | | 1.14 (p/a) |
| Entity Token (t) | 57,878 | 1.81(t/p) | 58,200 | 1.79 (t/p) |

## 3 Coordination Variants Resolution

Coordination variants are one common type of variants in general written texts like MEDLINE records. For example there are 1598 coordination variants in GENIA 3.02p corpus and each variant contains 2.1 entities in average. Table 2lists three types of the regular expressions generalized from the GENIA 3.02p training corpus of 16,684 sentences (in which 1421 coordination variants are distributed in 1329 sentences). There #, H, T, and R indicate core, head, tail, and coordinate terms respectively. For example, in the coordination '91 and 84 kDa proteins', '91' and '84' are the core terms, 'kDa proteins' is the tail term, and 'and' is the coordinate term.

The variant resolution was implemented with finite state machines (FSM) which are verified by a test set of 1850 sentences in which 174 variants are distributed in 165 sentences. Experimental results showed that this approach yielded 91.38% recall and 42.06% precision (indicated as ̈baseline approachïn Table 3). In practice, the precision can be improved by presenting more number of FSMs so as to cover all possible variant patterns, yet it will slow down the resolving throughput. In order to increase the sensitivity of coordination identification, a simple term clustering is employed. Suppose terms $t_i$, $t_j$ co-occur

[5] http://www.cis.upenn.edu/~treebank/tokenization.html

**Table 2.** Original patterns, expanded patterns, and examples.

| | Regular Expression | | Example |
|---|---|---|---|
| Type 1 | Original | H#(R#)$^+$ | human chromosomes 11p15 and 11p13 |
| | Expanded | (H#R)$^+$H# | humman chromosomes 11p15 and human chromosome 11p13 |
| Type 2 | Original | #(R#)$^+$T | c-fos, c-jun, and EGR2 mRNA |
| | Expanded | #T(R#)$^+$T | c-fos mRNA, c-jun mRNA, and EGR2 mRNA |
| Type 3 | Original | H#(R#)$^+$T | human T and B lymphocytes |
| | Expanded | #T(R#)$^+$T | human T lymphocytes and human B lymphocytes |

in one coordination variant, and terms $t_i$, $t_k$ co-occur in another one. Then we put $t_i$, $t_j$ and $t_k$ into one cluster. The clustering procedure was implemented recursively. With such term clustering strategy (indicated as 'unlimited-distance' in Table 3), the resolution precision is increased by 4%. This showed that the clustering approach is helpful to restrict the path movement in FSMs. To distinguish the closeness of the terms in the same cluster, we furthermore applied the Floyd-Warshall algorithm to cluster sets. That is, if terms $t_i$, $t_j$ co-occur in a sentence and terms $t_i$, $t_k$ co-occur in another one but $t_j$, $t_k$ do not co-occur in any sentence, then the $dist(t_j, t_k) = 2$. With this clustering strategy, the precision became 57.04% (increasing 15% with respect to the baseline method) at the expense of lower recall.

**Table 3.** Accuracy of coordination variants identification in GENIA 3.02p.

| | dist. | Variants | tp+fp | tp | Recall | Precision | F-Score |
|---|---|---|---|---|---|---|---|
| Baseline | N/A | 174 | 378 | 159 | 91.38% | 42.06% | 57.61% |
| Term Clustering | unlimited | 174 | 338 | 158 | 90.80% | 46.75% | 61.72% |
| | 1 | 174 | 270 | 154 | 88.51% | 57.04% | 69.37% |

## 4 Protein Entity Recognition

In this paper, protein entity recognition is approached and investigated by both rule-based and HMM models. The performance verification is implemented by using both SRC and GENIA 3.02p corpora in such a way that the corpora are divided into 90% for training phase and 10% for testing phase.

### 4.1 Rule-Based Approach

The rule-based recognition is implemented by employing the patterns of the protein nomenclature mined from SRC and GENIA corpora. The patterns are formed in terms of core, function or predefined terms. Core terms show the closest resemblance to regular proper names. Function terms describe the functions or characteristics of a protein. Table 4 shows the frequent regular expressions which 'C' indicates core term, 'F' indicates function term, and 'P' indicates predefined term, namely specifier, amino acid and unit.

**Table 4.** Top 5 regular expressions of protein entities in SRC and GENIA 3.02p.

| Regular Expression | SRC | Regular Expression | GENIA |
|---|---|---|---|
| $C^+$ | 25.70% | $C^+$ | 69.64% |
| $C^+F^+$ | 21.22% | $C^+F^+$ | 8.14% |
| $F^+$ | 15.57% | $C^+P^+$ | 5.84% |
| $F^+P^+$ | 12.62% | $F^+C^+$ | 2.91% |
| $C^+P^+$ | 9.36% | $F^+$ | 2.35% |

The function terms may be head or tail function term depending on the position they appear texts. From our observation of SRC, 58.48% head function terms appear before an initial uppercase token, and 74.07% tail function terms appear after an initial uppercase token or a specifier. We define 217 head function terms and 127 tail function terms. The rest of the terms other than predefined and function terms are treated as core terms candidates. The candidates may be the composition of common strings which are useful for identifying unknown words. For example, a common string 'CD' is acquired from a core term 'CD23', and then an unknown word 'CD25' will be seen as a core term.

The extraction of protein entities is done by six steps. The first three steps are aimed to produce the candidates by using term information. If a token is one of the three type terms, it will be annotated. Steps 4-6 are aimed to acquire protein entities as many as possible.

Step 1: boundary confirmation We scan the chunk forward (left to right) and backward (right to left) to fix entity boundaries by exploiting POS pattern information of protein entities, as shown in Tables 5 and 6.

**Table 5.** Top 5 POS patterns in SRC and GENIA.

| POS Pattern | SRC | POS Pattern | GENIA |
|---|---|---|---|
| NN | 79.38% | NN | 67.57% |
| NN,CD | 12.94% | JJ,NN | 7.13% |
| JJ,NN | 3.13% | NNS | 7.11% |
| JJ,NN | 3.02% | JJ,NNS | 2.94% |
| CD,NN | 0.26% | NN,CD | 0.96% |

**Table 6.** The top frequent POS tags at the first and the last positions of chunks.

| | First POS tag | | Last POS tag | |
|---|---|---|---|---|
| POS | SRC | GENIA | SRC | GENIA |
| CD | 0.27% | 0.43% | 13.12% | 1.91% |
| JJ | 6.32% | 13.23% | 3.03% | 0.57% |
| NN | 93.12% | 83.20% | 83.43% | 83.50% |
| NNS | 0.01% | 2.28% | 0.08% | 13.66% |
| VBN | 0.14% | 0.31% | 0.08% | 0.01% |

Step 2: remove invalid single-token chunks A single-token chunk will be treated as invalid if (a) its characters are in lower case, and the token is not a protein entity in training data or (b) it is a predefined term only.

Step 3: remove invalid multi-token chunks by using a general set of domain-independent rules. A chunk will be removed if it composes of the followings: (a) the predefined terms, (b) the single uppercase English letters, (c) the punctuation marks, and (d) the conjunctions. After the three steps, 68.21% and 52.63% invalid tokens in SRC and GENIA are removed 98.58% and 96.93% accuracy rates respectively.

Step 4: mine the tokens surrounding protein entities This step is to acquire more protein entities. The pattern is formulated as '$< T_{-2}, T_{-1}, \#, T_1, T_2 >$', where '#' is token's number of the protein entity, and the token '$T_i$' is the $i^{th}$ token relative to the protein entity. Two measurements namely, confidence and occurrence are used to justify the usefulness of the patterns. Confidence is the ratio of the number of correct instances divided by the number of all instances in training data, and occurrence is the number of all instances in training data. Patterns are selected whenever their occurrence and confidence are greater than one and 0.8 respectively, because our system is expected to achieve 80% correct rate, which is the ratio of the number of correct instances divided by the number of all retrieved instances.

Step 5: mine the bag-of-word surrounding protein entities For each protein entity we collect its preceding two tokens and following two tokens. The non-confidence is used to filter the candidates and it is defined as the ratio of the negative instances to all instances. Patterns are recognized whenever non-confidence is greater than 0.8 since our system is expected to yield 80% correct rate.

Step 6: employ syntactic rules Hypernyms may appear in front of hyponyms, and one common pattern is '$NP_0$ such as $\{\mathrm{NP}_1, \mathrm{NP}_2, \ldots, (\mathrm{and}|\mathrm{or})\ \}\ \mathrm{NP}_n$'. So we can mine those clue words by collecting the tokens preceding 'such as' and 'e.g.'. For example, 'protein' is the clue token of '... proteins, such as CBL and VAV, were phosphorylated on ...'. The clue words are the tokens of UMLS concepts and their corresponding synonyms which are tagged with 'protein' semantic type.

The model performance is evaluated in terms of precision (P), recall (R) and F-score (F) which is 2PR/(R+P). To present performance of rule-based systems, we use the notations of correct matching defined in [5]. Table 7 shows that the strict measure, which the proposed hit matches one answer key exactly, can yield 51%-52% F-Score. Table 7 shows that we can get higher F-score if we measure the performance with PNP ('protein name parts'), meaning each proposed token matches any token of the answer key. For example 'CD surface receptor' is treated as 'PNP' of 'activation of the CD28 surface receptor'. In practice, such kind of annotation result is acceptable. In addition, Table 7 also shows that the terms, mined from SRC, are adaptable since we can obtain almost the same performance results from GENIA corpus. Table 8 shows the improvement is obvious for steps 1 to 3, but steps 4 to 6 have little effect. On the other hand, the precision can be boosted obviously but not much for recall.

**Table 7.** Experimental results by rule-based approach.

| | Notation | tp+sn | tp+fp | tp | recall | precision | F-Score |
|---|---|---|---|---|---|---|---|
| SRC | SLOPPY | 3234 | 4782 | 2987 | 92.36% | 62.46% | 74.53% |
| | PNP | 3234 | 4782 | 2859 | 88.40% | 59.79% | 71.33% |
| | STRICT | 3234 | 4782 | 2077 | 64.22% | 43.43% | 51.82% |
| | LEFT | 3234 | 4782 | 2620 | 81.01% | 54.79% | 65.37% |
| | RIGHT | 3234 | 4782 | 2363 | 73.07% | 49.41% | 58.96% |
| | LorR | 3234 | 4782 | 2907 | 89.89% | 60.79% | 72.53% |
| | Notation | tp+sn | tp+fp | tp | recall | precision | F-Score |
| GENIA | SLOPPY | 3451 | 4923 | 3010 | 87.22% | 61.14% | 71.89% |
| | PNP | 3451 | 4923 | 2837 | 82.21% | 57.63% | 67.76% |
| | STRICT | 3451 | 4923 | 2123 | 61.52% | 43.12% | 50.70% |
| | LEFT | 3451 | 4923 | 2765 | 80.12% | 56.16% | 66.04% |
| | RIGHT | 3451 | 4923 | 2296 | 66.53% | 46.64% | 54.84% |
| | LorR | 3451 | 4923 | 2938 | 85.13% | 59.68% | 70.17% |

**Table 8.** The intermediate results of rule-based approach.

| | Procedure | tp+sn | tp+fp | tp | recall | precision | F-Score |
|---|---|---|---|---|---|---|---|
| SRC | step1 | 3234 | 10480 | 2051 | 63.42% | 19.57% | 29.91% |
| | step1-2 | 3234 | 5493 | 2043 | 63.17% | 37.19% | 46.82% |
| | step1-3 | 3234 | 4911 | 2040 | 63.08% | 41.54% | 50.09% |
| | step1-4 | 3234 | 4977 | 2104 | 65.06% | 42.27% | 51.25% |
| | step1-5 | 3234 | 4781 | 2077 | 64.22% | 43.33% | 51.83% |
| | step1-6 | 3234 | 4782 | 2077 | 64.22% | 43.43% | 51.82% |
| | Procedure | tp+sn | tp+fp | tp | recall | precision | F-Score |
| GENIA | step1 | 3451 | 7911 | 2160 | 62.59% | 27.30% | 38.02% |
| | step1-2 | 3451 | 5173 | 2129 | 61.69% | 41.16% | 49.37% |
| | step1-3 | 3451 | 5082 | 2127 | 61.63% | 41.85% | 49.85% |
| | step1-4 | 3451 | 5164 | 2155 | 62.45% | 41.73% | 50.03% |
| | step1-5 | 3451 | 4915 | 2120 | 61.43% | 43.13% | 50.68% |
| | step1-6 | 3451 | 4923 | 2123 | 51.52% | 43.12% | 50.70% |

### 4.2 HMM-Based Approaches

The statistical approach for NER is implemented by a concise HMM model (Concise-HMM) which employs a rich set of input features. Its performance is verified with SRC and GENIA 3.02p by comparing two other models, namely, traditional model (Traditional-HMM) and mutual information model (MI-HMM) which was presented in [9] and produced high F-scores in MUC-6 and MUC-7. The comparison is made in the same environment settings.

In this paper, all the models are trained with the same set of useful features including internal, external and global features. Internal features are those surface clues in tokens (e.g. initial character is upper case). There are 17 internal features mined from the training corpus. External features indicate the external information associated with tokens. We treated POS tags as our external features. Global features are the trigger nouns extracted from whole training

corpus by using Chi-square test. Besides, the complete-link clustering algorithm is applied to the mined nouns so as to reduce their dimensions. For window size of three sentences, we have 214 and 142 noun clusters in SRC and GENIA corpus respectively.

**Traditional HMM.** Given a token sequence $T_1^n = t_1 t_2 \dots t_n$, the goal is to find an optimal state sequence $S_1^n = s_1 s_2 \dots s_n$ that maximizes $\log Pr(S_1^n | T_1^n)$, the logarithm probability of state sequence $S_1^n$ corresponding to the given token sequence $T_1^n$. By applying Bayes's rule to

$$Pr(S_1^n | T_1^n) = \frac{Pr(S_1^n | T_1^n)}{Pr(T_1^n)} \tag{1}$$

we have

$$\arg\ {}^{\max}_{S} \log Pr(S_1^n | T_1^n) = \arg\ {}^{\max}_{S} \log Pr(S_1^n | T_1^n) + \log Pr(S_1^n)) \tag{2}$$

where

$$Pr(T_1^n | S_1^n) = \prod_{i=1}^{n} Pr(t_i | s_i) \tag{3}$$

and

$$Pr(S_1^n) = \prod_{i=1}^{n} Pr(s_i | s_{i-1}) \tag{4}$$

with the assumption of conditional probability independence and considering preceding state. Therefore equation (2) can be rewritten as:

$$\arg\ {}^{\max}_{S} \log Pr(S_1^n | T_1^n) = \arg\ {}^{\max}_{S} \left( \sum_{i-1}^{n} (\log Pr(t_i | s_i) + \log Pr(s_i | s_{i-1})) \right) \tag{5}$$

**MI-HMM.** Different from traditional HMM, MI-HMM is aimed to maximize the equation:

$$\arg\ {}^{\max}_{S} \log Pr(S_1^n | T_1^n) = \arg\ {}^{\max}_{S} \left( \log Pr(S_1^n) + \log \frac{Pr(S_1^n, T_1^n)}{Pr(S_1^n) \bullet Pr(T_1^n)}) \right) \tag{6}$$

In order to simplify the computation, the mutual information independence is assumed to be:

$$MI(S_1^n, T_1^n) = \sum_{i=1}^{n} MI(s_i, T_1^n) \tag{7}$$

or

$$\log \frac{Pr(S_1^n, T_1^n)}{Pr(S_1^n) \bullet Pr(T_1^n)} = \sum_{i=1}^{n} \log \frac{Pr(s_i, T_1^n)}{Pr(s_i) \bullet Pr(T_1^n)} \tag{8}$$

Applying it to equation (6), we have:

$$\arg\ {}^{\max}_{S} \log Pr(S_1^n | T_1^n) = \arg\ {}^{\max}_{S} \left( \log Pr(S_1^n) - \sum_{i=1}^{n} \log Pr(s_i) + \sum_{i=1}^{n} \log Pr(s_i | T_1^n) \right) \tag{9}$$

**Concise HMM.** The presented concise HMM is based on the idea of maximizing the fundamental $\log Pr(S_1^n|T_1^n)$. In the equation (9), $\log Pr(S_1^n|T_1^n)$ and $\sum_{i=1}^{n} \log Pr(s_i)$ are found to carry less meaning because the weak probabilities of states and state transitions are merely 3-by-3 and 3-by-1 matrices respectively. Thus, a concise HMM can be obtained by simplifying the formula (9) to be equation (10):

$$\arg \max_S \log Pr(S_1^n|T_1^n) = \arg \max_S \log Pr(S_1^n) - \sum_{i=1}^{n} \log Pr(s_i|T_1^n) \qquad (10)$$

Since the concise HMM does not take its state transition into account, we put previous state in the model to ensure correct state induction. Because the presented HMM approach concerned many features mentioned above, it is possible to train a high-accuracy probability model. To overcome spareseness problem, we use a back-off strategy which aims at the token sequence $T_1^n$ in $Pr(S_1^n|T_1^n)$ or in $Pr(s_i|T_1^n)$ where $T_1^n$ represents not only a token sequence but also the full set of sequence's features. There are two back-off levels. First level is based on different combinations of tokens and their features, and $T_1^n$ will be assigned in the descending order:

$< s_{-1}, t_{-1}, t_0, f_0 >, < s_{-1}, t_0, f_0 >, < s_{-1}, t_{-1}, f_0 >, < s_{-1}, f_0 >$

where $f_i$ represents the feature set including internal, external and global features. $t_i$ is a token, $s_i$ expresses a HMM state, and $i$ is the $i^{th}$ one relative to current token. Second level is based on different combinations of features, and $f_i$ in first level is assigned in the descending order:

$< f_i^I, f_i^E, f_i^G >, < f_i^I, f_i^E >, < f_i^I >$

where $f_i^I$ , $f_i^E$ and $f_i^G$ represent internal, external and global features, respectively.

### 4.3 Method Comparisons

Method comparisons for the three HMM-based models were made on both SRC corpus and GENIA corpus in the same environment settings. We used the same back-off model for concise and mutual information HMM, but not for traditional HMM. Table 9 shows that concise HMM with rule-based features (i.e. concise-ruled) yielded the best result. Traditional HMM obtains good high precision, but low recall since we chose a severe probability model to get the best F-score. It is also noticed that the performance of MI-HMM turned out to be the worst because the back-off model was used to optimize concise HMM. On the other hand, Table 10 shows all kinds of features turned out to be positive effect ($f^E > f^I > f^G$) for concise HMM. Such result is similar to that concluded from [10]. Table 11 lists the comparisons of the presented approaches to other well-known approaches on the public evaluation GENIA 3.x corpus. It is noticed that the presented rule-based approach with its simple general rules outperformed the other two complicated rule-based systems. On the other hand, the performance of the presented concise HMM-based models is comparable to the best model presented in [4]. However, we do not need any dictionary or rules in our model.

**Table 9.** HMM-based model comparison.

| | HMM | tp+sn | tp+fp | tp | recall | precision | F-Score |
|---|---|---|---|---|---|---|---|
| SRC | Concise | 3234 | 2953 | 2355 | 72.82% | 79.75% | 76.13% |
| | Concise-ruled | 3234 | 2949 | 2391 | 73.93% | 81.08% | 77.34% |
| | MI | 3234 | 3439 | 2384 | 73.72% | 69.32% | 71.45% |
| | Traditional | 3234 | 2396 | 2086 | 64.50% | 87.06% | 74.10% |
| | HMM | tp+sn | tp+fp | tp | recall | precision | F-Score |
| GENIA | Concise | 3451 | 3285 | 2553 | 73.98% | 77.72% | 75.80% |
| | Concise-ruled | 3451 | 3323 | 2596 | 75.22% | 78.12% | 76.65% |
| | MI | 3451 | 3415 | 2305 | 66.79% | 67.50% | 67.14% |
| | Traditional | 3451 | 2863 | 2263 | 65.58% | 79.04% | 71.68% |

**Table 10.** The effects of features in concise HMM.

| | Features | tp+sn | tp+fp | tp | recall | precision | F-Score | Diff. |
|---|---|---|---|---|---|---|---|---|
| SRC | All | 3234 | 2953 | 2355 | 72.82% | 79.75% | 76.13% | |
| | All-$f^G$ | 3234 | 2951 | 2335 | 72.20% | 79.13% | 75.51% | -0.62% |
| | All-$f^E$ | 3234 | 2894 | 2284 | 70.62% | 78.92% | 74.54% | -1.59% |
| | All-$f^I$ | 3234 | 2941 | 2303 | 71.21% | 78.31% | 74.59% | -1.54% |
| | Features | tp+sn | tp+fp | tp | recall | precision | F-Score | Diff. |
| GENIA | ALL | 3451 | 3285 | 2553 | 73.98% | 77.72% | 75.80% | |
| | All-$f^G$ | 3451 | 3267 | 2534 | 73.43% | 77.56% | 75.44% | -0.36% |
| | All-$f^E$ | 3451 | 3176 | 2442 | 70.76% | 76.89% | 73.70% | -2.10% |
| | All-$f^I$ | 3451 | 3213 | 2467 | 71.49% | 76.78% | 74.04% | -1.76% |

**Table 11.** Comparison to other systems on GENIA corpus.

| System | Method | GENIA | Recall | Precision | F-Score |
|---|---|---|---|---|---|
| Lee et. al. [3] | SVM | 3.0p | 78.80% | 61.70% | 69.20% |
| Lin et. al. [4] | ME-hybrid | 3.01 | 77.00% | 80.00% | 78.50% |
| KeX | Rule-based | 3.02p | 43.67% | 37.40% | 40.29% |
| Yapex | Rule-based | 3.02p | 45.06% | 54.17% | 47.48% |
| Ours | Rule-based | 3.02p | 61.52% | 43.12% | 50.70% |
| | concise-HMM | 3.02p | 73.98% | 77.72% | 75.80% |
| | concise-ruled | 3.02p | 75.22% | 78.12% | 76.64% |

## 5 Conclusions and Future Work

In this paper, we presented different textual mining strategies applicable to supporting full automation of protein entities recognition. Recognition for the entities in coordination variants is also concerned. To our best knowledge, our approach is the first one to cope with the term variants in the named entity extraction from biomedical texts. On the other hand, practical textual mining to protein entities recognition were presented by both rule and statistical models. Without the help of any dictionaries, the kernel recognition based on a concise HMM-based model turns out to be promising for protein entity extraction.

Future work includes the manual annotation correction of SRC for fine classification, exploitation of dictionaries for better recognition performance and the improvement of the resolution for coordination variants by using the semantic type information of biomedical thesaurus like UMLS. In addition, novel mining techniques to resolve other types of term variants should be explored for full NER automation.

## Acknowledgements

This research is partially supported by MediaTek Research Center, National Chiao Tung University, Taiwan and partially supported by National Science Council under the contract NSC 93-2213-E-009-074.

## References

1. Fukuda, K. and Tsunoda, T. and Tamura, A. and Takagi, T.: Towards Information Extraction: identifying Protein Names from Biological Papers. The 3rd Pacific Symposium on Biocomputing. (1998) 707-718.
2. Hou, W. J. and Chen, H. H.: Enhancing Performance of Protein Name Recognizers using Collocation. ACL 2003 Workshop on Natural Language Processing in Biomedicine, (2003) 25-32.
3. Lee, K.J. and Hwang, Y.S. and Rim, H.C.: Two-Phase Biomedical NE Recognition based on SVMs. ACL 2003 Workshop on Natural Language Processing in Biomedicine, (2003) 33-40.
4. Lin, Y. and Tsai, T. and Chiou, W. and Wu K. and Sung, T.-Y. and Hsu, W-L.: A Maximum Entropy Approach to Biomedical Named Entity Recognition. 4th Workshop on Data Mining in Bioinformatics (2004).
5. Olsson, F., Eriksson, G., Franzen, K., Asker, L., and Liden, P.: Notions of Correctness when Evaluating Protein Name Taggers. 19th International Conference on Computational Linguistics. (2002) 765-771.
6. Settles, B.: Biomedical Named Entity Recognition Using Conditional Random Fields and Rich Feature Sets. Int'l Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA), Geneva, Switzerland (2004).
7. Takeuchi, K. and Collier, N.: Bio-Medical Entity Extraction using Support Vector Machines. ACL 2003 Workshop on Natural Language Processing in Biomedicine, (2003) 57-64.
8. Tsuruoka, Y. and Tsujii, J.: Boosting Precision and Recall of Dictionary-based Protein Name Recognition. ACL 2003 Workshop on Natural Language Processing in Biomedicine (2003) 41-48.
9. Zhou, G.D. and Su, J.: Named Entity Recognition using an HMM-based Chunk Tagger. 40th Annual Meeting of the Association for Computational Linguistics (2002).
10. Zhou, G., Zhang, J., Su, J., Shen, D. and Tan, C. L.: Recognizing Names in Biomedical Texts: A Machine Learning Approach. Bioinformatics, Vol. 20, (2004)1178-1190.

# Automatic Extraction of Semantic Relationships for WordNet by Means of Pattern Learning from Wikipedia*

Maria Ruiz-Casado, Enrique Alfonseca, and Pablo Castells

Computer Science Dep., Universidad Autonoma de Madrid, 28049 Madrid, Spain
{Maria.Ruiz,Enrique.Alfonseca,Pablo.Castells}@uam.es

**Abstract.** This paper describes an automatic approach to identify lexical patterns which represent semantic relationships between concepts, from an on-line encyclopedia. Next, these patterns can be applied to extend existing ontologies or semantic networks with new relations. The experiments have been performed with the Simple English Wikipedia and WordNet 1.7. A new algorithm has been devised for automatically generalising the lexical patterns found in the encyclopedia entries. We have found general patterns for the hyperonymy, hyponymy, holonymy and meronymy relations and, using them, we have extracted more than 1200 new relationships that did not appear in WordNet originally. The precision of these relationships ranges between 0.61 and 0.69, depending on the relation.

## 1 Introduction

The exponential growth of the amount of data in the World Wide Web (WWW) requires the automatising of processes like searching, retrieving and maintaining information. One of the difficulties that prevents the complete automatising of those processes [1] is the fact that the contents in the WWW are presented mainly in natural language, whose ambiguities are hard to be processed by a machine.

The Semantic Web (SW) constitutes an initiative to extend the web with machine readable contents and automated services far beyond current capabilities [2]. A common practise is the annotation of the contents of web pages using an ontology. One of the most accepted definitions of an ontology is "an explicit specification of a conceptualisation" [3]. In most of the cases, ontologies are structured as hierarchies of concepts, by means of the relation called hyponymy (*is-a*, class inclusion or subsumption) and its inverse hyperonymy, which arranges the concepts from the most general to the most specific one. Additionally, there may be other relationships, such as meronymy (the part-whole relation) and its inverse holonymy; telicity (*purpose*), or any other which may be of interest, such as *is-author-of*, *is-the-capital-of*, *is-employee-of*, etc. In many cases, ontologies distinguish nodes that represent concepts (classes of things, e.g. *person*) from nodes that represent instances (examples of concepts, e.g. *John*) [4].

Like the web itself, sometimes, these ontologies have to include a high amount of information, or they undergo a rapid evolution. Therefore, it would be also highly

* This work has been sponsored by CICYT, project numbers TIC2002-01948 and TIN2004-03140.

A. Montoyo et al. (Eds.): NLDB 2005, LNCS 3513, pp. 67–79, 2005.
© Springer-Verlag Berlin Heidelberg 2005

desirable to automatise or semi-automatise the acquisition of the ontologies as well. This problem has been object of recent increasing interest, and new approaches for automatic ontology enrichment and population are being developed, which combine resources and techniques from Natural Language Processing, Information Extraction, Machine Learning and Text Mining [5, 6].

In this paper, we present a procedure for automatically enriching an existing lexical semantic network with new relationships extracted from on-line encyclopedic information. The semantic network chosen is WordNet [7], given that it is currently used in many applications, although the procedure is general enough to be used with other ontologies. The encyclopedia used is the Wikipedia, a collaborative web-based resource which is being constantly updated by its users. In this experiments, we have worked with its Simple English version[1], because the vocabulary and syntactic structures found in Simple English are easier to handle by a parser than those in fully unrestricted text. In addition, the fact that it is written with less supervision than an academic encyclopedia means that the language used is freer, sometimes colloquial, and the techniques that work well here are expected to be easier to port to the web than if we worked with a more structured reference text.

This paper is structured in the following way: Section 2 describes related work; Sections 3 and 4 detail the approach followed, and the evaluation performed; finally, Section 5 concludes and points out open lines for future work.

## 2 Related Work

Automatic extraction of information from textual corpora is now a well-known field with many different applications. Concerning automatic ontology enrichment, we may classify current approaches in the following groups:

- Systems based on distributional properties of words: it consists in studying co-occurrence distributions of terms in order to calculate a semantic distance between the concepts represented by those terms. This distance metric can next be used for conceptual clustering [8, 9], Formal Concept Analysis [10] or for classifying words inside existing ontologies [11–14]. The previous are usually applied to enrich the ontologies with new concepts. On the other hand, [15] learn association rules from dependency relations between terms which, combined with heuristics, are used to extract non-taxonomic relations.
- Systems based on pattern extraction and matching: these rely on lexical or lexicosemantic patterns to discover ontological and non-taxonomic relationships between concepts in unrestricted text. [16–18] manually define regular expressions to extract hyponymy and part-of relationships. [19] learns such patterns for company merge relationships. [20] quantifies the error rate of a similar approach as 32%. [21] describes a combination of a pattern-based and a distributional-based approach, also for hyperonymy. [22] describe a whole framework which incorporates terminology extraction and ontology construction and pruning which takes into account, amongst other things, substring relationships for identifying hyperonyms.

[1] http://simple.wikipedia.org/wiki/Main_Page

– Systems based on dictionary definitions analysis [23–25], take advantage of the particular structure of dictionaries in order to extract hyperonymy relationships with which to arrange the concepts in an ontology. Concept definitions and glosses have been found very useful, as they are usually concise descriptions of the concepts and include the most salient information about them [26]. There are also several works which extract additional relationships from WordNet glosses, by disambiguating the words in the glosses [26–29].

## 3 Procedure

The procedure followed consists in crawling the Simple English version of Wikipedia, collecting all the entries, disambiguating them, and associating each other with relations. The steps carried out, similar to those described in [16, 19], are the following:

1. *Entry Sense Disambiguation*: This step consists in preprocessing the Wikipedia definitions and associating each Wikipedia entry to its corresponding WordNet synset, so the sense of the entry is explicitly determined.
2. *Pattern extraction*: For each entry, the definition is processed looking for words that are connected with the entry in Wikipedia by means of a hyperlink. If there is a relation in WordNet between the entry and any of those words, the context is analysed and a pattern is extracted for that relation.
3. *Pattern generalisation*: In this step, the patterns extracted in the previous step are compared with each other, and those that are found to be similar are automatically generalised.
4. *Identification of new relations*: the patterns are applied to discover new relations other than those already present in WordNet.

The following sections detail all the steps in the procedure:

### 3.1 Entry Sense Disambiguation

The goal of this step is to mark each entry in the Wikipedia with its corresponding synset in WordNet. To this aim, the entries are downloaded, and they are processed in the following way:

1. Those web pages which contain more than one definition are divided in separate files.
2. Most of the HTML tags are removed.
3. The definitions are processed with a sentence splitter, a part-of.speech-tagger and a stemmer [30].
4. For each entry, choose the WordNet synset whose sense is nearer according to the definition.

The disambiguation procedure, described in detail in [31], is mainly based on the Vector Space Model and the dot-product similarity metric, co-occurrence information and some heuristics. Approximately one third of the entries in Wikipedia are not found in WordNet, one third appear with just one sense (they are monosemous), and one third have multiple possible senses (they are polysemous).

The output of this pre-processing step is a list of Wikipedia disambiguated entries.

### 3.2 Pattern Extraction

In the previous step, every entry from the encyclopedia has been disambiguated using WordNet as the sense dictionary. The aim of this step is the extraction of patterns relating two concepts such that they have already been disambiguated and they share a relation in WordNet. The process is the following:

1. For each term $t$ in the Wikipedia, with a definition $d$, we select every term $f$ such that there is a hyperlink within $d$ pointing to $f$. This assures that $f$'s entry also exists in Wikipedia, and its sense has been disambiguated in the previous step.
   The reason why we only select the terms which have an entry in Wikipedia is that we have obtained a higher accuracy disambiguating the entry terms than attempting a disambiguation of every word inside the definitions. In this way, we expect the patterns to be much more accurate.
   If a particular entry is not found in the disambiguated set, it is ignored, because it means that either the entry is not yet defined in the Wikipedia[2], or it was not found in WordNet and was not disambiguated previously.
2. Once we have found a hyperlink to other disambiguated entry, the following process is carried out:
   (a) Look up in WordNet relationships between the two terms.
   (b) If any relation is found, collect the sentence where the hyperlink appears (with part-of-speech tags).
   (c) Replace the hyperlink by the keyword TARGET.
   (d) If the entry term appears in the sentence, replace it by the keyword ENTRY.

This work uses WordNet 1.7, in which there are six possible relationships between nouns. The first four, hyperonymy, hyponymy, holonymy and meronymy have been included in this study. Concerning antonymy, this relationship in WordNet does not always refer to the same feature, as sometimes it relates nouns that differ in gender (e.g. *king* and *queen*), and, other times, in a different characteristic (e.g. *software* and *hardware*), so it would be very difficult to find a consistent set of patterns for it. With respect to synonymy, we found that there are very few sentences in Wikipedia that contain two synonyms together, as they are expected to be known by the reader and they are used indistinctly inside the entries.

For illustration, if the entry for *Lisbon* contains the sentence *Lisbon is part of Portugal*, the pattern produced would be the following: `ENTRY is/VBZ part/NN of/IN TARGET`. Note that the words are annotated with part-of-speech tags, using the labels defined for the Penn Treebank[32].

The output of this step consists of as many lists as relationships under study, each list containing patterns that are expected to model each particular relation for diverse pairs of words.

### 3.3 Pattern Generalisation (I): Edit Distance Calculation

In order to generalise two patterns, the general idea is to look for the similarities between them, and to remove all those things that they do not have in common.

[2] The Wikipedia is continuously refreshing its contents and growing, and some of the links of the definitions fail to bring to another definition.

The procedure used to obtain a similarity metric between two patterns, consists of a slightly modified version of the dynamic programming algorithm for *edit-distance* calculation [33]. The *edit distance* between two strings $A$ and $B$ is defined as the minimum number of changes (character insertion, addition or replacement) that have to be done to the first string in order to obtain the second one. The algorithm can be implemented as filling in a matrix $\mathcal{M}$ with the following procedure:

$$\mathcal{M}[0,0] = 0 \tag{1a}$$

$$\mathcal{M}[i,0] = M[\ i-1,0] + 1 \tag{1b}$$

$$\mathcal{M}[0,j] = M[0,j-1] + 1 \tag{1c}$$

$$\begin{aligned}\mathcal{M}[i,j] = min(&\mathcal{M}[i-1,j-1] + d(A[i],B[j]),\\ &\mathcal{M}[i-1,j] + 1,\\ &\mathcal{M}[i,j-1] + 1)\end{aligned} \tag{1d}$$

where $i\epsilon[1...|A|], j\epsilon[1...|B|]$ and

$$d(A[i],B[j]) = \begin{cases} 0 & \text{if } A[i] = B[j] \\ 1 & \text{otherwise} \end{cases}$$

In these equations, M[i,j] will contain the edit distance between the first $i$ elements of $A$ and the first $j$ elements of $B$. Equation (1a) indicates that, if $A$ and $B$ are both empty strings, the edit distance should be 0. Equations (1b) and (1c) mean that the edit distance between an empty string, and a string with $N$ symbols must be $N$. Finally, equation (1d) uses the fact that, in order to obtain a string[3] $A\sigma$ from a string $B\gamma$, we may proceed in three possible choices:

- We may obtain $A\gamma$ from $B\gamma$, and next substitute $\gamma$ by $\sigma$. If $\gamma$ and $\sigma$ are the same, no edition will be required.
- We may obtain $A\sigma\gamma$ from $B\gamma$, and next delete $\gamma$ at the end.
- We may obtain $A$ from $B\gamma$, and next insert the symbol $\sigma$ in the end.

In the end, the value at the rightmost lower position of the matrix is the edit distance between both strings. The same algorithm can be implemented for word patterns, if we consider that the basic element of each pattern is not a character but a whole token.

At the same time, while filling matrix $\mathcal{M}$, it is possible to fill in another matrix $\mathcal{D}$, in which we record which of the choices was selected as minimum in equation (1d). This can be used afterwards in order to have in mind which were the characters that both strings had in common, and in which places it was necessary to add, remove or replace characters. We have used the following four characters:

- `I` means that it is necessary to insert a token, in order to transform the first string into the second one.
- `R` means that it is necessary to remove a token.
- `E` means that the corresponding tokens are equal, so it is not necessary to edit them.
- `U` means that the corresponding tokens are unequal, so it is necessary to replace one by the other.

[3] $A\sigma$ represents the concatenation of string $A$ with character $\sigma$.

$A$: `It is a kind of`
$B$: `It is nice of`

| $\mathcal{M}$ | **0** | **1** | **2** | **3** | **4** |
|---|---|---|---|---|---|
| **0** | 0 | 1 | 2 | 3 | 4 |
| **1** | 1 | 0 | 1 | 2 | 3 |
| **2** | 2 | 1 | 0 | 1 | 2 |
| **3** | 3 | 2 | 1 | 1 | 2 |
| **4** | 4 | 3 | 2 | 2 | 2 |
| **5** | 5 | 4 | 3 | 3 | 2 |

| $\mathcal{D}$ | **0** | **1** | **2** | **3** | **4** |
|---|---|---|---|---|---|
| **0** | | I | I | I | I |
| **1** | R | E | I | I | I |
| **2** | R | R | E | I | I |
| **3** | R | R | R | U | I |
| **4** | R | R | R | R | U |
| **5** | R | R | R | R | E |

**Fig. 1.** Example of the edit distance algorithm. $A$ and $B$ are two word patterns; $\mathcal{M}$ is the matrix in which the edit distance is calculated, and $\mathcal{D}$ is the matrix indicating the choice that produced the minimal distance for each cell in $\mathcal{M}$.

Figure 1 shows an example for two patterns, $A$ and $B$, containing respectively 5 and 4 tokens. The first row and the first column in $\mathcal{M}$ would be filled during the initialisation, using Formulae (1b) and (1c). The corresponding cells in matrix $\mathcal{D}$ are filled in the following way: the first row is all filled with `I`'s, indicating that it is necessary to insert tokens to transform an empty string into $B$; and the first column is all filled with `R`'s indicating that it is necessary to remove tokens to transform $A$ into an empty string. Next, the remaining cells would be filled by the algorithm, looking, at each step, which is the choice that minimises the edit distance. $\mathcal{M}(5, 4)$ has the value 2, indicating the distance between the two complete patterns. For instance, the two editions would be replacing `a` by `nice`, and removing `kind`.

### 3.4 Pattern Generalisation (II): Algorithm

After calculating the edit distance between two patterns $A$ and $B$, we can use matrix $\mathcal{D}$ to obtain a generalised pattern, which should maintain the common tokens shared by them. The procedure used is the following:

1. Initialise the generalised pattern $G$ as the empty string.
2. Start at the last cell of the matrix $\mathcal{M}(i, j)$. In the example, it would be $\mathcal{M}(5, 4)$.
3. While we have not arrived to $\mathcal{M}(0, 0)$,
   (a) If $(\mathcal{D}(i, j) = \texttt{E})$, then the two patterns contained the same token A[i]=B[j].
      – Set $G = A[i]\ G$
      – Decrement both $i$ and $j$.
   (b) If $(\mathcal{D}(i, j) = \texttt{U})$, then the two patterns contained a different token.
      – $G = A[i]|B[j]\ G$, where | represents a disjunction of both terms.
      – Decrement both $i$ and $j$.
   (c) If $(\mathcal{D}(i, j) = \texttt{R})$, then the first pattern contained tokens not present in the other.
      – Set $G = *\ G$, where $*$ represents any sequence of terms.
      – Decrement $i$.
   (d) If $(\mathcal{D}(i, j) = \texttt{I})$, then the second pattern contained tokens not present in the other.
      – Set $G = *\ G$
      – Decrement $j$

If the algorithm is followed, the patterns in the example will produced the generalised pattern

```
It is a kind   of
It is nice     of
-------------------
It is a|nice * of
```

This pattern may match phrases such as *It is a kind of*, *It is nice of*, *It is a hyperonym of*, or *It is a type of*. As can be seen, the generalisation of these two rules produces one that can match a wide variety of sentences, and which may be indicating different kinds of relationships between concepts.

### 3.5 Pattern Generalisation (III): Generalisation with Part-of-Speech Tags

The previous example shows that, when two patterns are combined, sometimes the result of the generalisation is far too general, and matches a wide variety of sentences that don't share the same meaning. Therefore, in order to restrict the kinds of patterns that can combine to produce a generalisation, the algorithm has been extended to handle part-of-speech tags. Now, a pattern will be a sequence of terms, and each term will be annotated with a part-of-speech tag, as in the following examples:

(a) `It/PRP is/VBZ a/DT kind/NN of/IN`
(b) `It/PRP is/VBZ nice/JJ of/IN`
(c) `It/PRP is/VBZ the/DT type/NN of/IN`

The edit distance algorithm is modified in the following way: the system only allows replacement actions if the words from the two patterns $A$ and $B$ belong to the same general part-of-speech (nouns, verbs, adjectives, adverbs, etc.). Also, if this is the case, we consider that there is no edit distance between the two patterns. In this way, two patterns that do not differ in the part-of-speech of any of their words will be considered more similar than other pairs of patterns differing in the part-of-speech of one word. The $d$ function, therefore, is redefined as:

$$d(A[i], B[j]) = \begin{cases} 0 & \text{if } PoS(A[i]) = PoS(B[j]) \\ 1 & \text{otherwise} \end{cases} \quad (2)$$

The insertion and deletion actions are defined as before. Therefore, patterns (a) and (b) above would have an edit distance of 2, and the result of their generalisation is:

`It/PRP is/VBZ * of/IN`

On the other hand, the patterns (a) and (c) would have an edit distance of 0, and the result of their generalisation would be the following:

`It/PRP is/VBZ a|the/DT kind|type/NN of/IN`

Once the generalisation procedure has been defined, the following algorithm is used in order to generate the final set of generalised patterns:

1. Collect all the patterns from the Wikipedia entries in a set $\mathcal{P}$.
2. For each possible pair of patterns, calculate the edit distance between them.

3. Take the two patterns with the smallest edit distance, $p_i$ and $p_j$.
4. If the edit distance between them exceeds a threshold $\theta$, stop.
5. Otherwise,
   (a) Remove them from $\mathcal{P}$.
   (b) Calculate the more general pattern $p_g$ from them.
   (c) Add $p_g$ to $\mathcal{P}$.
6. Go back to step 2.

The previous algorithm is repeated for each relationship (e.g. hyponymy or meronymy). The output of the algorithm is the set containing all the rules that have been obtained by combining pairs of original rules. The purpose of the parameter $\theta$ is the following: if we set no limit to the algorithm, ultimately all the rules can be generalised to a single generalisation containing just one asterisk, which would match any text. Thus, it is desirable to stop merging rules when the outcome of the merge is too general and would be source of a large quantity of errors. The value of $\theta$ was set empirically to 5. For higher values of $\theta$, the system tried to generalise very different rules, resulting in rules with many asterisks and few lexical terms.

### 3.6 Identification of New Relations

Finally, given a set of patterns for a particular relation, they can be applied to all the entries in the Wikipedia corpus. Whenever a pattern matches, the target word is identified, and a candidate relationship is produced.

In this step, we took into account the fact that most relations of holonymy and meronymy are either between instances or between concepts, but not between an instance and a concept. For instance, it is correct to say that *Lisbon* is part of *Portugal*, but it does not sound correct to say that *Lisbon* is part of the concept *country*, even though Portugal is a country. Therefore, all the results obtained for holonymy and meronymy in which one of the two concepts related was an instance and the other was a concept were removed from the results. We have used the classification of WordNet synsets as instances or concepts provided by [34].

The output of this step is a list of extracted related pairs of entries for each relation.

## 4 Evaluation and Results

The algorithm has been evaluated with the whole Simple English Wikipedia entries, as available on November 15, 2004. Each of the entries was disambiguated using the procedure described in [31]. An evaluation of 360 entries, performed by two human judges, indicates that the precision of the disambiguation is 92% (87% for polysemous words). The high figure should not come as a surprise, given that, as can be expected, it is an easier problem to disambiguate the title of an encyclopedia entry (for which there exist much relevant data) than a word inside unrestricted text.

The next step consisted in extracting, from each Wikipedia entry $e$, a list of sentences containing references to other entries $f$ which are related with $e$ inside WordNet. This resulted in 270 sentences for hyponymy, 158 for hyperonymy, 247 for holonymy and 222 for meronymy. When analysing these patterns, however, we found that, both for

hyperonymy and meronymy, most of the sentences extracted only contained the name of the entry $f$ (the target of the relationship) with no contextual information around it. The reason was unveiled by examining the web pages:

- In the case of hyponyms and holonyms, it is very common to express the relationship with natural language, with expressions such as *A dog is a mammal*, or *A wheel is part of a car*.
- On the other hand, when describing hyperonyms and meronyms, their hyponyms and holonyms are usually expressed with enumerations, which tend to be formatted as HTML bullet lists. Therefore, the sentence splitter chunks each hyponym and each holonym as belonging to a separate sentence.

**Extraction of Hyponymy Relations.** A total of 1204 relations of hyponymy have been automatically extracted from the Wikipedia entries using the patterns that were found in the previous step (excluding repetitions). 352 out of them already appeared in WordNet (including those which appear through transitive closure), and the remaining 852 relations were evaluated by two human judges. Inter-judge agreement was very high for the case of hyponymy, reaching 99%. The overall precision is 0.69.

Table 1 shows some of the rules extracted, which were evaluated separately. The rule that applied most often is the classical hyponymy copular expression, `ENTRY is a TARGET`, which relates a concept with its hyperonym. There are five versions of this rule (numbered 1, 2, 3, 4 and 6) in the Table, allowing for extra tokens before and in between, and providing a long list of adjectives that may appear in the definition. Many of the errors produced by these patterns can be explained, given that, for sequences such as *the man with the telescope is the leader* in the corpus, the word *telescope* would be chosen as hyponym of *leader*, because the patterns do not have syntactic information.

Secondly, there are also patterns which have been extracted because of the characteristics of Wikipedia. For instance, there are several entries about months in the years, and all of them contain a variant of the sentence `XXX is the` $n$`th month in the year`. Therefore, rule 5 shows a pattern extracted from those sentences. Other example is that of colours, and all of which contain the same sentence, `List of colors`, in their definition.

Finally, rule 23 has been displayed as an example of a too specific rule that, because it is not very general, has not been able to identify any hyponymy relationship apart from those that were already in WordNet. This rule has been created, mostly, from definitions of planets in the Solar System (e.g. *Venus is the second planet from the Sun*) and from definitions of months in the year (e.g. *March is the third month in the Year*). When matched with the Wikipedia entries, it is just able to extract the known relationships for instances of planets and instances of months, so it does not generate new knowledge.

**Extraction of Hyperonymy Relations.** Concerning hyperonymy, as commented before, there were very few patterns to use, and they were very specific. Just four patterns were matched in the texts, resulting in four already known relationships.

**Extraction of Holonymy Relations.** Twenty rules for identification of holonymy relations matched the texts in 418 places. 115 were already present in WordNet (including

**Table 1.** Some of the rules obtained for the relation of hyponymy. Columns indicate the number of the rules, the new results produced by each rule, its precision and the text of the rule.

| No. | Match | Prec. | Rule |
|---|---|---|---|
| 1 | 2 | 1.0 | ENTRY/NN is/VBZ a/DT type/NN of/IN TARGET |
| 2 | 1 | 1.0 | ENTRY/NNP (/( */* :/, Jawa\|Kernow/NNP )/) is/VBZ */* a\|the/DT TARGET in\|of/IN England\|Indonesia/NNP |
| 3 | 139 | 0.86 | */* is\|was/VBD */* British\|English\|Greek\|alcoholic\|baked\|deadly\|non-metal\|old\|oldest/JJ TARGET |
| 4 | 370 | 0.7 | ENTRY/NN is/VBZ a/DT TARGET |
| 5 | 23 | 0.57 | TARGET of/IN the/DT Year/NN |
| 6 | 18 | 0.44 | */* ENTRY/NN is/VBZ a/DT TARGET founded\|used/VBN */* |
| ... | | | (up to 22 rules) |
| 23 | 0 | N/A | */* ENTRY\|Isotopes\|Jupiter\|Mercury\|Neptune\|Saturn\|Uranus\|Venus/NNP are\|is/VBZ */* big\|common\|different\|eighth\| fifth\|first\|largest\|nearest\|ninth\|second\|seventh\|sixth\|small \|third/JJ TARGET from\|in\|of\|on/IN */* Earth\|Ocean\|Sun\|days\|earth\|element\|sun\|year\|years/NNP |

**Table 2.** Rules obtained for the relation of holonymy, ordered by precision. Columns indicate the rules' number, number of new results found, precision and pattern.

| No. | Match | Prec. | Rule |
|---|---|---|---|
| 1 | 10 | 1.00 | ENTRY/NNP is/VBZ a/DT city\|province/NN in/IN TARGET |
| 2 | 5 | 1.00 | ENTRY/NNP */* the/DT */* capital\|city/NN */* capital\|city/NN of/IN TARGET |
| 3 | 1 | 1.00 | */* is\|was/VBZ one/CD of/IN the/DT */* States\|countries/NNPS in/IN the/DT TARGET |
| 4 | 25 | 0.84 | */* ENTRY/NNP is/VBZ */* a\|the/DT */* Lakes\|Republic\|canal\|capital\|city\|coast\|country\|northeast\|province\|region\|southwest\|state\|west/NN in\|of/IN TARGET |
| 5 | 120 | 0.59 | ENTRY/NNP is/VBZ a\|an\|the/DT */* in\|of/IN the/DT TARGET |
| 6 | 97 | 0.49 | */* Things\|city\|member\|north\|part\|planets\|state/NNS in\|of/IN the/DT TARGET |
| 7 | 4 | 0.75 | */* ENTRY/NNP is\|was/VBZ a/DT */* country\|part\|river/NN in\|of/IN */* eastern\|north\|northern/JJ TARGET |
| ... | | | (up to 20 rules) |

transitive closure), and the remaining 303 were evaluated by two judges. In this case, inter-judge agreement reached 95%. In order to unify the criteria, in the doubtful cases, similar cases were looked inside WordNet, and the judges tried to apply the same criteria as shown by those examples. The final precision for these patterns is 0.61.

Table 2 shows some of the rules for holonymy. Most of the *member part-of* and *substance part-of* relations were rightly extracted by rules 4, 5 and 6, which match sentences such as *X is in Y* or *X is a part of Y*. However, they also produced some wrong relations. Many patterns focused on locations, such as rules 1, 2, 3, 4, 6 and 7.

In the case of holonymy, an important source of errors was the lack of a multi-word expression recogniser. Many of the part-of relations that appear in Wikipedia are relations between instances, and a large portion of them have multi-word names. For instance, the application of the set of patterns to the sentence

*Oahu is the third largest of the Hawaiian Islands*

returns the relation *Oahu is part of Islands*, because *Hawaiian_Islands* has not been previously identified as a multi-word named entity.

Other errors were due to orthographic errors in the Wikipedia entry (e.g. *Lourve* instead of *Louvre*) and relations of holonymy which held in the past, but which are not true by now, such as *New York City is part of Holland* or *Caribbean Sea is part of Spain*.

**Extraction of Meronymy Relations.** Concerning the last kind of relationship studied, meronymy, 184 new relations were found, out of which 115 already were known, 42 were judged correct, and 27 were judged wrong, which results in an overall precision of 0.61. As is the case with hyperonymy, the number of patterns and relations extracted is much lower than for their inverse relations.

## 5 Conclusions and Future Work

This work addresses the problem of automatically identifying semantic relationships in free text. Some of the conclusions that can be drawn from this work are the following:

- A new algorithm for generalising lexical patterns has been described, implemented and evaluated. It is based on the edit distance algorithm, which has been modified to take into account the part-of-speech tags of the words. This algorithm is fully automatic, as it requires no human supervision.
- The set of patterns which has been found automatically from the Wikipedia entries, is able to extract new relations from text for each of the four relationships: hyperonymy, hyponymy, meronymy and holonymy. More than 1200 new relationships have been provided.
- The precision of the generated patterns is similar to that of patterns written *by hand* (although they are not comparable, as the experimental settings differ). The kind of hyponymy lexicosyntactic patterns as described by [16] were evaluated, in different settings, by [20] and [10], who report a precision of 0.65 and 0.39, respectively. [18] reports a 0.55 accuracy for a set of patterns that identify holonyms. Only [19] reports much higher accuracies (0.72, 0.92 and 0.93), when identifying relationships of merging between companies.

This work opens the following research lines: (a) to extract other kinds of relations, such as *location, instrument, telic* or *author*; (b) to generalise the experiment to other ontologies and encyclopedias, and even to apply it to fully unrestricted texts; and (c) to extend the formalism used to represent the patterns, so they can encode syntactic features as well.

## References

1. Ding, Y., Fensel, D., Klein, M.C.A., Omelayenko, B.: The semantic web: yet another hip? Data Knowledge Engineering **41** (2002) 205–227
2. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web - a new form of web content that is meaningful to computers will unleash a revolution of new possibilities. Scientific American **284** (2001) 34–43

3. Gruber, T.R.: A translation approach to portable ontologies. Knowledge Acquisition **5** (1993) 199–220
4. Degen, W., Heller, B., Herre, H., Smith, B.: Gol: Towards an axiomatized upper-level ontology. In: Proceedings of the International Conference on Formal Ontology in Information Systems, FOIS-2001. (2001)
5. Gómez-Pérez, A., Macho, D.M., Alfonseca, E., nez, R.N., Blascoe, I., Staab, S., Corcho, O., Ding, Y., Paralic, J., Troncy, R.: Ontoweb deliverable 1.5: A survey of ontology learning methods and techniques (2003)
6. Maedche, A., Staab, S.: Ontology learning for the semantic web. IEEE Intelligent systems **16** (2001)
7. Miller, G.A.: WordNet: A lexical database for English. Communications of the ACM **38** (1995) 39–41
8. Lee, L.: Similarity-Based Approaches to Natural Language Processing. Ph.D. thesis. Harvard University Technical Report TR-11-97 (1997)
9. Faure, D., Nédellec, C.: A corpus-based conceptual clustering method for verb frames and ontology acquisition. In: LREC workshop on Adapting lexical and corpus resources to sublanguages and applications, Granada, Spain (1998)
10. Cimiano, P., Staab, S.: Clustering concept hierarchies from text. In: Proceedings of LREC-2004. (2004)
11. Hastings, P.M.: Automatic acquisition of word meaning from context. University of Michigan, Ph. D. Dissertation (1994)
12. Hahn, U., Schnattinger, K.: Towards text knowledge engineering. In: AAAI/IAAI. (1998) 524–531
13. Pekar, V., Staab, S.: Word classification based on combined measures of distributional and semantic similarity. In: Proceedings of Research Notes of the 10th Conference of the European Chapter of the Association for Computational Linguistics, Budapest (2003)
14. Alfonseca, E., Manandhar, S.: Extending a lexical ontology by a combination of distributional semantics signatures. In: Knowledge Engineering and Knowledge Management. Volume 2473 of Lecture Notes in Artificial Intelligence. Springer Verlag (2002) 1–7
15. Maedche, A., Staab, S.: Discovering conceptual relations from text. In: Proceedings of the 14th European Conference on Artifial Intelligence. (2000)
16. Hearst, M.A.: Automatic acquisition of hyponyms from large text corpora. In: Proceedings of COLING-92, Nantes, France (1992)
17. Hearst, M.A. In: Automated Discovery of WordNet Relations. In Christiane Fellbaum (Ed.) WordNet: An Electronic Lexical Database. MIT Press (1998) 132–152
18. Berland, M., Charniak, E.: Finding parts in very large corpora. In: Proceedings of ACL-99. (1999)
19. Finkelstein-Landau, M., Morin, E.: Extracting semantic relationships between terms: supervised vs. unsupervised methods. In: Proceedings of the International Workshop on Ontologial Engineering on the Global Information Infrastructure. (1999)
20. Kietz, J., Maedche, A., Volz, R.: A method for semi-automatic ontology acquisition from a corporate intranet. In: Workshop "Ontologies and text", co-located with EKAW'2000, Juan-les-Pins, French Riviera (2000)
21. Alfonseca, E., Manandhar, S.: Improving an ontology refinement method with hyponymy patterns. In: Language Resources and Evaluation (LREC-2002), Las Palmas (2002)
22. Navigli, R., Velardi, P.: Learning domain ontologies from document warehouses and dedicated websites. Computational Linguistics **30** (2004)
23. Wilks, Y., Fass, D.C., Guo, C.M., McDonald, J.E., Plate, T., Slator, B.M.: Providing machine tractable dictionary tools. Journal of Computers and Translation (1990)
24. Rigau, G.: Automatic Acquisition of Lexical Knowledge from MRDs. PhD Thesis, Departament de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya (1998)

25. Richardson, S.D., Dolan, W.B., Vanderwende, L.: MindNet: acquiring and structuring semantic information from text. In: Proceedings of COLING-ACL'98. Volume 2., Montreal, Canada (1998) 1098–1102
26. Harabagiu, S., Moldovan, D.I.: Knowledge processing on an extended wordnet. In: WordNet: An Electronic Lexical Database. MIT Press (1998) 379–405
27. Harabagiu, S., Miller, G., Moldovan, D.: Wordnet 2 - a morphologically and semantically enhanced resource. In: Proc. of the SIGLEX Workshop on Multilingual Lexicons, ACL Annual Meeting, University of Maryland (1999)
28. Novischi, A.: Accurate semantic annotation via pattern matching. In: Proceedings of FLAIRS-2002. (2002)
29. DeBoni, M., Manandhar, S.: Automated discovery of telic relations for wordnet. In: Poceedings of the First International Conference on General WordNet, Mysore, India (2002)
30. Alfonseca, E.: Wraetlic user guide version 1.0 (2003)
31. Ruiz-Casado, M., Alfonseca, E., Castells, P.: Automatic assignment of wikipedia encyclopedic entries to wordnet synsets. In: press. (2005)
32. Marcus, M.P., Santorini, B., Marcinkiewicz, M.A.: Building a large annotated corpus of english: the penn treebank. Computational Linguistics **19** (1993) 313–330
33. Wagner, R., Fischer, M.: The string-to-string correction problem. Journal of Assoc. Comput. Mach. **21** (1974)
34. Alfonseca, E., Manandhar, S.: Distinguishing instances and concepts in wordnet. In: Poceedings of the First International Conference on General WordNet, Mysore, India (2002)

# Combining Data-Driven Systems for Improving Named Entity Recognition

Zornitsa Kozareva, Oscar Ferrández, Andres Montoyo, Rafael Muñoz, and Armando Suárez

Departamento de Lenguajes y Sistemas Informáticos
Universidad de Alicante
{zkozareva,ofe,montoyo,rafael,armando}@dlsi.ua.es

**Abstract.** The increasing flow of digital information requires the extraction, filtering and classification of pertinent information from large volumes of texts. An important preprocessing tool of these tasks consists of name entities recognition, which corresponds to a Name Entity Recognition (NER) task. In this paper we propose a completely automatic NER which involves identification of proper names in texts, and classification into a set of predefined categories of interest as Person names, Organizations (companies, government organizations, committees, etc.) and Locations (cities, countries, rivers, etc). We examined the differences in language models learned by different data-driven systems performing the same NLP tasks and how they can be exploited to yield a higher accuracy than the best individual system. Three NE classifiers (Hidden Markov Models, Maximum Entropy and Memory-based learner) are trained on the same corpus data and after comparison their outputs are combined using voting strategy. Results are encouraging since 98.5% accuracy for recognition and 84.94% accuracy for classification of NE for Spanish language were achieved.

## 1 Introduction

The vision of the information society as a global digital community is fast becoming a reality. Progress is being driven by innovation in business and technology, and the convergence of computing, telecommunications and information systems. Access to knowledge resources in the information society is vital to both our professional and personal development. However, access alone is not enough. We need to be able to select, classify, assimilate, retrieval, filter and exploit this information, in order to enrich our collective and individual knowledge and skills. This is a key area of application for language technologies. The approach taken in this area is to develop advanced applications characterized by more intuitive natural language interfaces and content-based information analysis, extraction and filtering. Natural Language Processing (NLP) is crucial in solving these tasks. In concrete, Name Entity Recognition (NER) has emerged as an important preprocessing tool for many NLP applications as Information Extraction, Information Retrieval and other text processing applications. NER

A. Montoyo et al. (Eds.): NLDB 2005, LNCS 3513, pp. 80–90, 2005.
© Springer-Verlag Berlin Heidelberg 2005

involves processing a text and identifying certain occurrences of words or expressions as belonging to a particular category of Named Entities (NEs) as person, organization, location, etc.

This paper describes a multiple voting method that effectively combines strong classifiers such as Hidden Markov Models, Maximum Entropy and Memory-based learner for the NE recognition and classification tasks for Spanish texts. Two different approaches have been developed by the researchers in order to solve the Named Entity Recognition task. The former approach is based on Machine Learning methods, such us Hidden Markov's Models, Maximum Entropy, Support Vector Machine or Memory-based. This approach uses a set of features providing information about the context (previous and posterior words), orthographic features (capital letter, etc.), semantic information, morphological information, etc. in order to provide statistical classification. The latter approach is based on Knowledge-based techniques. This approach uses a set of rules to implement a specific grammar for named entity and set of databases or gazetteer to look for specific words like names of people or locations. List of names or gazetteer can be also used in future for machine learning method.

Different systems have been developed for each approach, we emphasize two conferences: CoNLL[1] and ACE[2], and several systems achieving good scores. We point out two knowledge-based systems like Maynard et al. [6], Arevalo et al. [1] and several machine learning systems like Carreras et al. [2], Mayfield et al. [5], Florian et al. [4], etc.

The organization of this paper is as following: After this introduction, the features used by our classifiers are listed in Section 2; the sheer classifiers are detailed in Section 3. The different experiments and obtained results are examined and discussed in Section 4. The voting strategy we used is in Section 5 and finally we conclude (Section 6) with a summary of the most important observations and future work.

## 2 Description of Features

The Maximum Entropy and Memory-based learning classifiers we used for the NE tasks (detection and classification) utilize the identical features described below. In contrast HMM doesn't take any features because it depends on the probability of the NE and the tag associated with it. To gain better results we studied different feature combinations from the original set.

### 2.1 Features for NE Detection

For NE detection, we use the well-known BIO model, where a tag shows that a word is at the beginning of a NE (B), inside a NE (I) or outside a NE (O). For the sentence "Juan Carlos está esperando", the following tags have been

[1] http://cnts.uia.ac.be/conll2002/ner/
[2] http://www.nist.gov/speech/tests/ace/

- **a**: anchor word (e.g. the word to be classified)
- ***c[1-6]***: context of the word at position ±1, ±2, ±3
- ***C[1-7]***: capitalization of the word at position 0, ±1, ±2, ±3
- ***d[1-3]***: word +1,+2,+3 in dictionary of entities
- ***p***: position of anchor word in the sentence

**Fig. 1.** Features for NE detection.

associated, "B I O O O", where *Juan* starts a named entity; *Carlos* continues this entity and neither *está* nor *esperando* or the full stop are part of a NE.

Our BIO model uses a set composed of 18 features as described in Figure 1. They represent words, position in the sentence and entity triggers for each NE.

### 2.2 Features for NE Classification

The tags used for NE classification are PER, LOC, ORG and MISC as defined by CoNLL-2002 task. Their detection is possible by the help of the first seven features used by out BIO model (e.g. a, c[1-6], p) and the additional set described below in Figure 2. In Section 4 several experiments were made by shortening the original set into one containing the most informative ones and their influence upon system's performance is discussed.

- ***eP***: entity is trigger PER
- ***eL***: entity is trigger LOC
- ***eO***: entity is trigger ORG
- ***eM***: entity is trigger MISC
- ***tP***: word ±1 is trigger PER
- ***tL***: word ±1 is trigger LOC
- ***tO***: word ±1 is trigger ORG
- ***gP***: part of NE is in database or gazzetters for PER
- ***gL***: part of NE is in database or gazzetters for LOC
- ***gO***: part of NE is in database or gazzetters for ORG
- ***wP***: whole entity is PER
- ***wL***: whole entity is LOC
- ***wO***: whole entity is ORG
- ***NoE***: whole entity is not in the defined three classes
- ***f***: first word of the entity
- ***s***: second word of the entity

**Fig. 2.** Features for NE classification.

## 3 Classification Methods

We have used three classification methods, in concrete Memory-based learner, Maximum Entropy and HMM for the NE detection and classification tasks. Next subsections describe each method individually.

### 3.1 Memory-Based Learner

Memory-based learning is a supervised inductive learning algorithm for learning classification tasks. It treats a set of training instances as points in a multi-

dimensional feature space, and stores them as such in an instance base in memory. Test instances are classified by matching them to all instances in memory, and by calculating with each match the distance, given by a distance function between the new instance $x$ and each of the $n$ memory instances $y_1 \ldots_n$. Classification in memory-based learning is performed by the $k - NN$ algorithm that searches for the $k$ 'nearest neighbours' among the memory instances according to the distance function. The majority class of the $k$ nearest neighbors then determines the class of the new instance $x$. [3]. The memory-based software package we used is called Timbl [3]. Its default learning algorithm, instance-based learning with information gain weighting (IB1IG) was applied.

### 3.2 Maximum Entropy

The maximum entropy framework estimates probabilities based on the principle of making as few assumptions as possible, other than the constraints imposed. The probability distribution that satisfies the above property is the one with the highest entropy [7]. A classifier obtained by means of a ME technique consists of a set of parameters or coefficients which are estimated using an optimization procedure. Each coefficient is associated with one feature observed in the training data. The main purpose is to obtain the probability distribution that maximizes the entropy. Some advantages of using the ME framework are that even knowledge-poor features may be applied accurately; the ME framework thus allows a virtually unrestricted ability to represent problem-specific knowledge in the form of features.

$$f(x,c) = \begin{cases} 1 & if\ \text{c'=c\&cp(x)=true} \\ 0 & \text{otherwise} \end{cases} . \tag{1}$$

$$p(c|x) = \frac{1}{Z(x)} \prod_{i=1}^{K} \alpha_i^{f_i(x,c)} \tag{2}$$

The implementation of ME was done in C++[10] and the features used for testing are described in the section above. The implementation we used is a very basic one because no smoothing nor feature selection is performed.

### 3.3 Hidden Markov Models

Hidden Markov Models are stochastic finite-state automata with probabilities for the transitions between states and for the emission of symbols from states. The Viterbi algorithm is often used to find the most likely sequence of states for a given sequence of output symbols. In our case, let $T$ be defined as set of all tags, and $\sum$ the set of all NEs. One is given a sequence of NEs $W = w_1 \ldots w_k \in \sum^*$ , and is looking for a sequence of tags $T = t_1 \ldots t_k \in T^*$ that maximizes the conditional probability $p(T|W)$, hence we are looking for

$$arg \max_T p(T|W) = arg \max_T \frac{p(T)p(W|T)}{p(W)} \tag{3}$$

p(W) is independent of the chosen tag sequence, thus it is sufficient to find

$$arg \max_{T} p(T)p(W|T). \tag{4}$$

The toolkit we used is called ICOPOST[3] implemented for POS tagging purpose and adapted for NER [9].

## 4 Experiments and Discussion

Our NER system has two passages

1. detection : identification of sequence of words that make up the name of an entity.

2. classification : deciding to which category our previously recognized entity should belong.

The Spanish train and test data we used are part of the CoNLL-2002 [8] corpus. For training we had corpus containing 264715 tokens and 18794 entities and for testing we used Test-B corpus with 51533 tokens and 3558 entities. Scores were computed per NE class and the measures used are Precision (of the tags allocated by the system, how many were right), Recall (of the tags the system should have found, how many did it spot) and $F_{\beta=1}$(a combination of recall and precision). To calculate precision and recall for all tags in the system, Accuracy is used as Precision and Recall coincide (e.g. all NEs have a tag, and there is no case in which an entity has no class ).

$$Precision = \frac{number\,of\,correct\,answers\,found\,by\,the\,system}{number\,of\,answers\,given\,by\,the\,system} \tag{5}$$

$$Recall = \frac{number\,of\,correct\,answers\,found\,by\,the\,system}{number\,of\,correct\,answers\,in\,the\,test\,corpus} \tag{6}$$

$$F_{\beta=1} = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{7}$$

$$Accuracy = \frac{correctly\,classified\,tags}{total\,number\,of\,tags\,in\,the\,test\,corpus} \tag{8}$$

### 4.1 Recognition by BIO Model

During NE detection, Timbl and ME classifiers follow the BIO model described briefly in subsection 2.1, using the set of 18 features described in Figure 1 while HMM takes only the NE and the tag associated with it. Systems' performance can be observed in Table 1. For clearance of result calculation, we put in abbreviations the various tag combinations (B:B, B:I, B:O, etc.) and their values (column N). The first letter always points to the class the NE has in reality and the second one shows the class predicted by the classifier. For the case of B tags, B:I signifies that the NE is supposed to be B, but the classifier assigned an I tag

[3] http://acopost.sourceforge.net/

to it and for B:O the system put an O for an entity that is supposed to be B. The same holds for the other abbreviations. If a confusion matrix is built using the values in column "N" for each tag, the calculation of precision and recall can be obtained easily [3].

**Table 1.** BIO detection.

| | | Timbl(%) | | | | HMM(%) | | | | Maximum Entropy(%) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | N | Prec. | Rec. | $F_{\beta=1}$ | N | Prec. | Rec. | $F_{\beta=1}$ | N | Prec. | Rec. | $F_{\beta=1}$ |
| B | B:B | 3344 | 93.59 | 93.99 | 93.79 | 3262 | 90.14 | 91.68 | 90.90 | 1060 | 85.42 | 29.79 | 44.18 |
| | B:I | 88 | | | | 148 | | | | 54 | | | |
| | B:O | 126 | | | | 148 | | | | 2444 | | | |
| I | I:I | 2263 | 91.18 | 86.37 | 88.71 | 2013 | 87.52 | 76.83 | 81.83 | 673 | 81.48 | 25.69 | 39.06 |
| | I:B | 157 | | | | 246 | | | | 127 | | | |
| | I:O | 200 | | | | 361 | | | | 1820 | | | |
| O | O:O | 45152 | 99.28 | 99.55 | 99.42 | 45105 | 98.88 | 99.45 | 99.17 | 45202 | 91.38 | 99.66 | 95.34 |
| | O:B | 72 | | | | 111 | | | | 54 | | | |
| | O:I | 131 | | | | 139 | | | | 99 | | | |
| Accuracy | | 98.50 | | | | 97.76 | | | | 91.08 | | | |
| Only B&I | | precBI=93.16<br>recBI=90.76<br>$F_{\beta=1}BI$=92.27 | | | | precBI=89.12<br>recBI=85.38<br>$F_{\beta=1}BI$=87.21 | | | | precBI=83.84<br>recBI=28.05<br>$F_{\beta=1}BI$=42.04 | | | |

The coverage of tag O is high due to its frequent appearance, however its importance is not so significant as the one of B and I tags. For this reason we calculated separately system's precision, recall and F-measure for B and I tags together. The best score was obtained by the memory-based system Timbl with F-measure of 92.27%.

As a whole system's performance is calculated considering all BIO tags and the highest score of 98.50% Accuracy is achieved by Timbl. After error analysis we discovered that results can be improved with simple post-processing where in the case of I tag preceded by O tag we have to substitute it by B if the analyzed word starts with a capital letter and in the other case we simply have to put O (see the example in subsection 2.1). With post-processing Timbl raised its Accuracy to 98.61% , HMM to 97.96% and only ME lowered its score to 90.98%. We noticed that during ME's classifictaion occured errors in the O I I sequences which normally should be detected as B I I. The post-processing turns the O I I sequence into O B I which obviously is erroneous. First error is due to the classificator's assignement of an I tag after an O and the second one is comming from the post-processor's substitution of the first I tag in the O I I sequence by B. This errors lowered ME's performance.

### 4.2 Classification into Classes

After detection follows NE classification into LOC, MISC, ORG or PER class. Again to HMM model we passed the NE and its class. The achieved accuracy

is 74.37% and in Table 2 can be noticed that LOC, ORG and PER classes have good coverage while MISC has only 57.84% due to its generality.

**Table 2.** HMM Classifier.

| Class | Prec.% | Rec.% | $F_{\beta=1}$% |
|---|---|---|---|
| LOC | 80.02 | 73.16 | 76.43 |
| MISC | 56.46 | 59.29 | 57.84 |
| ORG | 77.60 | 77.71 | 77.66 |
| PER | 69.72 | 76.74 | 73.06 |
| Accuracy | 74.37% | | |

For the same task the other two systems used the set of features described in subsection 2.2. Initially we made experiments with a bigger set composed of 37 features extracted and collected from articles of people researching in the same area. They include the following 18 attributes: and the denoted in Figure 1 and 2 features: a, c[1-6], p, eP, eL, eO, eM, gP, gL, gO, wP, wL, wO, NoE.

- ***wtL[1-2]***: word ±1 is trigger LOC
- ***wtO[1-2]***: word ±1 is trigger ORG
- ***wtP[1-2]***: word ±1 is trigger PER
- ***wtL[1-2]***: word ±2 is trigger LOC
- ***wtO[1-2]***: word ±2 is trigger ORG
- ***wtP[1-2]***: word ±2 is trigger PER
- ***wtL[1-2]***: word ±3 is trigger LOC
- ***wtO[1-2]***: word ±3 is trigger ORG
- ***wtP[1-2]***: word ±3 is trigger PER

**Fig. 3.** Features for NE detection.

The obtained results from these attributes are in Table 3. Their accuracy is better than the one of HMM but still not satisfactory. Then we decided to investigate the most substantial ones, to remove the less significant and to include two more attributes. Thus our NE classification set became composed of 24 features as described above in subsection 2.2. Let us denote by $A$ the set of features: a, c[1-6], p, eP, eL, eO, eM, tP, tL, tO, gP, gL, gO, wP, wL, wO, NoE, f and s.

**Table 3.** Timbl and ME using 37 features.

| Class | *Timbl* | | | *Maximum entropy* | | |
|---|---|---|---|---|---|---|
| | Prec.% | Rec.% | $F_{\beta=1}$% | Prec.% | Rec.% | $F_{\beta=1}$% |
| LOC | 80.23 | 76.38 | 78.26 | 81.07 | 74.26 | 77.52 |
| MISC | 51.10 | 48.08 | 49.54 | 78.95 | 39.82 | 52.94 |
| ORG | 77.94 | 82.5 | 80.15 | 73.06 | 86.57 | 79.24 |
| PER | 83.17 | 82.04 | 82.60 | 78.64 | 78.64 | 78.64 |
| Accuracy | 77.26% | | | 76.73% | | |

In Table 4 are shown the results of Timbl and ME using set $A$. Their accuracy has been higher than the one of HMM, however ME performed better than Timbl. The memory-based learner works by measuring distance among elements which made us select and test as a second step only the most relevant and informative features.

**Table 4.** Timbl and ME using set $A$.

| Class | *Timbl* | | | *Maximum entropy* | | |
|---|---|---|---|---|---|---|
| | Prec.% | Rec.% | $F_{\beta=1}$% | Prec.% | Rec.% | $F_{\beta=1}$% |
| LOC | 82.02 | 80.81 | 81.41 | 88.25 | 81.09 | 84.52 |
| MISC | 64.83 | 61.95 | 63.35 | 85.65 | 58.11 | 69.24 |
| ORG | 81.61 | 84.93 | 83.23 | 80.66 | 90.29 | 85.20 |
| PER | 88.29 | 85.17 | 86.70 | 86.93 | 90.48 | 88.67 |
| Accuracy | 81.54% | | | 84.46% | | |

Let denote by $B$ the set with the most informative attributes which is a subset of our original set $A$: a, c[1], eP, gP, gL, gO, wP, wL, wO, NoE and f. Table 5 shows the results with the reduced set (e.g. $B$). It can be notices that Timbl increased its performance to 83.81% but ME lower it to 82.24%, because even knowledge-poor features are important and can be applied accurately to it. Timbl classified MISC class better with 0.35% than ME when using the whole feature set and got the higher coverage for this class among our classifiers.

**Table 5.** Timbl and ME using set $B$.

| Class | *Timbl* | | | *Maximum entropy* | | |
|---|---|---|---|---|---|---|
| | Prec. % | Rec. % | $F_{\beta=1}$ % | Prec. % | Rec. % | $F_{\beta=1}$ % |
| LOC | 85.35 | 82.20 | 83.74 | 86.33 | 79.24 | 82.64 |
| MISC | 77.54 | 63.13 | 69.59 | 91.19 | 51.92 | 66.17 |
| ORG | 83.92 | 86.50 | 85.19 | 74.64 | 93.36 | 82.96 |
| PER | 83.77 | 90.61 | 87.06 | 94.35 | 79.46 | 86.26 |
| Accuracy | 83.81% | | | 82.24% | | |

On principle our systems perform well with LOC, ORG and PER classes as it can be noticed in Tables 4 and 5, but face difficulties detecting MISC class who can refer to anything from movie titles to sports events, etc.

## 5 Classifier Combination

It is a well-known fact that if several classifiers are available, they can be combined in various ways to create a system that outperforms the best individual classifier. Since we had several classifiers available, it was reasonable to investigate combining them in different ways. The simplest approach to combining classifiers is through voting, which examines the outputs of the various models and selects the classifications which have a weight exceeding some threshold, where the weight is dependent upon the models that proposed this particular

classification. It is possible to assign various weights to the models, in effect giving one model more importance than the others. In our system, however, we simply assigned to each model equal weight, and selected classifications which were proposed by a majority of models. Voting was thus used to improve further the base model.

In the three separate votings we made a combinations of HMM and the feature set variations of ME and Timbl. In Table 6 voting's results per LOC, MISC and ORG classes are higher in comparison with the one of HMM and Timbl but still lower than ME. PER's score is greater than each individual system and voting's accuracy is only less than the one of ME.

As discussed in Section 4, the reduced set $B$ covers MISC class better than ME, so the second voting was among HMM and the classifiers' reduced set $B$. In Table 7 for LOC, MISC and PER class voting performs better among the three classifiers and only Timbl has greater coverage with ORG class. Compared to the accuracy of each individual system, the reached 83.95% score is the higher one.

The voting (Table 8) where best performing systems have participated reached 84.15% F-measure for LOC and 84.86% for ORG classes which is higher than the individual performance of HMM for the same classes but less than the results obtained by Timbl and ME. For MISC 71.50% F-measure is achieved, the highest score in comparison not only with a system individually but also with the other votings (Table 6 and 7) we had.

In conclusion applying voting among the best performing systems raised accuracy with 0.11% and led to 71.50% classification of MISC class which is particularly difficult due to its generality as discussed before.

For CoNLL-2002 Carreras[2](Carr.) gained the best score for NE classification. In Table 9 we show our voting results together with the one obtained by their system. It can be seen that we managed to improve classification for each one of the LOC, MISC, ORG and PER classes. Our third voting system im-

**Table 6.** Voting among Timbl, ME using set $A$ and HMM.

| | LOC(%) | | | MISC(%) | | | ORG(%) | | | PER(%) | | | All |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Classif. | Prec. | Rec. | $F_{\beta=1}$ | Prec. | Rec. | $F_{\beta=1}$ | Prec. | Rec. | $F_{\beta=1}$ | Prec. | Rec. | $F_{\beta=1}$ | Accur. |
| Timbl | 82.02 | 80.81 | 81.41 | 64.83 | 61.95 | 63.35 | 81.61 | 84.93 | 83.23 | 88.29 | 85.17 | 86.70 | 81.54 |
| ME | 88.25 | 81.09 | 84.52 | 85.65 | 58.11 | 69.24 | 80.66 | 90.29 | 85.20 | 86.93 | 90.48 | 88.67 | 84.46 |
| HMM | 80.02 | 73.16 | 76.43 | 56.46 | 59.29 | 57.84 | 77.60 | 77.71 | 77.66 | 69.72 | 76.74 | 73.06 | 74.37 |
| Vot 1 | 86.01 | 81.64 | 83.77 | 82.98 | 57.52 | 67.94 | 79.71 | 89.21 | 84.19 | 89.68 | 88.71 | 89.19 | 83.78 |

**Table 7.** Voting among Timbl, ME using set $B$ and HMM.

| | LOC(%) | | | MISC(%) | | | ORG(%) | | | PER(%) | | | All |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Classif. | Prec. | Rec. | $F_{\beta=1}$ | Prec. | Rec. | $F_{\beta=1}$ | Prec. | Rec. | $F_{\beta=1}$ | Prec. | Rec. | $F_{\beta=1}$ | Accur. |
| Timbl | 85.35 | 82.20 | 83.74 | 77.54 | 63.13 | 69.59 | 83.92 | 86.5 | 85.19 | 83.77 | 90.61 | 87.06 | 83.81 |
| ME | 86.33 | 79.24 | 82.64 | 91.19 | 51.92 | 66.17 | 74.64 | 93.36 | 82.96 | 94.35 | 79.46 | 86.26 | 82.24 |
| HMM | 80.02 | 73.16 | 76.43 | 56.46 | 59.29 | 57.84 | 77.60 | 77.71 | 77.66 | 69.72 | 76.74 | 73.06 | 74.37 |
| Vot 2 | 87.00 | 80.90 | 83.84 | 88.0 | 58.41 | 70.21 | 78.71 | 90.07 | 84.01 | 90.04 | 88.57 | 89.30 | 83.95 |

**Table 8.** Voting among Timbl using set $B$, ME using set $A$ and HMM.

| | LOC(%) | | | MISC(%) | | | ORG(%) | | | PER(%) | | | All |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Classif. | Prec. | Rec. | $F_{\beta=1}$ | Prec. | Rec. | $F_{\beta=1}$ | Prec. | Rec. | $F_{\beta=1}$ | Prec. | Rec. | $F_{\beta=1}$ | Accur. |
| Timbl | 85.35 | 82.20 | 83.74 | 77.54 | 63.13 | 69.59 | 83.92 | 86.5 | 85.19 | 83.77 | 90.61 | 87.06 | 83.81 |
| ME | 88.25 | 81.09 | 84.52 | 85.65 | 58.11 | 69.24 | 80.66 | 90.29 | 85.20 | 86.93 | 90.48 | 88.67 | 84.46 |
| HMM | 80.02 | 73.16 | 76.43 | 56.46 | 59.29 | 57.84 | 77.60 | 77.71 | 77.66 | 69.72 | 76.74 | 73.06 | 74.37 |
| Vot 3 | 86.92 | 81.55 | 84.15 | 86.25 | 61.06 | 71.50 | 81.51 | 88.5 | 84.86 | 86.94 | 92.38 | 89.58 | 84.57 |

**Table 9.** Comparing voting 1,2,3 with the results of Careras.

| | LOC(%) | | | MISC(%) | | | ORG(%) | | | PER(%) | | | All |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Classif. | Prec. | Rec. | $F_{\beta=1}$ | Prec. | Rec. | $F_{\beta=1}$ | Prec. | Rec. | $F_{\beta=1}$ | Prec. | Rec. | $F_{\beta=1}$ | Accur. |
| Vot 1 | 86.01 | 81.64 | 83.77 | 82.98 | 57.52 | 67.94 | 79.71 | 89.21 | 84.19 | 89.68 | 88.71 | 89.19 | 83.78 |
| Vot 2 | 87.00 | 80.90 | 83.84 | 88.0 | 58.41 | 70.21 | 78.71 | 90.07 | 84.01 | 90.04 | 88.57 | 89.30 | 83.95 |
| Vot 3 | 86.92 | 81.55 | 84.15 | 86.25 | 61.06 | 71.50 | 81.51 | 88.5 | 84.86 | 86.94 | 92.38 | 89.58 | 84.57 |
| Carr. | 85.76 | 79.43 | 82.47 | 60.19 | 57.35 | 58.73 | 81.21 | 82.43 | 81.81 | 84.71 | 93.47 | 88.87 | 81.39 |

proved F-measure with 1.68% LOC, 12.77% MISC, 3.05% ORG and 0.71% PER class classification. Each voting observed separately has higher F-measure than the one obtained by Carreras.

## 6 Conclusions and Future Work

In this paper we present a combination of three different Named Entity Recognition systems based on machine learning approaches applied to Spanish texts. Every named entity system we have introduced is using the same set or subset of features over the same training corpus for tuning the system and the same test corpus for evaluating it. Three different combinations have been developed in order to improve the score of each individual system. The results are encouraging since 98.50% overall accuracy was obtained for NE detection using BIO model and 84.94% for NE classification into LOC, ORG, PER and MISC classes. However, Maximum Entropy as individual system performs better with NE classification into LOC and ORG classes. As a whole we need to improve our scores by adding more information using new features. For future work we also intend to include morphological and semantic information, to develop a more sophisticated voting system and to adapt our system for other languages.

## Acknowledgements

This research has been partially funded by the Spanish Government under project CICyT number TIC2000-0664-C02-02 and PROFIT number FIT-340100-2004-14 and by the Valencia Government under project numbers GV04B-276 and GV04B-268.

## References


1. Montserrat Arevalo, Montserrat Civit, and Maria Antonia Martí. MICE: A module for Named Entity Recognition and Clasification. *International Journal of Corpus Linguistics*, 9(1):53 – 68, March 2004.
2. Xavier Carreras, Lluís Màrques, and Lluís Padró. Named entity extraction using adaboost. In *Proceedings of CoNLL-2002*, pages 167–170. Taipei, Taiwan, 2002.
3. Walter Daelemans, Jakub Zavrel, Ko van der Sloot, and Antal van den Bosch. TiMBL: Tilburg Memory-Based Learner. Technical Report ILK 03-10, Tilburg University, November 2003.
4. Radu Florian, Abe Ittycheriah, Hongyan Jing, and Tong Zhang. Named entity recognition through classifier combination. In Walter Daelemans and Miles Osborne, editors, *Proceedings of CoNLL-2003*, pages 168–171. Edmonton, Canada, 2003.
5. James Mayfield, Paul McNamee, and Christine Piatko. Named entity recognition using hundreds of thousands of features. In Walter Daelemans and Miles Osborne, editors, *Proceedings of CoNLL-2003*, pages 184–187. Edmonton, Canada, 2003.
6. Diana Maynard, Valentin Tablan, Cristian Ursu, Hamish Cunningham, and Yorick Wilks. Named Entity Recognition from Diverse Text Types. In R. Mitkov N. Nicolov G. Angelova, K. Bontcheva and N. Nikolov, editors, *Proceedings of the Recent Advances in Natural Language Processing*, Tzigov Chark, 2001.
7. Adwait Ratnaparkhi. *Maximum Entropy Models For Natural Language Ambiguity Resolution*. PhD thesis, Computer and Information Science Department, University of Pennsylvania, 1998.
8. Tijong Kim Sang. Introduction to the conll-2002 shared task: Language independent named entity recognition. In *Proceedings of CoNLL-2002*, pages 155–158, 2002.
9. Ingo Schröder. A case study in part-of-speech tagging using the icopost toolkit. Technical Report FBI-HH-M-314/02, Department of Computer Science, University of Hamburg, 2002.
10. Armando Suárez and Manuel Palomar. A maximum entropy-based word sense disambiguation system. In Hsin-Hsi Chen and Chin-Yew Lin, editors, *Proceedings of the 19th International Conference on Computational Linguistics, COLING 2002*, pages 960–966, August 2002.

# Natural Language Processing: Mature Enough for Requirements Documents Analysis?

Leonid Kof

Technische Universität München, Fakultät für Informatik,
Boltzmannstr. 3, D-85748 Garching bei München, Germany
kof@in.tum.de

**Abstract.** Requirements engineering is the Achilles' heel of the whole software development process, because requirements documents are often inconsistent and incomplete. Misunderstandings and errors of the requirements engineering phase propagate to later development phases and can potentially lead to a project failure.

A promising way to overcome misunderstandings is to extract and validate terms used in requirements documents and relations between these terms. This position paper gives an overview of the existing terminology extraction methods and shows how they can be integrated to reach a comprehensive text analysis approach. It shows how the integrated method would both detect inconsistencies in the requirements document and extract an ontology after elimination of inconsistencies. This integrated method would be more reliable than every of its single constituents.

## 1 Ontology as a Requirements Engineering Product

Requirements engineering is the very first stage of any software project. This stage is extremely important, because requirements engineering should ensure that the specification of the product to be built meets customer's wishes. (Are we building the *right* product?) The goal of the later development stages, to the contrary, is to ensure that a product is being built correctly *with respect to the specification*, produced in the RE phase. So, requirements engineering errors either potentially lead to project failure or must be corrected in later phases, which is much more expensive than correction in the RE phase.

It is often believed that RE errors are due to forgotten or misinterpreted requirements. Praxis shows, however, that misunderstandings come into play much earlier: the same seemingly unambiguous word used in the requirements document can be interpreted in different ways. Obviously, misinterpretation of concept meanings is fatal for requirements engineering.

Zave and Jackson [1] give an example of such a concept misinterpretation. This example handles a hypothetical university information system and the definitions of a "student" and the binary relation "enrolled" for this system:

A. Montoyo et al. (Eds.): NLDB 2005, LNCS 3513, pp. 91–102, 2005.
© Springer-Verlag Berlin Heidelberg 2005

Able: Two important basic types are *student* and *course*. There is also a binary relation *enrolled*. If types and relations are formalized as predicates, then

$$\forall s\ \forall c\ (enrolled(s, c) \Rightarrow student(s) \wedge course(c)).$$

Baker: Do only students enroll in courses? I don't think that's true.
Able: But that's what I mean by *student*!

This example shows that domain concepts must be precisely defined and that a simple glossary (term list) is insufficient. A more appropriate definition of domain concepts and relations between them would be an *ontology*.

According to Tom Gruber an ontology is a *specification of a conceptualization*[1]. In the context of this paper the ontology is defined as a taxonomy (term hierarchy), enriched by some general associations. The taxonomy itself consists of a set of terms and the "is-a"–relation.

The goal of this paper is to propose a comprehensive ontology extraction approach, based on existing methods. This proposed approach would both detect terminology inconsistencies in documents and extract an ontology from requirements documents. The paper shows how existing methods can be used in different stages of ontology building and how they would augment each other when integrated.

The paper is organized in the following way: Section 2 introduces ontology construction steps, without concrete recipes for particular steps. Section 3 presents an ontology construction approach implementing the steps introduced in Section 2. This approach was proven feasible in case studies. However, there is always room for improvement. Section 4 gives an overview of other existing text analysis approaches in RE and sketches their possible place in the ontology construction process. Section 5 integrates the text analysis methods into the comprehensive ontology building process, providing both detection of terminology inconsistencies and ontology extraction. Section 6 summarizes the integrated proposal introduced in Section 5.

## 2 Ontology Construction Steps

Ontology was introduced in artificial intelligence as a communication means for intelligent agents. Now it was recognized as a universal means to communicate concept dependencies. As software development involves experts from different disciplines, an ontology is also a good means to establish a common language in a software project.

Breitman and Sampaio do Prado Leite [2] see an application ontology as one of the products of the requirements engineering activity. They list in their paper several ontology construction methodologies. The listed methodologies all share the same basic steps, shown in Figure 1[2]. These steps include, apart from

[1] Cited after `http://www-ksl.stanford.edu/kst/what-is-an-ontology.html`

[2] "LEL", used in the figure, means "Language Extended Lexicon", a notation used in [2]

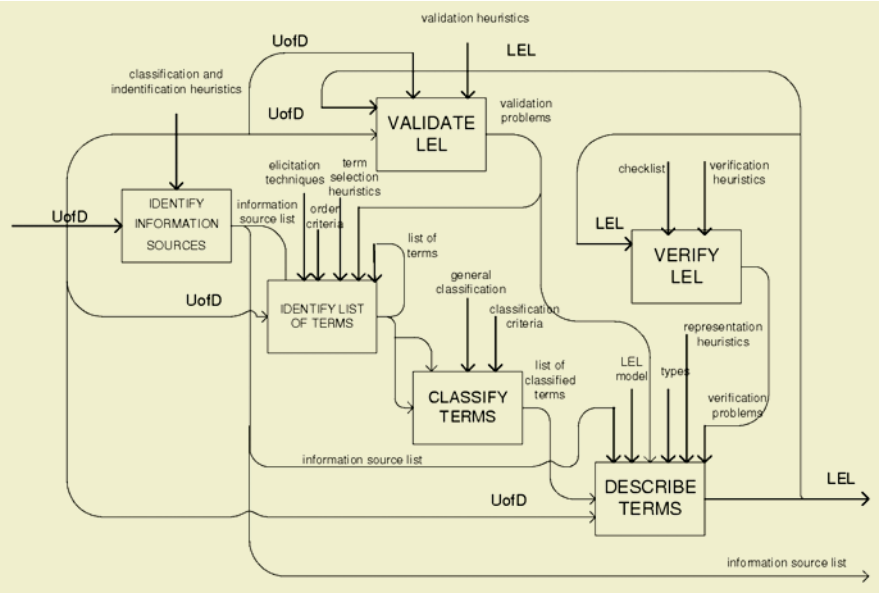



**Fig. 1.** Steps of Ontology Construction [2].

validation and verification, identification of information sources, identification of the list of terms, classification of the terms and their description.

All the approaches listed by Breitman & Sampaio do Prado Leite [2] are rather abstract in the sense that they do not specify *how* to identify information sources, *how* to classify terms, and so on. When ontology construction is considered as a phase of the requirements engineering process, identification of the information sources is simple: the primary information source is the requirements document. Other steps of ontology construction can be done by analyzing this document. The next section introduces a text analysis approach that performs these ontology building steps.

## 3 Ontology Building by Means of Text Analysis

This section is a very short summary of [3]. It shows how text analysis can be applied to the previously introduced ontology construction steps. Figure 2 shows single steps of the ontology extraction approach. The steps correspond to those shown in Figure 1: "parsing and subcategorization frames extraction" corresponds to the identification of the term list, "term clustering & taxonomy building" and "association mining" correspond to term classification. Term description, validation and verification have to be done manually and are not shown in Figure 2.

The approach shown in Figure 2 is interactive, i.e. some decisions must be made by the analyst. Interactivity is important because fully automatic extraction procedure could not detect inconsistencies, that are extremely common in

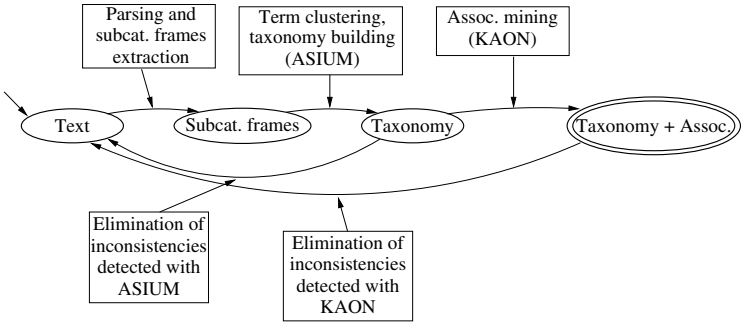



**Fig. 2.** Ontology Building Procedure [3].

requirements documents. The overall process of ontology construction consists of four steps: term extraction, term clustering, taxonomy building (as a cluster hierarchy) and relations mining.

**Extraction of terms from the requirements text:** To extract terms, each sentence is parsed and the resulting parse tree is decomposed. Noun phrases that are related to the main verb of the sentence are extracted as domain concepts. For example, from the sentence "`The control unit sends an alarm message in a critical situation`" "`send`" is extracted as the main verb, "`control unit`" as the subject and "`alarm message`" as the direct object.

**Term clustering:** The second phase clusters related concepts. Two concepts are considered as related and put in the same cluster if they occur in the same grammatical context. For example, if the requirements document contains two sentences like

1. "The control unit sends an alarm message in a critical situation"
2. "The measurement unit sends measurements results every 5 seconds",

the concepts "`control unit`" and "`measurement unit`" are considered as related, as well as "`alarm message`" and "`measurements results`".

**Taxonomy building:** Concept clusters constructed in the previous step are used for taxonomy construction by joining intersecting clusters to larger clusters. The resulting larger clusters represent more general concepts.
For example, the basic clusters {`alarm message`, `measurements results`} and {`control message`, `measurements results`} can be joined into the larger cluster:

{`alarm message`, `control message`, `measurements results`}

representing possible messages. The tool ASIUM [4] is used both to cluster terms and to build a taxonomy.
During this step the terminology is validated with respect to synonyms[3]. Synonyms are often contained in the same cluster. For example, if a cluster contains both "`signal`" and "`message`", the analyst can identify them as synonyms.

[3] different names for the same concept

**Associations/relations mining:** There is a potential association between two concepts if they occur in the same sentence. Each potential association again has to be validated by the requirements engineer. Note that the validation of the association proposed by the association mining tool automatically implies a validation of the requirements document. If the tool suggests an association that *can not* be valid (i.e., a pair containing completely unrelated concepts), then we have detected an evidence that the requirements document contains some inconsistent junk that must be eliminated (see [5] for an in–depth treatment of association mining).

The remainder of this section presents term extraction in more detail, because these details will be important for the integrated approach as well (see Sections 4 and 5).

### 3.1 Term Extraction

Term extraction bases on parsing of each sentence and the representation of every sentence as a parse tree. To build a parse tree, the parser by Michael Collins [6] was used. This parser provides for each parse tree node information about the head (most important) child. An example parse tree is shown in Figure 3. Meanings of the tags used in Figure 3 are introduced in the "Bracketing Guidelines for Treebank II Style Penn Treebank Project" by Bies et al. (`http://www.cis.upenn.edu/\~treebank/home.html`). In a nutshell, $S$ marks complete sentences, $VP$ marks verb phrases, $VB$ marks verbs, $MD$ modal verbs, $NP$ marks noun phrases, $PP$ marks prepositional phrases and $NN$ marks nouns.

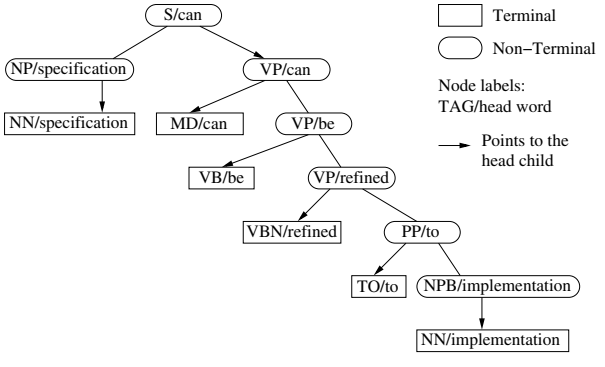



**Fig. 3.** Parse tree for "`Specification can be refined to implementation.`".

The term extraction algorithm extracts not only the terms, but also the predicates. The predicates will be used later for term clustering. To extract the predicate, the extraction algorithm descends from the root node to the head leaf. That is, it descends to the root's head child, then to its head child and so on. This descend process yields the main verb of the sentence. For example,

this method extracts “can” from “can be refined . . . ” (Figure 3). “Can” is not really interesting for term classification, so it is necessary to correct the extracted predicate.

The correction algorithm works in the following way: It starts with the verb node extracted initially, i.e. “MD/can” in the case of Figure 3, and looks for sibling verb or verb phrase nodes. In the case of Figure 3 the algorithm finds “VP/be”. It descends from “VP/be” to its head child node “VB/be” and looks for the sibling verb or verb phrase nodes again. In such fashion it reaches “VBN/refined”. This node does not have any sibling verb or verb phrase nodes, so “VBN/refined” is the verb node that is interesting for term classification.

To extract the subject, the extraction algorithm starts with the main predicate node, e.g., “VP/can” in Figure 3 and traverses the parse tree to the left until it finds a noun phrase. In Figure 3 it finds “NP/specification”. Then it descends to the head child of the noun phrase, which is “NN/specification”. The objects are extracted in a similar way.

The algorithm described above extracts just concepts consisting of a single word. There is also an algorithm extension, extracting compound concepts like “failure of some unit”: It starts with the leaf concept node (“failure”) and goes up in the parse tree, looking for noun phrases and “of”–constructions. The extension is not presented here in more detail for the sake of brevity. See [3] for the detailed description.

The provided information about subjects and objects is used for term clustering, as explained above. It will also be used later for resolution of pronominal anaphora (see Subsection 4.1).

## 4 Alternative Text Analysis Approaches

The approach presented in the previous section was tested on several case studies [3] where it produced good results. However, there is still room for improvement. For example, the method presented above cannot cater for cross–sentence references and cannot extract terms that occur solely in grammatically incorrect sentences. The goal of this section is to show other existing approaches that would augment the original approach and produce even better results, when integrated.

The whole plethora of the developed text analysis methods can be classified in three categories. Ben Achour [7] classifies the linguistic methods as either lexical or syntactic or semantic. Lexical methods, as for example AbstFinder [8] are the most robust ones. AbstFinder extracts the terms (lexica) that occur repetitively in the specification text. This method is extremely robust because it does not rely on part–of–speech analysis, parsing or something like that. It just considers sentences as character sequences and searches for common subsequences in different sentences. It does not perform any term classification. Subsection 4.1 describes it in more detail, in order that it can be used in the integrated approach in Section 5.

Syntactical approaches analyze sentence structure. They are more demanding to the grammatical correctness of the text than lexical approaches, but in

return they extract more information from the text. The approach presented in Section 3 is a syntactical one. Other syntactical approaches suggest other methods of term classification and clustering. These methods will be considered in Subsection 4.2.

Semantical approaches promise more than the other two classes: They interpret each sentence as a logical formula. However, semantical approaches, as for example [9], are extremely fragile, as they require firm sentence structure. For this reason they are barely applicable to real world requirements documents and will not be considered further in this paper.

### 4.1 Lexical Approaches: Term Identification

The great advantage of the lexical methods as compared to syntactical and semantical ones is their robustness. This robustness stems form the fact that they consider each sentence just as a character sequence. AbstFinder by Goldin and Berry [8] is based on this idea: It finds common character subsequences in every sentence pair. For example, consider the following two sentences (taken from one of the case studies discussed in [3]):

> The steam-boiler is characterized by the following elements:

and

> Above m2 the steam-boiler would be in danger after five seconds, if the pumps continued to supply the steam-boiler with water without possibility to evacuate the steam.

They contain a common character sequence "steam-boiler", so "steam-boiler" is identified as a potential domain concept. AbstFinder is interactive, so the requirements analyst may decide which of the extracted character sequences really represent domain–specific terms.

**Anaphora Resolution:** The term extraction method introduced above assumes that the terms are explicitly present in the text. However, the usage of pronouns is very frequent, which undermines this assumption. For example, in the following two sentences, taken from the one of the case studies from [3], the second sentence does not name the received message explicitly:

> The program enters a state in which it waits for the message steam-boiler-waiting to come from the physical units. As soon as this message has been received the program checks whether the quantity of steam coming out of the steam-boiler is really zero.

So, AbstFinder would not identify "`message steam-boiler-waiting`" as a common substring of the two sentences. Usage of pronouns poses problems to the term extraction approach based on parse trees (introduced in Subsection 3.1) as well: It would extract just "this message" as a subject of "received" from the second sentence.

This problem can be solved by the means of anaphora resolution [10]. Resolution of pronominal anaphora would identify "`this message`" in the second sentence with "`message steam-boiler-waiting`" in the first one. An additional advantage of applying anaphora resolution would be detection of referential ambiguities. A referential ambiguity, according to the definition by Kamsties et al. [11] "is caused by an anaphora in a requirement that refers to more than one element introduced earlier in the sentence or in a sentence before". For example, in the sentences

```
The controller sends a message to the pump.
It acknowledges correct initialization.
```

"it" can refer both to the pump and to the controller and to the message. Explicit anaphora resolution would disambiguate this reference. In the case of wrong resolution it would make the referential ambiguity visible.

Anaphora resolution, as presented by Judita Preiss [10], depends on the extraction of grammatical roles. The term extraction algorithm, presented in Subsection 3.1, extracts subjects and objects from each sentence, so it can be used as preprocessor for anaphora resolution.

### 4.2 Syntactical Approaches: Clustering and Taxonomy Building

Syntactical approaches make use of sentence structure to classify the extracted terms. The definition of a cluster, used in the tool ASIUM [4] described before, is very simple: A cluster is built by all the subjects or all the objects of some verb. It is also possible to use other sentence information for the classification purpose. Nenadić et al. [12] introduce following definitions of related terms:

**Contextual Similarity** of two terms measures the number of common and different contexts for the two terms whose similarity should be determined. For this measure the context is defined as a sequence of particular words with their Part–of–Speech (POS) tags (noun, verb, etc.) occurring in the sentence before and after the term. It is up to the analyst to use all the context words and tags or to define some words or word classes (adjectives, conjunctions, ...) as irrelevant and filter them out. It depends on the text domain which contexts (POS sequences, lexica, etc.) provide better term clustering. For this similarity measure to work, the requirements analyst has to decide which contexts to use. This decision can rely on the quality measure for contexts, also introduced by Nenadić et al. [12].

**Lexical Similarity** of two terms measures the presence of common lexical heads (e.g., "message" in "start message" and "stop message") and the number of common modifiers. For example, "first start message" and "second start message" are more similar according to this measure than "start message" and "stop message". Lexical heads are provided by the parser, used in Subsection 3.1. So, lexical similarity can be measured on the basis of parse subtrees for each term, extracted in Subsection 3.1.

**Syntactical Similarity** checks for the presence of certain standard constructions. For example, in the construction "Xs, such as A, B, and C", $X$, $A$, $B$ and $C$ are seen as similar. The syntactical similarity measure is discrete: It can be either 0, if terms are not similar, or 1, if terms are similar.

To decide whether two terms are similar, a linear combination of the three above measures is calculated. Terms with high net similarity can be grouped to clusters. Subsequent taxonomy building on the basis of term clusters and cluster intersections can be done in exactly the same way as in Section 3.

## 5 Integrated Ontology Building Approach

Previous section showed that there are many methods potentially able to improve the original ontology extraction approach from Section 3. This section shows how all the approaches can be integrated. The goal of the integration is to join the strengths and to hide the weaknesses of the isolated methods.

Figure 4 shows an overview of the proposed integrated approach. Just as the approach shown in Figure 2, it starts with the specification text, written in natural language, and extracts an ontology. However, it consists of much more steps and extracts more information from the text. The remainder of this section presents each step in detail.

**Parsing and anaphora resolution:** The goal of this very first step is to get rid of pronominal cross–sentence references. Anaphora resolution is necessary for the later steps, because it replaces pronouns by fully-fledged terms, so that these terms instead of pronouns can be extracted.
The results of anaphora resolution should be examined by the domain expert. It is possible that some anaphora be resolved incorrectly, either due

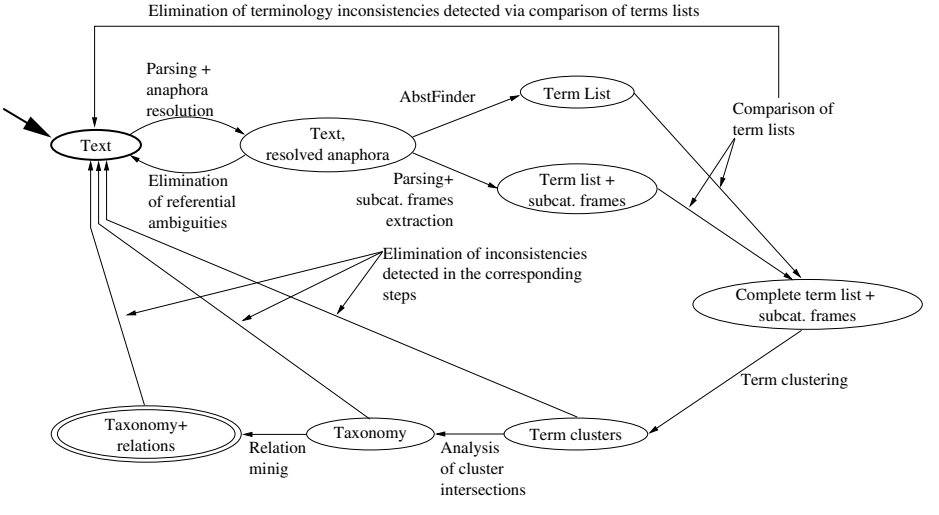



**Fig. 4.** Integrated Ontology Extraction Approach.

to referential ambiguity (several possibilities to resolve an anaphora) or due to deficiencies of the used parser or resolution algorithm. The manual revision of the resolution results would make sure that the terms are substituted correctly.

**AbstFinder:** AbstFinder considers the sentences just as character sequences and extracts character sequences that are common for at least two sentences. The requirements analyst may decide which sequences really represent a term. The extraction of *common* character sequences explains the necessity for anaphora resolution: The resolved pronouns become sensible character sequences.

**Parsing and subcategorization frames extraction:** The text with resolved anaphora should be parsed anew. (Actually, it is necessary to parse only sentences that were changed because of anaphora resolution.) Then, the terms can be extracted using the technique described in Section 3.1. In this way the subcategorization frames (verbs and their arguments) are extracted. Verbs can be used later to cluster terms.

**Comparison of term lists:** Neither AbstFinder nor the extraction of subcategorization frames can *guarantee* the extraction of *all* the terms: AbstFinder extracts terms that occur at least twice in the text, and subcategorization frames extraction can extract terms that occur in grammatically correct sentences only.

Comparison of the two extracted term list can give important information about the text: It shows which terms are often (at least twice) used in the text, but solely in grammatically incorrect sentences. It shows also which terms are used just once. If this term list comparison discovers some omissions in the document, it is up to the requirements analyst to change the text to correct the flaws.

**Term clustering:** To build a taxonomy, it is necessary to find related terms first. The criteria that can be used to cluster related terms were introduced in Subsection 4.2: contextual, lexical and syntactic term similarity. The weights of each of the similarity measures can vary depending on the analyzed text. The produced term clusters should be examined by the requirements analyst. Unrelated terms put in the same cluster usually signalize either a terminology inconsistency or inaccurate phrasing somewhere in the text. The sentences using inconsistent terminology can be found by simple text search: for the inconsistent cluster it is known which terms, used in which context, caused the cluster inconsistency. So, it is sufficient to look for the sentences containing the term in the corresponding context. The detected inconsistencies should be corrected before the analysis continues.

It could be argued that the clustering heuristic itself is a potential source of cluster inconsistencies. However, the clustering heuristic presented by Nenadić et al. [12], that shall be used in the integrated approach, can be trained on the particular domain, which minimizes this inconsistency source.

**Taxonomy building:** To build a taxonomy, it is necessary to determine, which clusters are related. This can be done for example by analysis of cluster intersections and joining them to larger clusters, representing more general concepts. This step is the same as in the simpler ontology extraction approach, described in Section 3.

**Relation mining:** In the last step the taxonomy is augmented by more general relations. This step is exactly the same as in the simpler ontology extraction approach, described in Section 3: There is a potential association between two concepts if they occur in the same sentence. Each potential association again has to be validated by the requirements engineer. The validated associations are absorbed into the ontology.

The proposed approach requires manual intervention. However, manual intervention is necessary to detect inconsistencies. As Goldin and Berry state [8], complete automation is not desirable if it could lead to information loss or wrong results. Thus, interactivity is not a weakness but an important feature of the proposed approach. Due to this interactivity the extracted ontology is validated *by construction*. The result of the whole procedure is a validated application domain ontology *and* a corrected textual specification, free from terminology inconsistencies. The corrected textual specification is itself as important as ontology extraction.

## 6 Conclusion

Requirements engineering is a non–trivial task and the proposed approach is not able to solve all the requirements engineering problems. However, it tackles an extremely important step, namely establishing a common language for the stakeholders. An application domain ontology serves as such a common language. After the construction of this common language it is also important to validate the results, which is also achieved by the proposed approach.

This paper showed how the existing text analysis approaches aiming at ontology extraction can be combined to produce better results than each approach on its own. Drawbacks of every single technique are compensated in the proposed integrated approach by complementary analysis methods: Extraction of subcategorization frames works for grammatically correct sentences only, but it can be augmented by AbstFinder, analyzing character sequences. For AbstFinder to provide better results, it is necessary to replace pronouns by the terms they refer to, and so on. It makes no sense to list here all the interdependencies: this would lead to complete repetition of the previous section.

The final claim of the paper is that natural language processing is mature enough to be applied to ontology extraction in the context of requirements engineering, in spite of the necessary manual intervention. Automation is possible for every ontology construction step and a comprehensive approach is “just” a matter of integration of the existing techniques. Surely the integrated approach needs validation on case studies. The simpler approach has already been validated [3] and, obviously, the integrated approach can be at least as good as the simpler one.

## Acknowledgements

Here I want to thank David Faure, Claire Nédellec, Helmut Schmid and Goran Nenadić for their insightful cooperation during the work. I also want to thank the anonymous reviewers who helped to improve the paper.

## References

1. Zave, P., Jackson, M.: Four dark corners of requirements engineering. ACM Trans. Softw. Eng. Methodol. **6** (1997) 1–30
2. Breitman, K.K., Sampaio do Prado Leite, J.C.: Ontology as a requirements engineering product. In: Proceedings of the 11th IEEE International Requirements Engineering Conference, IEEE Computer Society Press (2003) 309–319
3. Kof, L.: An Application of Natural Language Processing to Domain Modelling – Two Case Studies. International Journal on Computer Systems Science Engineering **20** (2005) 37–52
4. Faure, D., Nédellec, C.: ASIUM: Learning subcategorization frames and restrictions of selection. In Kodratoff, Y., ed.: 10th European Conference on Machine Learning (ECML 98) – Workshop on Text Mining, Chemnitz Germany (1998)
5. Maedche, A., Staab, S.: Discovering conceptual relations from text. In W.Horn, ed.: ECAI 2000. Proceedings of the 14th European Conference on Artificial Intelligence, Berlin, IOS Press, Amsterdam (2000) 321–325
6. Collins, M.: Head-Driven Statistical Models for Natural Language Parsing. PhD thesis, University of Pennsylvania (1999)
7. Ben Achour, C.: Linguistic instruments for the integration of scenarios in requirement engineering. In Cohen, P.R., Wahlster, W., eds.: Proceedings of the Third International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ'97), Barcelona, Catalonia (1997)
8. Goldin, L., Berry, D.M.: AbstFinder, a prototype natural language text abstraction finder for use in requirements elicitation. Automated Software Eng. **4** (1997) 375–412
9. Fuchs, N.E., Schwertel, U., Schwitter, R.: Attempto Controlled English (ACE) language manual, version 3.0. Technical Report 99.03, Department of Computer Science, University of Zurich (1999)
10. Preiss, J.: Choosing a parser for anaphora resolution. In Cohen, P.R., Wahlster, W., eds.: DAARC 2002, 4th Discourse Anaphora and Anaphor Resolution Colloquium, Lisbon, Edições Colibri (2002) 175–180
11. Kamsties, E., Berry, D.M., Paech, B.: Detecting ambiguities in requirements documents using inspections. In: Workshop on Inspections in Software Engineering, Paris, France (2001) 68 –80
12. Nenadić, G., Spasić, I., Ananiadou, S.: Automatic discovery of term similarities using pattern mining. In: Proceedings of CompuTerm 2002, Taipei, Taiwan (2002) 43–49

# Improving Text Categorization Using Domain Knowledge

Jingbo Zhu and Wenliang Chen

Natural Language Processing Lab
Institute of Computer Software and Theory
Northeastern University, Shenyang, P.R. China, 110004
{zhujingbo,chenwl}@mail.neu.edu.cn

**Abstract.** In this paper, we mainly study and propose an approach to improve document classification using domain knowledge. First we introduce a domain knowledge dictionary NEUKD, and propose two models which use domain knowledge as textual features for text categorization. The first one is BOTW model which uses domain associated terms and conventional words as textual features. The other one is BOF model which uses domain features as textual features. But due to limitation of size of domain knowledge dictionary, we study and use a machine learning technique to solve the problem, and propose a BOL model which could be considered as the extended version of BOF model. In the comparison experiments, we consider naïve Bayes system based on BOW model as baseline system. Comparison experimental results of naïve Bayes systems based on those four models (BOW, BOTW, BOF and BOL) show that domain knowledge is very useful for improving text categorization. BOTW model performs better than BOW model, and BOL and BOF models perform better than BOW model in small number of features cases. Through learning new features using machine learning technique, BOL model performs better than BOF model.

## 1 Introduction

Text categorization (TC) is the problem of automatically assigning one or more predefined categories to free text documents. TC is a hard and very useful operation frequently applied to the assignment of subject categories to documents, to route and filter texts, or as a part of natural language processing systems. A growing number of statistical classification methods and machine learning techniques have been applied to text categorization in recent years, such as Rocchio[1][2], SVM[3], Decision Tree[4], Maximum Entropy model[5], naïve Bayes[6]. In those models, typically the document vectors are formed using bag-of-words model. Each document text is represented by a vector of weighted terms. The terms attached to documents for content representation purposes may be words or phrases derived from the document texts by an automatic indexing procedure.

As we know, it is natural for people to know the topic of the document when they see specific words in the document. For example, when we read a news, if title of the news includes a word "姚明(Yao Ming)", as we know, "姚明(Yao Ming)" is a famous China basketball athlete in US NBA game, so we could recognize the topic of the document is about "篮球, 体育(Basketball, Sports)" with our domain knowledge. In this paper, we call the specific word "姚明(Yao Ming)" as a *Domain Associated Term* (DAT). A DAT is a word or a phrase (compound words) that enable humans to

A. Montoyo et al. (Eds.): NLDB 2005, LNCS 3513, pp. 103–113, 2005.
© Springer-Verlag Berlin Heidelberg 2005

recognize intuitively a topic of text with their domain knowledge. As we know, domain knowledge is a kind of commonsense knowledge. We think that domain knowledge is very useful for text understanding tasks, such as information retrieval, text categorization, and document summarization.

Some researchers used knowledge bases to knowledge-based text categorization[7]. First they group words into special semantic clusters according to their definition of knowledge bases such as WordNet or HowNet. Then they use these clusters as features for text categorization. As we know, WordNet and HowNet are lexical and semantic knowledge dictionaries. Sangkon Lee *et. al.* [8] proposed a new passage retrieval method which divides a text into several passages by using field-associated terms like our DATs. But their knowledge bases are generated by hand, and in particular, due to limitation of size of knowledge bases, they can't include enough words or features for text categorization. In this paper, we study and propose two models which use domain knowledge as textual features to improve text categorization. And we use a machine learning technique to solve problem of knowledge-based text categorization caused by limitation of size of domain knowledge dictionary.

The following paper is organized as follows. In section 2, our domain knowledge dictionary is introduced. The baseline NB system based on BOW model is given in section 3. In section 4 we propose two new models using domain knowledge as textual features for text categorization. In section 5, we propose a machine learning technique to improve knowledge- based text categorization. Comparison experimental results of four models are given in section 6. At last, we address conclusions and future work in section 7.

## 2 Domain Knowledge Dictionary

First, we introduce briefly the domain knowledge hierarchy description framework (DKF) which can be divided into three levels shown in Figure 1: *Domain Level (DL)*, *Domain Feature Level (DFL)* and *Domain Associated Term Level (DATL)*. The DL is the top level which includes many domains, such as "体育(Sports)", "军事(Military Affairs)". The DFL is the second level which includes many domain features. A domain has one or more domain features. For example, domain "军事(Military Affairs)" has many domain features, such as "军队(Army Feature)", "武器(Weapon Feature)" and "战争(War Feature)". The DATL is the third level which includes many domain associated terms. As we know, many domain associated terms could indicate a same domain feature. For example, for domain feature "战争(War)", it includes many domain associated terms such as "中东战争(Mid-East War)", "伊拉克战争(Iraq War)" and "阿富汗战争(Afghanistan War)".

Since 1996 we employed a semi-automatic machine learning technique to acquire domain knowledge from a large amount of labeled and unlabeled corpus, and built a domain knowledge dictionary named NEUKD[9][10]. Items defined in the NEUKD include domain associated term, domain feature and domain. Currently 40 domains, 982 domain features and 413,534 domain associated terms are defined in NEUKD. Some instances defined in NEUKD are shown in Table 1. For example, term "三峡工程(The Sanxia Project)" indicates domain feature "水利工程(Irrigation Project)" of domain "水利(Irrigation Works)".

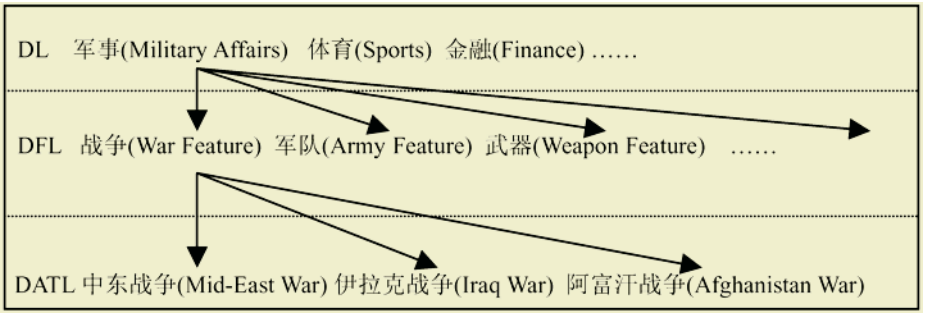



**Fig. 1.** Parts of domain knowledge hierarchy description framework (DKF).

**Table 1.** Some instances defined in NEUKD.

| Domain Associated Terms | Domain Features | Domain |
|---|---|---|
| 姚明<br>(Yao Ming) | 篮球，运动员<br>(Basketball, Athlete) | 体育<br>(Sports ) |
| 三峡工程<br>(The Sanxia project) | 水利工程<br>(Irrigation Project) | 水利<br>(Irrigation Works) |
| 赛季<br>(Match Season) | 比赛<br>(Match) | 体育<br>(Sports ) |
| 阿森纳队<br>(Arsenal Team) | 足球<br>(Football) | 体育<br>(Sports) |
| 中国工商银行<br>(Industrial and commercial bank of China) | 银行<br>(Bank) | 金融<br>(Finance) |

## 3 Baseline NB System

In recent years Naïve Bayes (NB) approaches has been applied for document classification, and found to perform well. The basic idea in naïve Bayes approaches is to use the joint probabilities of words and categories to estimate the probabilities of categories when a document is given. The naïve part of NB method is the assumption of word independency, i.e., the conditional probability of a word given a category is assumed to be independent from the conditional probabilities of other words given that category. This assumption makes the computation of the NB classifiers far more efficient than the exponential complexity of non-naïve Bayes approaches because it does not use word combinations as predictors[11]. There are several versions of the NB classifiers. Recent studies on a multinomial mixture model have reported improved performance scores for this version over some other commonly used versions of NB on several data collections[6]. There are several versions of the NB classifiers. McCallum and Nigam gave comparative analysis between two different NB models: multivariate Bernoulli model and multinomial model. They found that the multivariate Bernoulli performs well with small vocabulary sizes, but that the multinomial performs better at larger vocabulary size.

In this paper we use the multinomial mixture model of NB by to classify documents. We only describe multinomial NB model briefly since full details have been presented in [6]. The basic idea in naïve Bayes approaches is to use the joint prob-

abilities of words and categories to estimate the probabilities of categories when a document is given. Given a document *d* for classification, we calculate the probabilities of each category *c* as

$$P(c \mid d) = \frac{P(c)P(d \mid c)}{P(d)}$$
$$= P(c)\prod_{i=1}^{|T|} \frac{P(t_i \mid c)^{N(t_i|d)}}{N(t_i \mid d)!}$$

Where P(c) is the class prior probabilities, $N(t_i|d)$ is the frequency of word $t_i$ in document *d,* T is the vocabulary and |T| is the size of T, $t_i$ is the $i^{th}$ word in the vocabulary, and $P(t_i|c)$ thus represents the probability that a randomly drawn word from a randomly drawn document in category *c* will be the word $t_i$. We can calculate Bayes-optimal estimates for these parameters from a set of labeled training data.

In this paper, we use NEU_TC data set[12] to evaluate the performance of NB classifier and our classifiers. The NEU_TC data set contains Chinese web pages collected from web sites. The pages are divided into 37 classes according to "Chinese Library Categorization"[13]. It consists of 14,459 documents. We do not use tag information of pages. We use the toolkit CipSegSDK[14] for word segmentation. We removed all words that had less than two occurrences. The resulting vocabulary has about 60000 words.

In the experiments, we use 5-fold cross validation where we randomly and uniformly split each class into 5 folds and we take four folds for training and one fold for testing. In the cross-validated experiments we report on the average performance. For evaluating the effectiveness of category assignments by classifiers to documents, we use the conventional recall, precision and $F_1$ measures. Recall is defined to be the ratio of correct assignments by the system divided by the total number of correct assignments. Precision is the ratio of correct assignments by the system divided by the total number of the system's assignments. The $F_1$ measure combines recall (r) and precision (p) with an equal weight in the following form:

$$F_1(r, p) = \frac{2rp}{r + p}$$

In fact, these scores can be computed for the binary decisions on each individual category first and then be averaged over categories. The way is called macro-averaging method. For evaluating performance average across class, we use the former way called micro averaging method in this paper which balances recall and precision in a way that gives them equal weight. The micro- averaged $F_1$ measure has been widely used in cross-method comparisons.

The most commonly used document representation is the so called vector space model[15]. In the vector space model, documents are represented by vectors of terms (textual features, e.g. words, phases, etc.). A document *D* can be represented by a description vector *dv* as: $dv = <c_1, c_2, \ldots\ldots c_n>$. Where *n* is the total number of the selected terms and $c_i$ denotes the term weight of a term $t_i$ in the document *D*. Conventional bag-of-words model (BOW) uses conventional words as textual features, so each document text can be represented by a vector of weighted words.

In this paper, we use the BOW model as baseline NB system. Given above experimental settings, CHI measure is used for feature selection which is the best features

selection methods according to our experiments, the best performance of baseline NB system is 74.6% F1.

## 4 Domain Knowledge as Textual Features

### 4.1 BOTW Model

In this paper, we wish to use domain knowledge dictionary (NEUKD) to improve text categorization. In BOW model, conventional words are used as textual features for text categorization. As above mentioned, more than 400000 domain associated terms (DATs) are defined in the NEUKD, such as "姚明(Yao Ming) ", "三峡工程(The Sanxia project)", and "中国工商银行(Industrial and commercial bank of China)" shown in table 1. In this paper, we use both those domain associated terms and conventional words as textual features, called BOTW models (short for bag-of-terms and words model).

Now we give an example to explain simply the differences between BOW and BOTW models. For example, in the previous examples, a DAT "三峡工程(The Sanxia project, Sanxia is a LOCATION name of China)" can be used as a textual feature in BOTW model. But in BOW model it is not used as a textual feature, we consider two words "三峡(The Sanxia)" and "工程(project)" as two different textual features, respectively. From above examples, it is natural for us to understand those domains associated terms are a richer and more precise representation of meaning than keywords (conventional words).

In fact, the classification computation procedure based on BOTW model is same as BOW model. CHI also is used to feature selection in our experiments. According to experimental results, the best performance of BOTW-based classifier is 76.7% F1 which is higher 2.1% than baseline system (BOW model).

### 4.2 BOF Model

As above mentioned, in our NEUKD, each DAT is associated with one or more domain features which the DAT indicates. Such as the DAT "三峡工程(The Sanxia Project)" indicates domain feature "水利工程(Irrigation Project)" of domain "水利(Irrigation Works)". Similar to BOTW model, we want to use those domain features as textual features in NB classifier, called BOF model (short for bag-of-features model). In other words, we do not use "三峡工程(The Sanxia Project)" as a textual feature, but its domain feature "水利工程(Irrigation Project)" as a textual feature in the BOF model.

In BOF model, we firstly transform all DATs into domain features according to definitions in NEUKD, and group DATs of same domain features as a cluster, called Topic Cluster.

*Let T denote set of domain feature, t is a domain feature of T, F denote set of Topic Cluster, DF is set of DATs, $df_i$ is $i^{th}$ DAT in DF.*

If a DAT $df_i$ has a domain feature $t_j$, then $df_i$ can been added into the Topic Cluster $F(t_j)$. We group all domain features in NEUKD into the Topic Clusters. For Examples,

Topic Cluster named "体育(sports)" includes some DATs, such as "赛季(match season)", "阿森纳队(Arsenal)", "奥运会(Olympic Games)", "乒乓球(Table Tennis)", "姚明(Yao Ming)".

In BOF model, we use topic clusters as textual features for text categorization. Also the classification computation procedure based on BOF model is same as BOW model. According to experimental results, the best performance of BOF-based classifier is 68% F1 which is less than BOW and BOTW models. The main reason is that due to the limitation of size of our NEUKD, a large amount of words are removed in training procedure, because no domain features of those removed words are defined in our NEUKD. According to statistical analysis of words occurring in training corpus, we find that 65.01% words occurring in training corpus are not included in the NEUKD. In fact, many of those removed words are useful for text categorization. As denoted in section 2, about 1000 domain features are defined in our NEUKD, so for BOF model, the maximum number of textual features is the total number of domain features defined in NEUKD. But it is very significant that when BOF model performs better than BOW and BOTW model in small number of textual features cases. Detailed analysis will be given in following sections.

## 5 BOL Model

To solve the above problem of the limitation of NEUKD, in this paper, we propose a machine learning technique to improve BOF model. The basic ideas are that we wish to learn new words from labeled documents, group them into the predefined topic clusters based on NEUKD which are formed and used as textual features in BOF model discussed in section 4.2, and use new topic clusters as textual features for text categorization. We call the new model as BOL model which is extended version of BOF model. First we group all DATs originally defined in NEUKD into topic clusters as described in BOF model, which are used as seeds in following learning procedure. Then we want to group other words (not be defined in NEUKD) into these topic clusters.

In this section, we introduce how to learn some words from labeled documents using topic clusters and class distribution of words. We are focus on two topics:

- How to measure the similarity between a word and a topic cluster;
- Learning algorithm.

The first question of such procedures is how to measure the similarity between a word and a cluster. Class distribution of words has showed good performance in words clustering [16][17]. We use a form of "Kullback-Leibler divergence to the mean." Unlike previous works, we propose a new similarity measure for learning algorithm. In our algorithm, the word can only be grouped into one cluster.

### 5.1 How to Measure the Similarity

We should define a similarity measure between a word and a topic cluster, and add the word into the most similar cluster that no longer distinguishes among the word and other members (words or DATs) of the topic cluster. Then, the parameters of the cluster become the weighted average of the parameters of its members.

Firstly, we define the distribution $P(C|w_t)$ as the random variable over classes C, and its distribution given a particular word $w_t$. When we have two words $w_t$ and $w_s$, they will be put into the same cluster f. The new distribution of the cluster is defined

$$P(C|f)=P(C|w_t \vee w_s)$$
$$=\frac{P(w_t)}{P(w_t)+P(w_s)}P(C|w_t)+\frac{P(w_s)}{P(w_t)+P(w_s)}P(C|w_s) \quad (1)$$

Now we consider the case that a word $w_t$ and a topic cluster f will be put into a new cluster $f_{new}$. The distribution of $f_{new}$ is defined as

$$P(C|f_{new})=P(C|w_t \vee f)$$
$$=\frac{P(w_t)}{P(w_t)+P(f)}P(C|w_t)+\frac{P(f)}{P(w_t)+P(f)}P(C|f) \quad (2)$$

Secondly, we turn back the above question of how to measure the difference between two probability distributions. Kullback-Leibler divergence is used to do this. The KL divergence between the class distributions induced by $w_t$ and $w_s$ is written as $D(P(C \mid w_t) \| P(C \mid w_s))$, and is defined as

$$-\sum_{j=1}^{|C|} P(c_j \mid w_t) \log(\frac{P(c_j \mid w_t)}{P(c_j \mid w_s)}) \quad (3)$$

But KL divergence has some odd properties. In order to cover its problems, Baker and McCallum[16] proposed a measure named "KL divergence to the mean" to measure the similarity of two distributions. It is defined

$$S_{Baker}=\frac{P(w_t)}{P(w_t)+P(w_s)}D(P(C|w_t)\|P(C|w_t \vee w_s))$$
$$+\frac{P(w_s)}{P(w_t)+P(w_s)}D(P(C|w_s)\|P(C|w_t \vee w_s))$$

In this paper, we usually measure the similarity of a word and a topic cluster. The cluster has included many words that defined in NEUKD. "KL divergence to the mean" has some problems when it measures the similarity between a word and a cluster. In most cases, if the cluster includes more words, then the result is more similar. Experimental results show that several clusters include so many words while most clusters include only few words. The reason is that Baker and McCallum's "KL divergence to the mean" doesn't account for global information. It can't work well if the numbers of features in the clusters are very different at beginning.

Thus, in learning algorithm we use a new measure that does not have this problem. We add a factor according to the number of words in the cluster. The new similarity of a word $w_t$ and a cluster $f_j$ is defined

$$S=\frac{N(w_t)+N(f_j)}{\sum_{i=1}^{|L|}N(f_i)+|W|}S_{Baker}(w_t,f_j) \quad (4)$$

$$S_{Bucket}(w_t, f_j) = \frac{P(w_t)}{P(w_t)+P(f_j)} D(P(C|w_t) \| P(C|w_t \vee f_j)) + \frac{P(f_j)}{P(w_t)+P(f_j)} D(P(C|f_j) \| P(C|w_t \vee f_j))$$

Where $N(f_i)$ denote the number of words in the cluster $f_i$, W is the list of candidate words. Equation 4 can be understood as the balance of all clusters according to the number of words in them. Our experimental results show that it can work well even if the numbers of features in them are very different at the beginning.

### 5.2 Learning Algorithm

**Table 2.** The Learning Algorithm.

- Preprocessing: Text segmentation, extracting candidate words, and sort the candidate words by CHI method. As above mentioned, all candidate words which are not defined in NEUKD will be grouped into topic clusters in this process.
- Initialization: The words, which are defined in NEUKD, are first added to corresponding topic clusters according to their associated domain features, respectively.
- Loop until all candidate words have been put into topic clusters:
  - Measure similarity of a candidate word and each topic cluster, respectively.
  - Put the candidate word into the most similar topic cluster.

## 6 Experimental Results

### 6.1 Experiment 1: Comparison of BOW, BOTW, BOF, and BOL Classifiers

Using experimental settings discussed in section 2 to evaluate the performance of these four models based on NB classifier, we construct four systems in the experiments, including BOW, BOTW, BOF and BOL classifier. CHI measure is used to feature selection in all system. Detailed comparison results are shown in figure 2.

In figure 2, we could find that BOTW classifier always performs better than BOW classifier when the number of features is larger than about 500. As above mentioned, BOTW classifier considers domain associated items (DAIs) as textual features. From comparative experimental results of BOTW and BOW classifiers, we think that domain associated items are a richer and more precise representation of meaning than conventional words.

Because the total number of domain features in NEUKD is only 982, in figure 2 we find the maximum number of features (domain features) for BOF and BOL classifier is less than 1000. When the number of features is between 200 and 1000, BOF classifier performs better than BOW and BOTW classifiers. It is also obvious that BOL classifier always performs better than other three classifiers when the number of features is less than 1000. As above mentioned, in BOL model, we use a machine learning technique to solve the problem of limitation of size of NEUKD, and group rest 65.01% words into predefined topic clusters as textual features in BOL model. So the classifier based on BOL model can yield better performance than BOF model.

### 6.2 Experiment 2: Performance Analysis Based on Different Size of Corpus

In this experiment, we study the performance of BOW and BOL models when varying number of features and size of training corpus. In Figure 3, T10, T30 and T50 denote the different number of training corpus as 10, 30 and 50 training documents for each category. Naturally, the more documents for training procedure are used, the better the performance of classifier is.

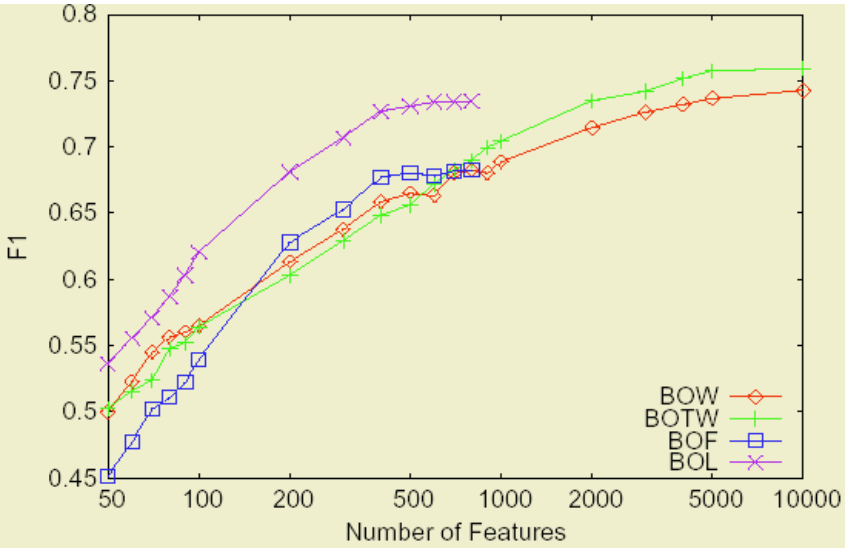



**Fig. 2.** Experimental results of BOW, BOTW, BOF, BOL classifiers.

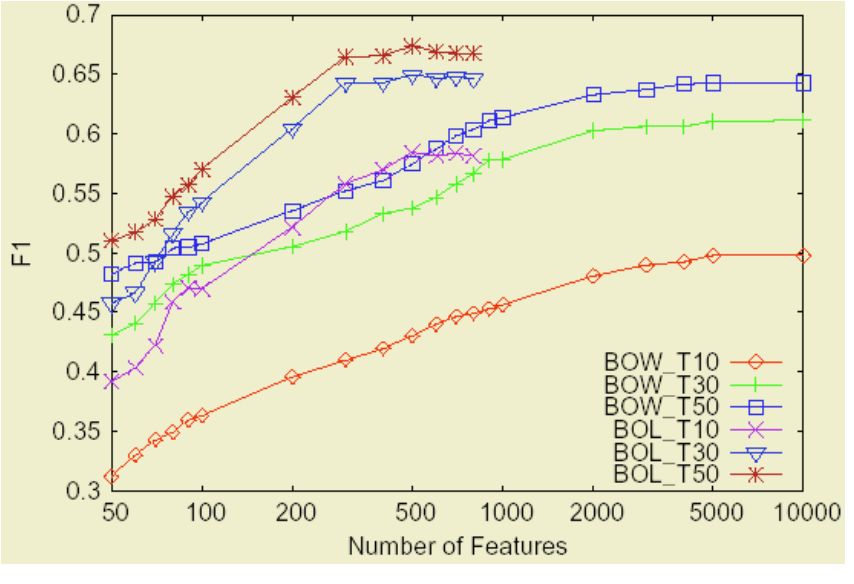



**Fig. 3.** Performance Analysis Based on Different Size of Training Corpus.

In Figure 3, BOL_T10 classifier yields 58.4% F1 with 500 features and BOW_T10 only yields 49.7% F1 with same number of features. And when the number of features is 500, BOW_T50 classifier provides only 57.5% F1, which is lese 0.9% than BOL_T10 classifier. It is obvious that BOL performs better than BOW in small number of features cases.

The best result of BOL_T50 classifier is 67.4% F1, which is higher 9% than BOL_T10 classifier. And the best result of BOW_T50 is 64.2% F1, which is higher 14.5% than BOW_T10 classifier. And the best performance of BOL_T30 classifier is 65.0% F1, which is higher 0.8% than the best performance of BOW_T30 classifier. The best performance of BOL_T50 is 67.4% F1, which is higher 3.2% F1 than the best performance of BOW_T50 classifier. When given small size of training corpus, BOL performs better than BOW. As we know, small size of training corpus would cause serious data sparseness problem. From above comparative experimental results we find that domain knowledge is beneficial to solve data sparseness problem.

## 7 Conclusions and Future Work

In this paper, we study and propose an approach to improve text categorization by using domain knowledge dictionary (NEUKD). We propose two models using domain knowledge as textual features. The first one is BOTW model which uses domain associated terms and conventional words as textual features. The other one is BOF model which uses domain features as textual features. But due to limitation of size of domain knowledge dictionary, many useful words are removed in training procedure. We study and use a machine learning technique to solve the problem to improve knowledge-based text categorization, and propose a BOL model which could be considered as the extension version of BOF model. We use NB system based on BOW model as baseline system. Comparison experimental results of those four models (BOW, BOTW, BOF and BOL) denote that domain knowledge is very useful for improving text categorization. In fact, a lot of knowledge-based NLP application systems also face the problem of limitation of size of knowledge bases. Like our work discussed in this paper, we think using machine learning techniques is a good way to solve such problem. In the future work, we will study how to apply the domain knowledge to improve information retrieval, information extraction, topic detection and tracking (TDT) etc.

## Acknowledgements

This research was supported in part by the National Natural Science Foundation of China & Microsoft Asia Research Centre(No. 60203019), the Key Project of Chinese Ministry of Education (No. 104065), and the National Natural Science Foundation of China (No. 60473140).

## References

1. David J. Ittner, David D. Lewis, and David D. Ahn, Text categorization of low quality images. In Symposium on Document Analysis and Information Retrieval, Las Vegas, , Las Vegas. 1995.

2. D. Lewis, R. Schapire, J. Callan, and R. Papka, Training Algorithms for Linear Text Classifiers, Proceedings of ACM SIGIR, pp.298-306, 1996.
3. T. Joachims, Text categorization with Support Vector Machines: Learning with many relevant features. In Machine Learning: ECML-98, Tenth European Conference on Machine Learning, pp. 137--142. 1998
4. D. Lewis, A Comparison of Two Learning Algorithms for Text Categorization, Symposium on Document Analysis and IR, 1994
5. K. Nigam, John Lafferty, and Andrew McCallum, Using maximum entropy for text classification. In IJCAI-99 Workshop on Machine Learning for Information Filtering, pages 61--67, 1999
6. McCallum and K.Nigam, A Comparison of Event Models for naïve Bayes Text Classification, In AAAI-98 Workshop on Learning for Text Categorization,1998
7. Scott, Sam Stan Matwin. Text classification using WordNet hypernyms. Proceedings of the COLING/ACL Workshop on Usage of WordNet in Natural Language Processing Systems, Montreal, 1998.
8. Sangkon Lee and Masami Shishibori. Passage Segmentation Based on Topic Matter, Computer Processing of Oriental Languages, 2002. V15, No 3, P305-340
9. Zhu Jingbo and Yao Tianshun. FIFA-based Text Classification, Journal of Chinese Information Processing, V16, No3, 2002.(In Chinese)
10. Chen Wenliang, Zhu Jingbo, Yao Tianshun, Automatic Learning Field Words by Bootstrapping, Proceedings of the 7th national conference on computational linguistics (JSCL 2003), 2003, (In Chinese)
11. Yiming Yang and Xin Liu, A re-examination of text categorization methods. In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval, 1999
12. Chen Wenliang, Chang Xingzhi, Wang Huizhen, Zhu Jingbo, and Yao Tianshun, Automatic Word Clustering for Text Categorization Using Global Information. AIRS2004, Beijing, 2004
13. China Library Categorization Editorial Board. China Library Categorization (The 4th ed.) (In Chinese), Beijing, Beijing Library Press 1999
14. Yao Tianshun, Zhu Jingbo, Zhang li, and Yang Ying, Natural Language Processing- research on making computers understand human languages, Tsinghua University Press, 2002, (In Chinese).
15. G.Salton and M.J.McGill, An introduction to modern information retrieval, McGraw-Hill, 1983
16. L.D.Baker and A.K.MCallum. Distributional clustering of words for text classification. In Proc. 21st Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, pages 96--103, 1998.
17. F. Pereira. N. Tishby. L. Lee. Distributional clustering of English words. In 30th Annual Meeting of the ACL, p183-190, 1993.

# A Process and Tool Support for Managing Activity and Resource Conflicts Based on Requirements Classification*

Hwasil Yang[1], Minseong Kim[1], Sooyong Park[1], and Vijayan Sugumaran[2]

[1] Department of Computer Science, Sogang University
Sinsu-Dong, Mapo-gu, Seoul, 121-742, Republic of Korea
{hwasil,minskim,sypark}@sogang.ac.kr
[2] Department of Decision and Information Sciences
Oakland University, Rochester, MI 48309, USA
sugumara@oakland.edu

**Abstract.** The more complicated and large-scaled software systems get, the more important software requirements become, and detecting conflicts between requirements is one of the essential matters that must be considered for successful software projects. Formal methods have been proposed to tackle this problem by adding formality and removing ambiguity. However, they are hard to understand by non-experts, which limit their application to restricted domains. In addition, there is no overall process that covers all the steps for managing conflicts. We propose a process for systematically identifying and managing requirements conflicts. This process is composed of four steps: requirements authoring, partition, conflicts detection and conflicts management. The detection and management of the conflicts are done based on the requirements partition in natural language and supported by a tool. To demonstrate its feasibility, the proposed process has been applied to a home integration system (HIS) and the results are analyzed.

## 1 Introduction

Requirements of software systems have been shown to be an important factor that has great impact on the success of software projects [6]. Requirements analysis is a highly critical step in the software lifecycle. During this step, a wide range of inconsistencies that originate from conflicting requirements can arise as requirements are elicited from multiple stakeholders to achieve various functions. For example, in a home integration system (HIS) [12], let us suppose a flood control function, which shuts off the water main to the home during a flood, is required along with a fire control function, which turns sprinklers on during a fire. One possible scenario could see sprinklers turning on during a fire and flooding the basement before the fire is brought under control. This would trigger the flood control function to shut off the home's water main, rendering the sprinklers useless. In this case, the system does not perform correctly because of conflicts between requirements, i.e. the flood and fire control functions. Therefore, resolving such problems sooner rather than later in the development process is essential for successful development of the software implementing

* This research was supported by University IT Research Center (ITRC) Project, South Korea, and Oakland University, USA.

A. Montoyo et al. (Eds.): NLDB 2005, LNCS 3513, pp. 114–125, 2005.
© Springer-Verlag Berlin Heidelberg 2005

these requirements. Formal specification languages and formal methods have been suggested specifically to tackle this problem. They, supported by automated tools, make it possible for engineers to elicit and specify the software requirements carefully and precisely. However, they are hard to understand by non-experts, which limit their practical application to some restricted domains. In addition, we argue that there is no process to systematically identify and manage conflicts between requirements produced in natural language. As a result, we present a linguistics based technique for detecting requirements conflicts by using goals and scenarios. Goals and scenarios for requirements specification are specified using phrases in natural language, namely English. This makes requirements documents easy to understand and communicate even to non-technical people. Natural language is user-oriented and used in everyday life, and commonly used in the early phases of software development. Despite many problems such as the lack of formality, structure and ambiguity of natural languages, they are still regarded as one of the most important communication medium between developers and customers [14]. Thus, we propose a systematic process and a supporting tool for detecting and managing requirements conflicts in natural language.

The paper is organized as follows. Section 2 presents related work and discusses some of the limitations of existing approaches. Requirements conflicts and the types of conflicts are defined in section 3. In section 4, the process to detect and manage conflicts between requirements are described. In section 5, our approach is applied to a home integration system (HIS) to demonstrate its applicability with a supporting tool. The paper concludes with some words on further work in section 6.

## 2 Related Research and Problems

### 2.1 Feature Interaction Problems

Requirements interaction or conflict management may seem quite similar to the area of feature interaction. Both areas are concerned with the detection and resolution of negative interactions. However, the area of feature interaction has a narrower view of requirements. Feature interaction problem is an unexpected or negative result caused by interactions between features at run time [1]. They often happen when new features are added to existing systems with various features [7, 8]. Research and analysis of feature interaction problem have been discussed in the literature [1, 3, 12]. Although feature interaction is not a new problem, especially in the telecommunication domain where new features are added to large-scale systems, relatively little research has been done out side of telecommunication. Also, feature interaction can not handle non-functional requirements which are related to quality of software and if the developer does not have knowledge about the domain, feature detection itself is not easy.

### 2.2 Requirements Conflicts

As software gets complex, goal and requirements that are gathered from different stakeholders can lead to conflicts. It is very important that these conflicts get identified and resolved in a timely manner. Robinson [11] has convincingly argued that many inconsistencies originate from conflicting goals, and hence inconsistency management should proceed at the goal level. Also, the definition of conflicts is not clear

in existing research and a systematic method to detect conflicts does not exist. Lamsweerde [4] proposed a methodology called KAOS to control conflicts at the goal level. KAOS can verify goals and control conflicts through modeling and specifying goals at an abstract level. However, this method requires considerable time and effort in applying to large systems.

Another issue related to requirements conflicts is similar to the feature interaction problem. Shehata et al., [2] proposed a requirement analysis technique to reduce system error due to unpredicted interaction. However, the proposed method can only be used by developers with considerable domain knowledge. Also, classification of requirements is not easy because of ambiguity in classification standards.

## 3 Basic Definitions

### 3.1 The Definition of Requirements Conflicts

When unexpected or contradictory interaction between requirements has a negative effect on the results, these kinds of relations are defined as requirements conflicts [1]. This is caused through unpredicted interaction between requirements that accomplish various tasks within the system. This paper proposes a process for managing these kinds of requirements conflicts. For example, 'automatic response function' and 'reception refusal function' are performed in the cellular phone domain. If a limited number called the phone, conflicts between these two functions can arise.

### 3.2 The Type of Requirements Conflicts

In this paper, the type of requirements conflict depends on the cause of the conflict and the authoring structure defined in section 4.1. Requirements sentences are described in goals and scenarios using the authoring structure, which contains Action (Verb) + Object (Object) + Resource (Resource)' [5, 10]. If action and object are the cause of the conflict, in this case an activity conflict can arise. A resource conflict may arise when different components try to use the same resources causing a conflict. A specific definition of the two is given below.

1) Activity conflicts: Activity conflicts arise when requirements belonging to different targets have different objects and achieve the same action or have similar objects and perform opposite actions at the same time. As a result, these conflicts have a negative effect in satisfying the requirements.
2) Resource conflicts: Resources conflicts arise when functions that have different targets attempt to use limited resource at the same time.

## 4 Process for Managing Requirements Conflicts Based on Requirements Partition

The purpose of the requirements conflicts management process is to find unexpected errors, which can happen since independent functions are executed at the same time in a single system, through the automatic classification of requirements. Figure 1 shows the overall structure of the process for requirements conflicts management and the details are described below.

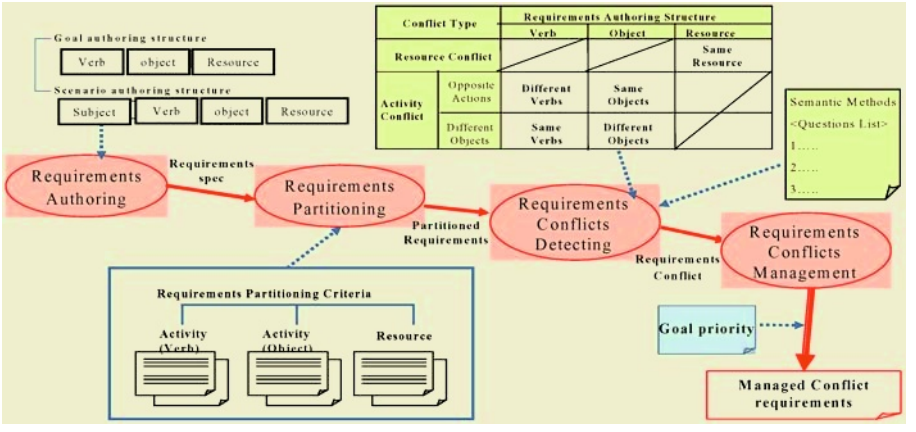



**Fig. 1.** Process of Requirements Conflicts Management.

### 4.1 Requirements Authoring

To begin with, requirements elicited from multiple stakeholders are documented through the following techniques.

- **Function and resource dictionary**
  To capture and specify software requirements carefully and precisely for detecting conflicts, various functions that must be achieved in the system should be recoded in a function dictionary before specific requirements are specified. In the function dictionary, functions that describe similar action patterns are written with same words.
- **Requirements authoring level**
  Based on [10, 13], we organize requirements collection in a three level abstraction hierarchy as follows. They can be useful in clarifying the concerns of requirements and help in conflict detection.
  - Business Level: The aim of this level is to identify and describe the highest level goal which the system under consideration should achieve. Stakeholders or organizations care about these goals.
  - Function Level: Main functions of the system for achieving goals or requirements at the business level are described at this level. Functions that are known to the user are identified and written. Requirements conflicts are detected at the function level.
  - Resource Level: This level describes the resources that are required to support achieving the functions developed at the previous level. For example, in the cellular phone domain, if 'a calling function' is described at the function level, resources required for supporting that function are identified and described at the resource level.
- **Requirements authoring structure**
  This paper uses the goal and scenario authoring structure [5, 10] (see Fig. 2) for requirements specification. Goal and scenario oriented approach is an effective way

to identify and describe requirements [5, 9, 10, 13]. A goal provides the rationale for requirements – a requirement exists because of some underlying goal that provides a basis for it. Since scenarios describe real situations, they capture real requirements and help people in reasoning about complex systems. Thus, the authoring structure offers a guideline for the developer to describe requirements precisely using goals and scenarios.

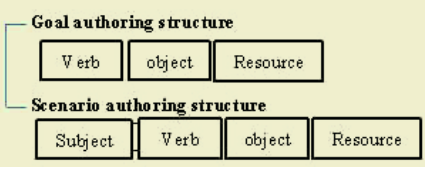



**Fig. 2.** Goal and Scenario Authoring Structure.

- **Goal priority**
  This step decides on priority between described requirements, i.e. goals. Priority of requirements is considered according to the importance of requirement, development possibility and expense etc. Priority of requirements will be used as a standard for managing conflicts when conflicts between requirements arise.

### 4.2 Requirements Partition

In this step, requirements described at the requirement authoring level are classified by the entities of the requirement authoring structure.

#### 4.2.1 Conflict Management Chunk

As defined in section 3.2, a verb and an object are used to detect and analyze activity conflicts. As the component of a requirements sentence, a verb is an independent sentence component using an object. Therefore, this paper defines the notion of "conflicts management chunk" to easily identify requirements conflicts (see Fig. 3). As a basic comparison unit for detecting requirements conflicts, the activity chunk consists of the verb and the object. When the classification is conducted according to the authoring structure, there are some limitations because the verb can be expressed in different ways. However, an object is a specific thing in the real world and is expressed using limited words in a domain. Therefore, even though the verb used may be different, we can reduce the variability of the verb through the activity chunk composed of the verb and the object, which is fixed in a domain.

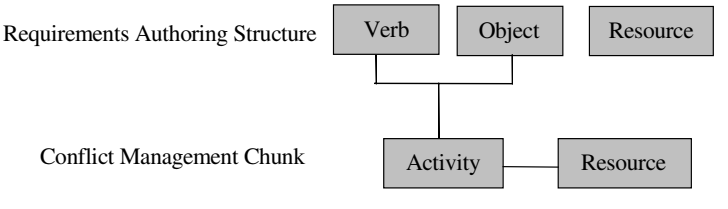



**Fig. 3.** Conflict Management Chunk.

### 4.2.2 Partition Based on Authoring Entity

Similar requirements are grouped according to the conflict management chunk within the partition step. That is, the classification of requirements is conducted based on the structure of requirements authoring, i.e. verb, object and resource for identifying conflicts because the types of requirements conflicts are decided based on the structure. Fig. 4 depicts the requirements partition by the entities, i.e. verb, object, and resource.

| | Verb | Object | Resource |
|---|---|---|---|
| | Activity | | Resource |
| Sc1.1 | Detect | The smoke | With the smoking sensor |
| Sc1.2 | Activate | The alarm | Through the alarm controller |
| Sc1.3. | Turn on | Water system | through the water controller |
| .... | ... | ... | ... |

| | Verb | Object | Resource |
|---|---|---|---|
| | Activity | | Resource |
| Sc2.1 | Detect | The water | With the moisture sensor |
| Sc1.2 | Activate | The alarm | Through the alarm controller |
| Sc1.3. | call | The police station | Through the telephone service |
| .... | ... | ... | ... |

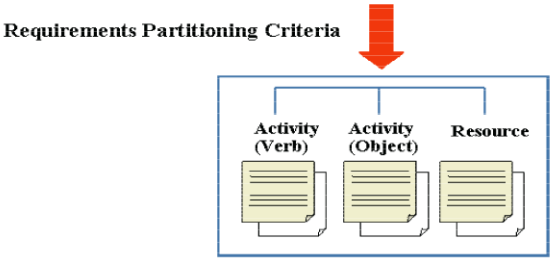



**Fig. 4.** Requirements Partition.

## 4.3 Requirements Conflicts Detection

Two kinds of methods, i.e. a syntactic method and a semantic method are applied to the partitioned requirements for detecting requirements conflicts.

### 4.3.1 Condition for Detecting Requirements Conflicts

The aim of the syntactic method is to identify candidate requirements conflicts by using a pre defined condition. The condition for detecting conflicts was developed based on the conflict definition and the requirement authoring structure as well as the causes of conflicts, namely "Different Verb ∩ Same object", "Same Verb ∩ Different Object", or "Same Resource". Fig.5 illustrates the relationships between requirements sentences. In more detail, in Fig. 5, 'S' indicates that two different requirements are specified using the same word and 'D' refers to the case that two different requirement sentences are described using different vocabulary. Activity conflicts are marked by a circle and resource conflicts are marked by a box. It is worth noting that the case of (S, S, D) in Fig. 5 is not a requirements conflict. For example, "Call the fire station with the telephone service" and "Call the fire station with the Internet" would not be a conflict because these don't use the same resource. Also, conflicts are defined as the opposite relationships between requirements. In the (D, D, D) case, there is no relation among requirements because these use different actions, objects, or resources. Fig. 6 shows the conflicts detection condition on the basis of the relations among requirements.

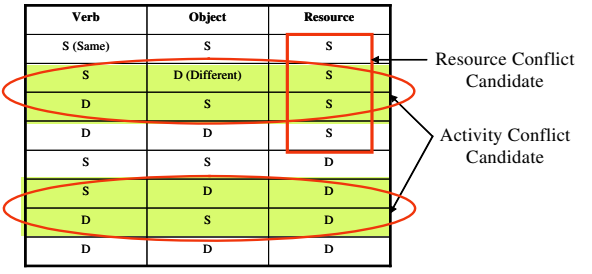



**Fig. 5.** Relations between requirements.

| Conflict Type | | Requirements Authoring Structure | | |
|---|---|---|---|---|
| | | Verb | Object | Resource |
| Resource Conflict | | | | Same Resource |
| Activity Conflict | Opposite Actions | Different Verbs | Same Objects | |
| | Different Objects | Same Verbs | Different Objects | |

**Fig. 6.** Requirements Conflicts Detection Condition.

### 4.3.2 Syntactic Method for Detecting Conflicts

As mentioned before, the purpose of the syntactic method is finding requirements conflicts candidates through the conflicts detection condition. The conflicts detection condition is applied to the requirements groups partitioned by the entities of the authoring structure. For detecting activity conflicts, the condition is applied to the requirements group classified by an object or a verb. Resource conflicts candidates derived from using the same resource are also detected through the condition. As a result, it is possible to reduce the scope of comparison among requirements for identifying conflicts. However, this method has a limitation, that is, it is unable to semantically analyze the requirements produced for detecting real conflicts in the candidate conflicts. To solve this problem, the semantic method is suggested in the next section.

### 4.3.3 Semantic Method for Detecting Conflicts

The purpose of the semantic method is to find actual requirements conflicts through a question list. The question list is defined according to the causes of conflict occurrence and shown below.

**a. Resource conflicts are caused by limited resources (Same Resource)**

Resource conflicts can be identified in the candidate conflicts by analyzing the requirements with these questions.

1) Are two different requirements included in different goals?
2) Are two different requirements using limited resources at the same time?

If the answer is ‘yes’, these requirements could result in resource conflicts.

**b. Activity conflicts are caused by opposite verbs (Different Verb+ Same Object)**

Activity conflicts can arise when different requirements use different actions with the same object.

1) Are two different requirements included in different goals?
2) Do two different requirements prevent the requirements from being satisfied with respect to each other?
3) Is it required to achieve the requirements at the same time? Is it required to perform the opposing actions at the same time?

**c. Activity conflicts are caused by different objects (Same Verb + Different Object).**

1) Are two different requirements included in different goals?
2) Do different objects have negative effects on each other?

After doing the partition and the syntactic method, the scope of requirements for detecting conflicts by the semantic method is reduced. As a result, comparing time and effort to identify and analyze requirements conflicts can also be reduced.

### 4.4 Requirements Conflicts Management

The goal priority decided during the requirements authoring as described in section 4.1 can provide the criteria for managing the conflicts detected, i.e., which requirements should be performed first when conflicts between requirements occur. Thus, based on the priority of the goals, namely requirements, the requirements that have high priority are performed first and an additional condition for handling conflicts is attached to the remaining requirements.

## 5 Case Study

### 5.1 Problem Domain

To demonstrate the feasibility of our approach, we have selected the example of home integration system (HIS) [12]. HIS enhances the comfort, safety, and security of a home. The HIS enables homeowners to access, control, and integrate equipment in their homes such as those listed below. We apply our approach for detecting and managing requirements conflicts to HIS and evaluate it by comparing with other examples. The following sections explain how our process has been implemented using a supporting tool called RECOMA (REquirements COnflicts MAnagement tool) that we have developed. This tool has been designed with a layered architecture. Figure 7 shows the architecture of the supporting tool. The Presentation layer shows predefined screens and provides graphical interface to the user, and the data layer expresses the different kinds of data that are handled by the tool. The Application layer in the middle controls and connects the presentation layer and the data layer. This tool can operate and present various data structures and hierarchies on different platforms and applications because of its modular architecture. This tool has been designed using the Java language, and was implemented using Java development toolkit (JDK 1.4.0_02). Because of space limitation, a detailed description of the tool has not been included, however, we do present some sample screens created using RECOMA from the HIS example for the reader to see how it functions.

## 5.2 Application to HIS

### 5.2.1 Requirements Authoring

The Requirement authoring step creates the function dictionary and resources dictionary to record the requirements. Basic functions such as 'controlling fire', 'managing flood', 'detecting intrusion' and 'controlling temperature' are required in HIS. Also, resources such as 'telephone service', 'Internet' are used. Based on this dictionary, requirements are described according to the requirement authoring structure at the requirements authoring level, i.e. the business and function levels (see Fig. 8). Requirements that are described according to authoring structure are stored in a database. Once requirements authoring is finished, priority between goals is decided by the developer (see the "Priority" part in the middle portion of Fig. 8).

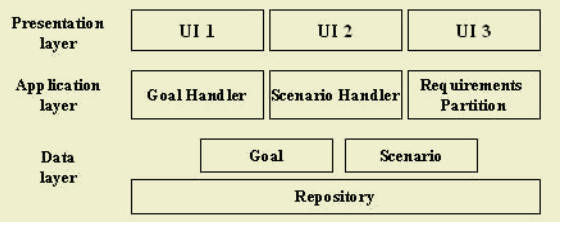



**Fig. 7.** RECOMA Architecture.

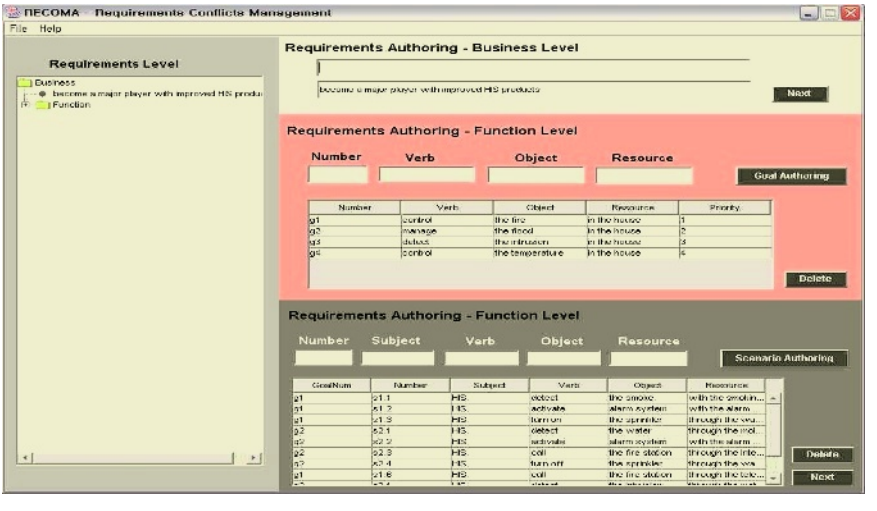



**Fig. 8.** Requirements Authoring by Goals and Scenarios.

### 5.2.2 Requirements Partition

In this step, requirements are classified through the entities of the authoring structure in HIS. The top part of Figure. 9 shows that requirements are grouped by the resource. That is, requirements sentences that have the same word in the "Resource" part of the authoring structure are grouped by comparing the resources, i.e. by pushing the "By Resource" button in the tool. Fig. 9 shows that Sc1.2, Sc2.2 and Sc3.2 have the same

resource (i.e., with the alarm clock). Other requirement groups are also created by the entities of the authoring structure, i.e. verb and resource to classify requirements.

### 5.2.3 Requirements Conflicts Detection

**1) The syntactic method for detecting conflicts**

To find conflict candidates, the syntactic method is used by applying the conflict detection condition. The developer may choose the type of conflict to detect, i.e., resource or activity conflict in the tool (see the middle portion of Fig.9). Thus, based on the requirements classified, the candidate conflicts are detected by using the appropriate detection condition. This is accomplished in the tool by pushing the "Detect" button after selecting the condition in the middle portion of Fig. 9.

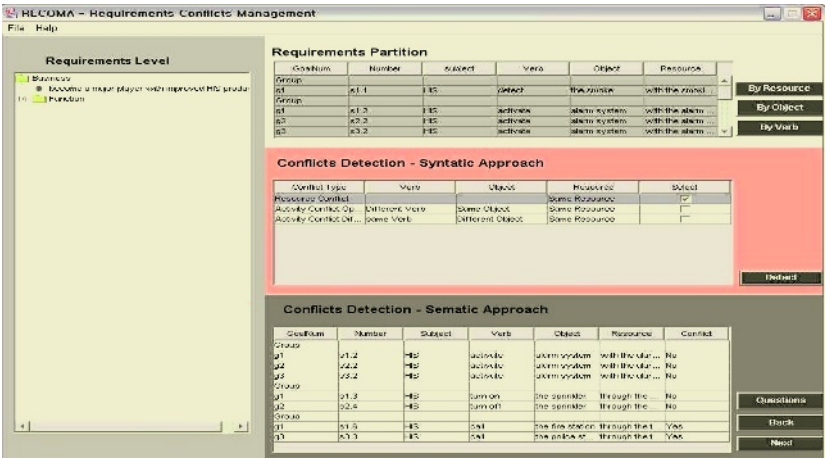



**Fig. 9.** Requirements Partition and Conflicts Detection.

**2) The semantic method for detecting conflicts**

After the syntactic method, requirements conflicts are detected by applying the 'Question List' defined earlier for semantic analysis. The bottom of Figure 9 shows some example conflicts in the HIS example using the semantic method. The details are described below.

**a. Resource conflicts** (Resource(Same)).

Sc1.6 and Sc3.3 use a limited resource, which is "telephone service". Thus, the developer can decide whether there is a conflict are or not through applying the Question List, i.e., "Are two requirements included in different target?", "Are you using limited resources at the same time?" etc. Afterwards, if those are conflicts, "Yes" is entered into the "Conflict" section (see the bottom of Fig. 9).

**b. Activity conflicts by opposite verb** (Verb(Different) + Object(Same)).

In another case, the scenarios "Sc1.3 Turn on the sprinkler through the water controller" and "Sc2.4 Turn off the sprinkler through the water controller" indicate opposite activity with the same object at the same time. Thus, sc1.3 and sc2.4 have negative effect on each other and hence these requirements are flagged as having activity conflicts with the same object.

**c. Activity conflict by different object** (Verb(Same) + Object(Different)).
The requirements “Sc1.6 Call the fire statio*n* through the telephone service" and "Sc 3.3 Call the police station through the telephone service" have different objects, same verb and resource. This situation can be analyzed using the question list to see if there is a requirements conflict because ‘sc1.6’ and ‘sc3.3’ use limited resource at the same time. Similarly, “Sc2.3 Call the fire station through the Internet" and "Sc3.3. Call the police station through the telephone service" have different object and resource but same verb. Activity conflict doesn’t happen here because they use different resources.

### 5.2.4 Requirements Conflicts Management

Our tool presents a new panel for handling requirements conflicts that have been detected. This panel lists the conflicting requirements and the user can add additional conditions for resolving the conflicts based on priorities. For example, ‘Sc1.6’ and ‘Sc3.3’ are identified as a resource conflict in the previous step. According to the priority of the goals decided in the requirements authoring step, extra conditions are specified for the lower level requirement, namely, ‘Sc3.3’.

## 5.3 The Results of Examples

Our approach was applied to the HIS as well as Cellular phone domain and we compared the results to the same cases without using our approach. For analyzing conflicts, the total number of comparisons to be done in the HIS case (with 28 requirements) without our process is 378 (28C2 => (28 * 27) / (2*1)). In the cellular phone case, total number of requirements is 37 and the comparison number is 666 (37C2 = > (37 * 36) / (2 * 1)) for detecting conflicts.

**Table 1.** Number of Requirements Comparison.

| | | | |
|---|---|---|---|
| Requirements authoring | | HIS | 28 |
| | | Cellular phone | 37 |
| Requirements partition | | HIS | 1134 |
| | | Cellular phone | 1998 |
| Requirements Conflicts Detection | Syntactic Method | HIS | 51 |
| | | Cellular phone | 65 |
| | Semantic Method | HIS | 28 |
| | | Cellular phone | 35 |

In contrast to being done manually by developers, in our approach, the comparisons in the partition and syntactic analysis steps are conducted automatically through the supporting tool, i.e. RECOMA. As a result, even though the developer needs to have *a priori* knowledge and capability to handle these many comparisons, a developer can reduce the number of comparisons to be made in order to detect conflicts. Table 1 shows the comparison numbers for detecting requirements conflicts at each step of the process. As the table shows, utilizing our process and the supporting tool can reduce the duration and cost of requirements analysis for detecting requirements conflicts.

## 6 Conclusion

It is important to detect unexpected conflicts between requirements for software projects to be successful. Traditionally, formal methods have been used to detect requirements conflicts. However, these methods need a lot of time and effort to use them properly. In addition, there is no overall process to detect requirements conflicts through the analysis of requirements in natural language. This paper has presented a process to systematically detect and manage requirements conflicts along with a supporting tool. The efficiency of detecting conflicts can be improved through our process and the supporting tool. Additionally, it is possible to reduce analysis time, cost, and developer's effort for identifying analyzing requirements conflicts. For our future work, we intend to expand our approach for identifying and managing conflicts not only between functional requirements, but also non-functional requirements.

## References

1. E.J. Cameron and H.Velthuijsen, "Feature Interaction in Telecommunications systems," IEEE Communication Magazine, vol 31, no.8, pp 46-51, August 1993.
2. Mohamed shehata and Armin Eberlein "Requirements Interaction Detecting Using Semi-Formal Methods", Proceedings of the 10th IEEE International Conference and Workshop on the Engineering of Computer-Based System(ECBS'03).
3. Kimbler, K., Bouma L. G.: Feature Interactions in Telecommunication and Software Systems V. IOS Press, Amsterdam, 1998.
4. Axel van Lamsweerde, "Managing Conflicts in Goal-Driven Requirements Engineering", IEEE Transactions on Software Engineering, Vol. 23, No. 11, November 1998.
5. J. Kim, M. Kim, H. Yang, S. Park, A Method and Tool Support for Variant Requirements Analysis: Goal and Scenario Based Approach, APSEC 2004, S. Korea, 2004, pp. 168-175.
6. F.P.Brooks, "No Silver Bullet: Essence and Accidents of Software Engineering", IEEE Computer, Vol. 20 No 4. April 1987, pp. 10-19.
7. P. Zave, "Architectural Solutions to Feature Interaction problem in Telecommunication" Feature Interactions in Telecommunication and Software Systems V,K Kimbler and L,G Bouma, eds., pp.10-22, IOS Press, September 1998.
8. Y.Peng, F. Khendek, P. Grogono and G.Butler, "Feature Interactions Detection Technique Based On Feature Assumptions," Feature Interaction in Telecommunication and Software system V, K. Kimber and L.G Bouma, eds., pp.291-298, IOS Press, September 1998.
9. A.I. Antón, "Goal-based requirements analysis," in Proc., 2nd Int. Conf. Requirements Engineering (ICRE'96), Colorado Springs, CO, April 1996, pp. 136–144.
10. C. Rolland, C. Souveyet, and C. Ben Achour, "Guiding goal modeling using scenarios," IEEE Trans. Software Eng., vol. 24, pp. 1055–1071, Dec. 1998
11. W.N. Robinson,. "Integrating Multiple Specifications Using Domain Goals," Proc. IWSSD-5-Fifth Int'T Workshop Software Specification and Design, pp.219-225, 1989.
12. K.C. Kang et al., "Feature-Oriented Product Line Engineering," IEEE Software, Vol. 9, No. 4, Jul./Aug. 2002, pp. 58-65.
13. S. Park, M. Kim, V. Sugumaran. "A Scenario, Goal and Feature Oriented Domain Analysis Approach for Developing Software Product Lines," Industrial Management & Data Systems, Vol. 104, No. 4, 2004, pp.296-308.
14. Davis, A.M. Predictions and farewells. IEEE Software. Vol. 15, No. 4, 1998. pp. 6-9.

# Web-Assisted Detection and Correction of Joint and Disjoint Malapropos Word Combinations*

Igor A. Bolshakov[1] and Sofia N. Galicia-Haro[2]

[1] Center for Computing Research (CIC),
National Polytechnic Institute (IPN), Mexico City, Mexico
igor@cic.ipn.mx
[2] Faculty of Sciences, National Autonomous University of Mexico (UNAM),
Mexico City, Mexico
sngh@fciencias.unam.mx

**Abstract.** An experiment on Web-assisted detection and correction of malapropism is reported. Malapropos words semantically destroy collocations they are in, usually with retention of syntactical links with other words. A hundred English malapropisms were gathered, each supplied with its correction candidates, i.e. word combinations with one word equal to an editing variant of the corresponding word in the malapropism. Google statistics of occurrences and co-occurrences were gathered for each malapropism and correcting candidate. The collocation components may be adjacent or separated by other words in a sentence, so statistics were accumulated for the most probable distance between them. The raw Google occurrence statistics are then recalculated to numeric values of a specially defined Semantic Compatibility Index (SCI). Heuristic rules are proposed to signal malapropisms when SCI values are lower than a predetermined threshold and to retain a few highly SCI-ranked correction candidates. Within certain limitations, the experiment gave promising results.

## 1 Introduction

Malapropism is a type of semantic error that replaces one content word by another legitimate word similar in sound or letters but semantically incompatible with the context and thus destroying text cohesion. Particularly, a malapropos word destroys any collocations it is in, i.e. combinations of two syntactically linked (maybe through an auxiliary word like a preposition) and semantically compatible content words. As a consequence, the resulting word combinations usually retain their syntactical type but lose their sense, e.g., *travel around the **world*** transforms to *travel around the **word***. Hence two interconnected tasks emerge in text preparation: revealing erroneous words and supplying the user with selected candidates for their correction.

In [7] a method for malapropism detection and correction is proposed that relies on semantic anomalies in a text indicated by words distant from all contextual ones in terms of WordNet; closer words are searched as editing variants of the anomalous words. The distance is determined through paradigmatic relations (synonyms, hypo-

* Work done under partial support of Mexican Government (CONACyT, SNI) and CGEPI-IPN, Mexico. Many thanks to Denis Filatov, Alexander Gelbukh, and Patrick Cassidy for their help with manuscript preparation.

A. Montoyo et al. (Eds.): NLDB 2005, LNCS 3513, pp. 126–137, 2005.
© Springer-Verlag Berlin Heidelberg 2005

nyms, hyperonyms), mainly between nouns. The syntactic links between words are ignored; matched words are usually from different sentences or even paragraphs.

The general idea in [2] is similar: an anomalous word does not match its context, but the anomaly detection is based on syntactico-semantic links between content words: just these are destroyed by malapropisms. A much smaller context – only one sentence – is needed for error detection, and words of four principal parts of speech (POS) – nouns, verbs, adjective, and adverbs – are considered as collocation components (= collocatives). To test whether a content word pair is a collocation, three types of linguistic resources are presumed: a precompiled collocation database like CrossLexica [1], a text corpus, or a Web searcher. However, the experiment described in [2] is limited and inadequate.

This paper extends [2] by intensive experimentation with Google as a resource for collocation testing. The Web is widely considered now as a huge (but noisy) linguistic resource [5, 6]. For our purposes, it proved necessary to revise the earlier algorithm of malapropism detection & correction and to create new threshold procedures for these operations. Especially important for us is to investigate English collocations of various syntactical types with collocatives either sequentially adjacent (thus forming bigrams well explored nowadays [5]) or distant from each other (such collocations are insufficiently explored in computational linguistics because of the underestimation of dependency grammar approach [9], cf. although [11, 12]).

More specifically our objectives are:

- To clarify the notion of collocation, to classify main collocation syntactical types, and to demonstrate collocatives rather distant in a sentence; to explore frequencies of collocative co-occurrences against the distance between them;
- To define various types of malapropism correction candidates (paronyms);
- To collect a hundred rather common English collocations with their collocatives at the most probable distances, to create malapropisms from them, and then to gather word combinations as primary correction candidates;
- To develop a method of malapropism detection & correction with introduction of a Semantic Compatibility Index (SCI) as a numeric measure for semantic compatibility of collocatives;
- To extract from Google raw statistics of occurrences and co-occurrences for all collected malapropism samples and their primary candidates;
- To transform Google statistics to SCI values for detecting malapropisms and selecting only the best correction candidates to be shown to the user.

## 2 Collocations in Their Adjacent and Disjoint Forms

Syntactico-semantic links between collocatives are considered as in dependency grammars [9]. At the syntactic level, each sentence can be represented as a dependency tree with directed links 'head → its dependent' between tree nodes labeled by word forms of the sentence. Going along these links in the same direction of the arrows from one content node through any functional nodes down to another content node, we obtain a labeled substructure corresponding to a word combination. If this is a sensible text, we call the detected word combination a *collocation*. Such a definition of collocations ignores their frequencies and idiomaticity.

**Table 1.** Frequent types and structures of English collocations.

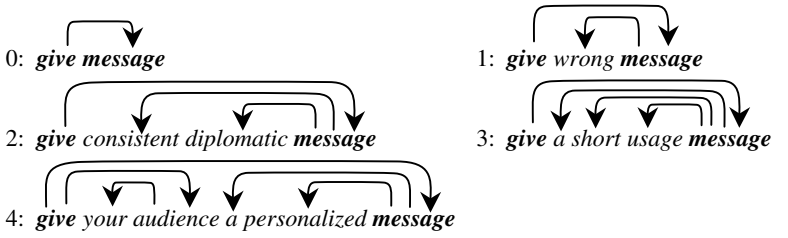
| Type title | Type code | Dependency subtree | Example | % in our set |
|---|---|---|---|---|
| Modified → its Modifier | 1.1<br>1.2<br>1.3<br>1.4<br>1.5 | **[Adj] ← [N]**<br>**[N] → [Adj]**<br>**[Adv] ← [Adj]**<br>**[Adj] → [Adv]**<br>**[V]→ [Adv]** | *strong tea*<br>*pension claimed*<br>*morally inferior*<br>*kicking wildly*<br>(*to*) *mind bitterly* | 18<br>1<br>3<br>1 |
| Noun → its Noun Complement | 2.1 | **[N] → Pr → [N]** | *signs of life* | 8 |
| Noun → its Noun Attribute | 3.1 | **[N] ← [N]** | *socket outlet* | 17 |
| Verb → its Noun Complement | 4.1<br>4.2<br>4.3<br>4.4 | **[V] → [N]**<br>**[V] → Pr → [N]**<br>**[V → Pp] [N]**<br>**[V → Pp] Pr → [N]** | *give books*<br>*goes to cinema*<br>*looked through* (*the*) *books*<br>*fall behind with* (*the*) *rent* | 16<br>6<br>5<br>1 |
| Verb→ its Verbal Complement | 5.1<br>5.2 | **[V] → [*to* → V]**<br>**[V] → [V]** | *tries to find*<br>*let* (*us*) *miss* | 2<br>1 |
| Verb→ its Adjective Complement | 6.1<br>6.2 | **[V] → [Adj]**<br>**[V → Pp] [Adj]** | *looks fine*<br>*comes in handy* | 4<br>1 |
| Verb Predicate → its Subject | 7.1<br>7.2<br>7.3 | **[N] ← [V]**<br>**[V] → [N]**<br>**[N] ← [V→ Pp]** | *light failed*<br>(*there*) *exist people*<br>*braces hold up* | 5<br>1<br>1 |
| Adjective → its Noun Complement | 8.1<br>8.2 | **[Adj] → Pr → [N]**<br>**[Adj] → [N]** | *easy for girls*<br>*tacking seams* | 4<br>1 |
| Coordinated Pair | 9.1<br>9.2 | **[N] → Cc → [N]**<br>**[Adj] → Cc → [Adj]** | *bits and pieces*<br>*safe and sound* | 2<br>1 |

There are several syntactic types of collocations in each language. Frequent types of English collocations are given in Table 1. The types and subtypes are determined by POS of collocatives and their order in texts; **N** symbolizes noun, **Adj** is adjective or participle, **Adv** is adverb, **Pr** is preposition; each collocative is given in brackets. Some collocatives should be taken as multiwords, e.g. English phrasal verbs. Indeed, verbs *look after, look down, look forward, look for, look out, look through* and *look up* are different in meaning and their syntactic features, and the role of their postpositives **Pp** differs from homonymous prepositions **Pr**. So the subtypes 4.3 and 4.4 of 'Verb → its Noun Complement' collocations contain multiword collocatives **[V → Pp]**.

Usually dependency links reflect subordination between words, cf. subtypes 1.1 to 8.2. However, there exist also coordinate dependencies, and we also consider stable coordinate pairs as collocations with collocatives of the same POS linked through the coordinating conjunction **Cc** = *and*, *or*, *but*... (cf. subtypes 9.1 and 9.2).

Though collocatives are always adjacent in their dependency trees, they can be distant in common linear text. The distribution of possible distances depends on the collocation type and specific collocatives. E.g., 3.1-collocatives are usually adjacent,

whereas the 4.1-collocation *give* → *message* can contain intermediate contexts of lengths 0 to 4 and even longer:




A specific collocation or malapropism met in a text has its certain distance between collocatives. However, to explore collocations in a rather general manner, we should put each collocative pair in its most probable distance.

## 3 The Most Probable Distance Between Collocatives

Before determining the most probable distances between specific collocatives by means of the Web, it is necessary to clarify correspondences between the Web frequencies of collocative co-occurrences and of occurrences of real collocations potentially formed by them. Google statistics of co-occurrences of any two strings with any *N* intermediate words in between can be gathered by queries in quotation marks containing these strings separated with *N* asterisks (* wildcard operators). So we intend to compare frequencies of the two kinds of events in the following way.

We take at random a ten of various commonly used collocations with unknown length of intermediate context. Then co-occurrence frequencies for each collocative pair are evaluated with *N* intermediate asterisks, $N = 0...5$. In large numbers, we cannot determine automatically whether counted co-occurrences are real collocations or merely a random encounters of words, possibly from different sentences. To evaluate the true portion (TP) of collocations in the automatically counted amounts, we look through the first hundred page headers with co-occurrences for various lengths of intermediate context, mentally analyzing their syntax. Multiplying the Google statistics GS by TP values, we got approximate collocation statistics (CS), cf. Table 2.

It can be seen that in the interval 0...5 GS has one or more local maximums, whereas the first local maximum of CS is disposed at 0, 1 or 2, and in nine cases of 10 it is unique, coinciding with the first local maximum of GS. So we can believe Google statistics in that the most probable distance between collocatives of real collocations corresponds to the first local maximum of GS, with the reservation that the both are searched in the interval [0...2] of intermediate context lengths (i.e. is in the interval [1..3] of distances between collocatives).

To explain the above-mentioned in more detail, the majority of collocative co-occurrences counted by the Web at the distances not exceeding 3 between collocatives are real collocations, whereas at the greater distances they are mostly coincidences of words, without direct syntactic links between them. This in no way means that collocations cannot have more distant collocatives, but the Web is not suited for collocation testing at greater distances, in contrast with collocation databases [1, 10] that are always applicable. The problem of the Web statistics validity for collocation testing deserves to be investigated deeper, but for this paper it was not required.

**Table 2.** Statistics of co-occurrences and collocations.

| Collocation | Stat. type | Number of intermediate words | | | | | |
|---|---|---|---|---|---|---|---|
| | | 0 | 1 | 2 | 3 | 4 | 5 |
| *act of … force* | GS | **6910** | 3590 | 5470 | 9800 | 10600 | 10200 |
| | TP | 0.99 | 0.67 | 0.14 | 0.07 | 0.00 | 0.00 |
| | CS | **6840** | 2400 | 765 | 686 | 0 | 0 |
| *main … goal* | GS | **1470000** | 37500 | 19700 | 17200 | 19900 | 16700 |
| | TP | 0.99 | 0.96 | 0.25 | 0.04 | 0.01 | 0.01 |
| | CS | **1455300** | 36000 | 4920 | 690 | 200 | 167 |
| *lift … veil* | GS | 17300 | **38400** | 5970 | 1880 | 993 | 438 |
| | TP | 1.00 | 1.00 | 0.93 | 0.88 | 0.64 | 0.44 |
| | CS | 17300 | **38400** | 5550 | 1650 | 635 | 193 |
| *moved with … grace* | GS | 557 | **4780** | 3930 | 1980 | 801 | 191 |
| | TP | 0.96 | 1.00 | 0.96 | 0.94 | 0.81 | 0.37 |
| | CS | 534 | **4780** | 3770 | 1860 | 648 | 71 |
| *give … message* | GS | 6040 | 106000 | **221000** | 156000 | 96600 | 145000 |
| | TP | 0.73 | 0.85 | 0.73 | 0.63 | 0.27 | 0.03 |
| | CS | 4410 | 90100 | **161300** | 98300 | 26100 | 4350 |
| *bridge across … river* | GS | 726 | 37800 | **52800** | 6530 | 1840 | 753 |
| | TP | 0.99 | 0.96 | 0.97 | 0.85 | 0.55 | 0.54 |
| | CS | 718 | 36300 | **51200** | 5550 | 1010 | 407 |
| *dump … waste* | GS | 12800 | **30500** | 16800 | 13100 | 12900 | 18400 |
| | TP | 0.85 | 0.81 | 0.43 | 0.10 | 0.15 | 0.11 |
| | CS | 10900 | **24700** | 7220 | 1310 | 1930 | 2024 |
| *needs … rest* | GS | 22300 | **75000** | 34600 | 29200 | 30100 | 30400 |
| | TP | 0.89 | 0.79 | 0.29 | 0.19 | 0.18 | 0.06 |
| | CS | 19850 | **59250** | 10030 | 5550 | 5420 | 1824 |
| *light … failed* | GS | 6360 | **14100** | 7750 | 7120 | 6630 | 7590 |
| | TP | 0.87 | 0.04 | 0.03 | 0.02 | 0.00 | 0.00 |
| | CS | **5530** | 564 | 232 | 142 | 00 | 00 |
| *partial … loss* | GS | **118000** | 28800 | 46000 | 22000 | 10900 | 10300 |
| | TP | 0.95 | 0.97 | 0.85 | 0.48 | 0.13 | 0.04 |
| | CS | **112100** | 27900 | 39100 | 10560 | 1420 | 412 |

## 4 Paronyms of Various Kinds

The notion of malapropisms, as well as of their correction candidates is based on some similarity between words. We can assume at least three overlapping kinds of word similarity: in letters, in sounds, and in morphs. Let us name any similar words paronyms. In all cases, we should consider paronyms of the same POS – for retaining the syntactic links between words while replacing a word by its paronym.

A letter paronym differs from its source word by a few elementary editing operations: insertion of a letter in any position, omission of a letter, replacing of a letter by another one, and permutation of two adjacent letters. E.g., word *pact* has 1-distant letter paronymy group {*act, fact, pack, pace, pant, part, pat, tact*}.

Phonetics assumes some phonetic representation of words, and phonetic paronym differs from its source word by a few editing operations on symbols of this representation. E.g., the adjective *sick* [sɪk] has 1-distant phonetic paronyms *thick* [θɪk], and the verb *seek* [si:k] has 1-distant paronym *sick* [sɪk].

Morphemic paronyms have the same root morph but differ in auxiliary morphs (prefixes or suffixes), e.g. the adjective *sens-ible* have morphemic paronym *sens-itive*.

In any language, only a limited portion of words have paronyms, and paronymy groups are on an average rather small. Hence, in contrast to spell checkers, it is reasonable to gather paronyms of all three kinds before their real use. The paronymy dictionaries already exist for Russian (letter and phonetic types, cf. [3]) and for Spanish (letter type, cf. [4]), but we are not aware of such dictionaries for English.

In this paper we involve only 1-distant letter paronymy, though for English the phonetic paronymy could be more important. We merely suppose that our method does not significantly depend on the kind of paronymy involved.

## 5 Algorithm for Malapropism Detection and Correction

The main idea of our algorithm is to look through all pairs of content words *W*(*i*) within a sentence under revision, testing each pair on its syntactic combinability and semantic compatibility. If the pair is syntactically combinable but semantically incompatible, a malapropism is signaled. (Since we consider malapropos words only as destroyers of collocations, below we will name malapropisms the complete erroneous word combinations.) Then all pairs formed by a collocative and its counterpart's paronym are tested on semantic compatibility. If a pair fails, it is discarded; otherwise it is included into a list of secondary candidates. Then the list is ranked and only the best candidates are kept. The following procedure revises a sentence:

```
Detect&Correct_Malapropisms
  for each W(i) in sentence repeat
    for each W(j)such that j < i
      if ContentWord(W(j)) & ContentWord(W(i))
        & SyntCombinable(W(j),W(i))
        & not SemCompatible(W(j),W(i)) then
        { ListOfPairs = ∅
          for each paronymy dictionary
            repeat  % for all paronyms of the left collocative
              TakeNextParonym(P,W(j))
              if SemAdmissible(P,W(i))then
                InsertToListOfPairs(P,W(i))
            until NoMoreParonymFor(W(j))
            repeat  % for all paronyms of the right collocative
              TakeNextParonym(P,W(i))
              if SemAdmissible(W(j),P))then
                InsertToListOfPairs(W(j),P)
              until NoMoreParonymFor(W(i))
          Filter(ListOfPairs)
          LetUserTest(ListOfPairs) }
```

Boolean function **SyntCombinable**(*V, W*) determines if the word pair (*V, W*) forms a syntactically correct word combination. Hence, it contains a partial dependency parser searching all those conceivable dependency subtrees with *V* and *W* at the extremes that do not contradict their immediate context (cf. subtrees in Table 1).

Boolean functions **SemCompatible**(*V,W*) and **SemAdmissible**(*V,W*) both determine if the pair (*V,W*) is semantically compatible. The procedure **Filter**(*ListOfPairs*) selects the best candidates. Operations of these three heavily depend on the available resource for collocation testing.

When the resource is a text corpus, **SemCompatible**(*V, W*) determines the number *N*(*V,W*) of co-occurrences of *V* and *W* in a limited distance from one another in the

whole corpus. If $N(V,W)$ equals zero, the function is *False*. If $N(V,W)$ is positive, for a definite decision it is necessary to syntactically analyze each co-occurrence, which is considered impractical in large numbers. In the case of ambiguity whether the co-occurrences are real collocations or mere coincidences in a text span, only statistical criteria are applicable. According to one criterion, the pair is compatible if the relative frequency $N(V,W)/S$ (i.e. empirical probability) of the co-occurrence is greater than the product of relative frequencies $N(V)/S$ and $N(W)/S$ of $V$ and $W$ taken separately ($S$ is the size of the corpus). Using logarithms, we obtain the following threshold rule of pair compatibility:

$$\mathrm{MII}(V,W) \equiv \ln((S \times N(V,W)) / (N(V) \times N(W))) > 0,$$

where $\mathrm{MII}(V, W)$ is the mutual information index [8].

For Web searchers only a statistical approach is possible. They automatically deliver statistics on queried words or word combination occurrences measured in Web-pages. We can re-conceptualize MII with all $N$ as numbers of relevant pages and $S$ as the page total managed by the searcher. However, now $N/S$ is not the empirical probability: the same words entering a page are counted only once, while the same page is counted repeatedly for each word included. Thus MII is not now grounded on a strict statistical model, only belief retains that $N/S$ is monotonically connected with the corresponding probability. Since MII depends on $N(V,W)$ and $N(V)\times N(W)$, we feel free to construe any other criteria from these 'building blocks.' One of them [8] is

$$\mathrm{MMII}(V,W) \equiv N(V,W) \times \ln((S \times N(V, W)) / (N(V) \times N(W))).$$

We prefer a different one under the name Semantic Compatibility Index (SCI):

$$\mathrm{SCI}(V,W) \equiv \ln\left(P \times N(V,W) \Big/ \sqrt{N(V) \times N(W)}\right),$$

where $P$ is a constant. Both SCI and MMII depend on $N(V,W)$ to more extent than MII. Just as MII, SCI retains its value when $N(V)$, $N(W)$, $N(V,W)$, and S change proportionally. This feature is especially important, since the total volume under searcher's control steadily increases and all measured values always fluctuate because of the inner evaluation strategy of the searcher. A particular property of SCI is the exclusion of $S$, so it becomes unnecessary to evaluate it repeatedly.

In order to avoid the logarithm of zeros, we take SCI in the form

$$\mathrm{SCI}(V,W) \equiv \begin{cases} \ln(N(V,W)) + \ln(P) - (\ln(N(V)) + \ln(N(W))) / 2, & \text{if } N(V,W) > 0, \\ NEG, & \text{if } N(V,W) = 0, \end{cases}$$

where *NEG* is a big negative constant; $P$ is positive constant to be chosen experimentally.

**SemCompatible** outputs *False* and thus signals the pair $(V_m,W_m)$ as malapropos if $\mathrm{SCI}(V_m,W_m)$ is negative, whereas **SemAdmissible** outputs *True* and admits the primary candidate $(V,W)$ as a secondary one if the SCI values for both candidate and its malapropism conform to the following threshold rule:

$$(\mathrm{SCI}(V_m,W_m) = NEG) \textbf{ and } (\mathrm{SCI}(V,W) > Q) \textbf{ or}$$
$$(\mathrm{SCI}(V_m,W_m) > NEG) \textbf{ and } (\mathrm{SCI}(V,W) > \mathrm{SCI}(V_m,W_m)),$$

where $Q$ ($NEG < Q < 0$) is a constant to be chosen experimentally.

**Filter** procedure operates with whole groups of secondary candidates, ranking them by SCI values. The best candidates are all with positive SCI (let be $n$ of them), whereas with a negative SCI value one more is admitted, if $n$=1, and two, if $n$=0.

## 6 An Experimental Set of Malapropisms

When a malapropism is detected in text, it is not initially known which collocative is erroneous, and we should try to correct both. The situation is clarified in Fig. 1. The upper two collocative nodes form malapropism. The nodes going south-west and south-east are paronyms of each malapropism's nodes. Each paronym should be matched against the opposite malapropism's node, and any pair could turn out to be sensible, but only one combination corresponds to the intended collocation; we call it the *true correction*.

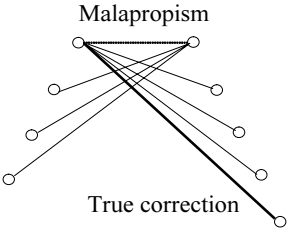



**Fig. 1.** Correction candidates and true correction.

Sometimes an error transforms one collocation to another semantically legal collocation, which may be rarer and in contradiction to the extra-collocational context, e.g., *normal* ***manner*** changed to *normal* ***banner*** or *give* ***message*** changed to *give* ***massage***. We name such errors quasi-malapropisms. Their detection (if possible) sometimes permits one to restore the intended words, just as for malapropisms proper.

We have gathered our experimental set in the following way. A hundred valid collocations were taken at random, mainly from [10]. Most of them are commonly used, but we also take few British expressions like *crown passed* and *royal crown*, to test our method in rarer cases. Collocatives in each collocation were then separated to their most probable distance in the way described in Section 3. Thereby the number of intermediate asterisks in the search pattern was determined for the whole group. Then one collocative in each pair was changed to another real word of the same POS through an elementary editing operation, thus forming one test malapropism. Other editing operations were then applied to the both components of the resulting test malapropism, one change for each correction candidate. Each resulting word combination was included in the set.

The beginning fragment of the resulting set is given in Fig. 2. It consists of enumerated groups (samples) with headlines containing malapropisms. After the group number the collocation subtype code goes (cf. Table 1). Then goes the position number of the erroneous collocative: 1 or 2, after which the quasi-malapropism case is marked with '**!**'. Next goes the malapropism string in quotation marks, somewhere having a short context in parentheses, only to help the mental correction.

The line with candidate corrections begins with the number of the changed word (1 or 2) and the case of the true correction is marked with '**!!**' (Such a candidate always enters the corresponding group).

In total, the hundred sections include 595 correction candidates, i.e. 5.95 primary candidates per error. The number of quasi-malapropisms equals 11.

```
  1)2.1 1 (without) "sighs of life"
    1   "sights of life"
    1!! "signs of life"
    2   "sighs of lie"
    2   "sighs of lime"
    2   "sighs of line"
    2   "sighs of lift"
    2!  "sighs of wife"
  2)4.3 1 (he) "hooked through**books"
    1   "booked through**books"
    1   "cooked through**books"
    1   "hooted through**books"
    1!! "looked through**books"
    1   "rooked through**books"
    2   "hooked through**cooks"
    2   "hooked through**hooks"
    2   "hooked through**looks"
    2   "hooked through**rooks"
  3)4.1 1 (to) "round***intentions"
    1   "bound***intentions"
    1   "ground***intentions"
    1!! "sound***intentions"
    2   "round***indentions"
    2   "round***intensions"
    2   "round***inventions"
  4)2.1 1 (get into) "pact of force"
    1!! "act of force"
    1   "fact of force"
    1   "pack of force"
    1   "pace of force"
    1   "pant of force"
    1   "part of force"
    1   "pat of force"
    1   "tact of force"
    2   "pact of forte"
  5)4.1 2 (to) "lead**precession"
    2   "lead**recession"
    2!! "lead**procession"
    1   "head**precession"
    1   "lend**precession"
    1   "leak**precession"
    1   "plead**precession"
```

**Fig. 2.** Several malapropisms and their correction candidates.

## 7 An Experiment with Google and Its Results

The beginning fragment of the experimental set supplied with statistics of occurrences and co-occurrences of the involved collocatives is given in Fig. 3. The Google session included 100×3+595×2=1490 accesses. All tentative multiword collocatives, even pseudo phrasal verbs, were found in the noisy Web.

Only 14 malapropisms caused zero-valued occurrences, while absurd primary candidates gave 166 zeros. Thus the raw Google statistics are not suited well for detecting malapropisms and eliminating their absurd corrections, so a further elaboration of the statistics is necessary.

```
  1)2.1 1 (without) "sighs of life"    146,sighs:1240000,life:416000000
    1   "sights of life"               382,sights:8510000
    1!! "signs of life"                418000,signs:48100000
    2   "sighs of lie"                 0,lie:7370000
    2   "sighs of lime"                0,lime:6010000
    2   "sighs of line"                0,line:398000000
    2   "sighs of lift"                0,lift:26200000
    2!  "sighs of wife"                21,wife:61800000
  2)4.3 1 (he) "hooked through**books" 118,"hooked through":8590,books:376000000
    1   "booked through**books"        2,"booked through":421000
    1   "cooked through**books"        4,"cooked through":183000
    1   "hooted through**books"        0,"hooted through":254
    1!! "looked through**books"        3910,"looked through":481000
    1   "rooked through**books"        0,"rooked through":1
    2   "hooked through**cooks"        0,cooks:4700000
    2   "hooked through**hooks"        1,hooks:6290000
    2   "hooked through**looks"        1,looks:62200000
    2   "hooked through**rooks"        0,rooks:487000
  3)4.1 1 (to) "round***intentions"    225,round:81800000,intentions:7780000
    1   "bound***intentions"           696,bound:36700000
    1   "ground***intentions"          685,ground:89900000
    1!! "sound***intentions"           1410,sound:139000000
    2   "round***indentions"           18,indentions:14100
    2   "round***intensions"           2,intensions:60700
    2   "round***inventions"           70,inventions:4030000
```

**Fig. 3.** Several malapropisms and their primary candidates with Google statistics.

To obtain all negative SCI values for all true malapropisms, we take $P = 4000$. The constant $NEG = -11$ is taken lower than SCI values for all positively counted events.

The constant $Q = -7.5$ is adjusted so that all candidates with non-zero occurrences have SCI values greater then this threshold. The distribution of SCI values (rounded to the nearest integers) for malapropisms and their true corrections is in Fig. 4. The peak for malapropisms is reached at –4, for their true corrections, between 2 and 3.

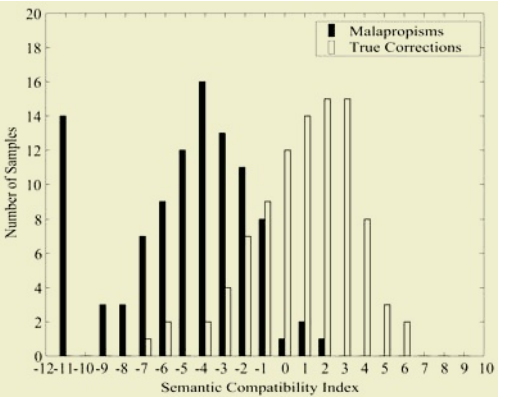



**Fig. 4.** Distribution of SCI for malapropisms and their true corrections.

Though none of the 11 quasi-malapropisms was taken into account while selecting the constant *P*, our algorithm detected eight of them as malapropisms proper: their SCI values are too low to be acknowledged as collocations. The three undetected were quite common pairs *let ** miss*, *give * massage* and *decide to refer*, replacing the intended *let ** kiss*, *give * message* and *decide to defer.* Hence our algorithm detects unintended real word errors with the precision 0.97 – for our experimental set.

**SemAdmissible** function leaves 247 secondary candidates of 595 primary ones (the decrease 2.41), while **Filter** procedure reduces them to 160 elite candidates (the total decrease is 3.72). So the lists of the best candidates contain usually two entries, cf. several malapropism groups with SCI values and decision qualifications in Fig. 5.

```
1)2.1 1 (without) "sighs of life"     -3.66 DETECTED
  1!! "signs of life"                  2.47 1ST CANDIDATE
2)4.3 1 (he) "hooked through**books" -1.34 DETECTED
  1!! "looked through**books"          0.15 1ST CANDIDATE
3)4.1 1 (to) "round***intentions"     -3.33 DETECTED
  1!! "sound***intentions"            -1.76 2ND CANDIDATE
  2   "round***indentions"            -1.53 1ST CANDIDATE
4)2.1 1 (get into) "pact of force"    -8.63 DETECTED
  1!! "act of force"                  -1.63 1ST CANDIDATE
  1   "part of force"                 -3.32 2ND CANDIDATE
5)4.1 2 (to) "lead**precession"       -2.83 DETECTED
  2   "lead**recession"                3.37 1ST CANDIDATE
  2!! "lead**procession"               3.36 2ND CANDIDATE
```

**Fig. 5.** Several malapropisms and best candidates with SCI values.

Among the best candidates for the detected 97 errors, there are 94 true corrections, and only four of them are not first-ranked. The occasional omission of a true correction does not seem too dangerous, since the user can restore it in the case of error detection. The most commonly used collocations among primary candidates always enter into the elite list, as true corrections or not. As many as 60 secondary candidates

propose changing the correct word in the pair, but in the elite their number diminished by more that 10 times. Hence the results of our experiment are rather promising: SCI is a good measure for detecting malapropisms and selecting their best correction candidates.

## 8 Conclusions and Future Work

A method is proposed for detection and correction of malapropisms. It is based on Google occurrence statistics recalculated as a novel numeric Semantic Compatibility Index for syntactically linked words (collocatives). Collocatives are always adjacent in embedding dependency trees, but sequentially they can be adjacent or rather distant within a sentence, in contrast to the well-known bigrams. The experiment was done on a set of a hundred malapropisms (including the so-called quasi-malapropisms, which are legitimate but unintended collocations and thus contradict the outer context) with their 595 primary correction candidates. As many as 97 of them were detected and for 94 of them their true correction candidates were entered highly ranked into the short lists of best candidates delivered for the user's consideration.

It would be worthwhile to extend (or test) the result of our study by (1) taking another experimental set; (2) changing the Web searcher; (3) involving phonetic and morphemic errors and corresponding paronymy dictionaries. Of course, it is would be helpful to develop a local dependency grammar parser sufficient for the proposed procedure of the malapropism detection & correction, since in this experiment we singled out collocation components manually.

Thus noisy and unsteady nature of the Web evaluations is moderated to some degree by our algorithm. Another serious flaw of the Web is its relative slowness. We may propose two ways out. The first is to request selling language-specific sectors of popular Web searcher's storage – to use them as local corpora, particularly for text correction. Indeed, progress in the size of hard disc memories has exceeded even the growth of the Web. The second way is to develop multistage systems including a collocation database of reasonable size, a large text corpus, and a Web-oriented part, so that the combined system accesses the Web only in those exceptional cases that are suggested by the collocation database supplied with some inference ability.

## References

1. Bolshakov, I.A. Getting One's First Million…Collocations. In: A. Gelbukh (Ed.). *Computational Linguistics and Intelligent Text Processing*. Proc. 5th Int. Conf. on Computational Linguistics CICLing-2004, Seoul, Korea, February 2004. LNCS 2945, Springer, 2004, p. 229-242.
2. Bolshakov, I.A., A. Gelbukh. On Detection of Malapropisms by Multistage Collocation Testing. In: A. Düsterhöft, B. Talheim (Eds.) Proc. 8th Int. Conference on Applications of Natural Language to Information Systems NLDB´2003, June 2003, Burg, Germany, GI-Edition, LNI, V. P-29, Bonn, 2003, p. 28-41.
3. Bolshakov, I.A., A. Gelbukh. Paronyms for Accelerated Correction of Semantic Errors. *International Journal on Information Theories & Applications.* V. 10, 2003, p. 198-204.
4. Gelbukh, A., I.A. Bolshakov. On Correction of Semantic Errors in Natural Language Texts with a Dictionary of Literal Paronyms. In: J. Favela et al. (Eds.) *Advances in Web Intelligence*. Proc. Int. Atlantic Web Intelligence Conf. AWIC 2004, Cancun, Mexico, May 2004. LNAI 3034, Springer, 2004, p. 105-114.

5. Keller, F., M. Lapata. Using the Web to Obtain Frequencies for Unseen Bigram. *Computational linguistics*, V. 29, No. 3, 2003, p. 459-484.
6. Kilgarriff, A., G. Grefenstette. Introduction to the Special Issue on the Web as Corpus. *Computational linguistics*, V. 29, No. 3, 2003, p. 333-347.
7. Hirst, G., D. St-Onge. Lexical Chains as Representation of Context for Detection and Corrections of Malapropisms. In: C. Fellbaum (ed.) *WordNet: An Electronic Lexical Database*. MIT Press, 1998, p. 305-332.
8. Manning, Ch. D., H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.
9. Mel'čuk, I. *Dependency Syntax: Theory and Practice*. SONY Press, NY, 1988.
10. *Oxford Collocations Dictionary for Students of English*. Oxford University Press, 2003.
11. Sekine, S., J.J. Carrol, S. Ananiadou, J. Tsujii. Automatic Learning for Semantic Collocation. Proc. 3rd Conf. ANLP, Trento, Italy, 1992, p. 104-110.
12. Wermter, J., U. Hahn. Collocation Extraction Based on Modifiability Statistics. Proc. 20th Int. Conf. on Computational Linguistics Coling'2004, Geneva, Switzerland, August 2004, p. 980-986.

# Web Directory Construction Using Lexical Chains

Sofia Stamou[1], Vlassis Krikos[1], Pavlos Kokosis[1],
Alexandros Ntoulas[2], and Dimitris Christodoulakis[1]

[1] Computer Technology Institute, Computer Engineering Department
Patras University, 26500 Patras, Greece
{stamou,dxri}@cti.gr, {krikos,kokosis}@ceid.upatras.gr
[2] Computer Science Department
University of California Los Angeles, USA
ntoulas@cs.ucla.edu

**Abstract.** Web Directories provide a way of locating relevant information on the Web. Typically, Web Directories rely on humans putting in significant time and effort into finding important pages on the Web and categorizing them in the Directory. In this paper we present a way for automating the creation of a Web Directory. At a high level, our method takes as input a subject hierarchy and a collection of pages. We first leverage a variety of lexical resources from the Natural Language Processing community to enrich our hierarchy. After that, we process the pages and identify sequences of important terms, which are referred to as lexical chains. Finally, we use the lexical chains in order to decide where in the enriched subject hierarchy we should assign every page. Our experimental results with real Web data show that our method is quite promising into assisting humans during page categorization.

## 1 Introduction

Millions of users today access the plentiful Web content to locate information that is of interest to them. However, the task of locating relevant information is becoming daunting as the Web grows larger. Currently, there are two predominant approaches that users follow in order to satisfy their information needs on the Web: searching and browsing [25]. During searching, the users visit a Web Search Engine (e.g. Google) and use an interface to specify a query which best describes what they are looking for. During browsing, the users visit a Web Directory (e.g. the Yahoo! Directory), which maintains the Web organized in subject hierarchies, and navigate through these hierarchies in the hope of locating the relevant information. The appearance of a variety of Web Directories in the last few years (such as the Yahoo! Directory [8], the Open Directory Project (ODP) [4], the Google Directory [1] etc.) indicates that the Web Directories are a popular means for locating information on the Web.

Typically, the information provided by a Web Search Engine is automatically collected from the Web without significant human intervention. However, the construction and maintenance of a Web Directory involves a staggering amount of human effort because it is necessary to assign an accurate subject to every page inside the Web Directory. To illustrate the size of the effort necessary, one can simply consider the fact that Dmoz, one of the largest Web Directories, relies on more than 65,000

A. Montoyo et al. (Eds.): NLDB 2005, LNCS 3513, pp. 138–149, 2005.
© Springer-Verlag Berlin Heidelberg 2005

volunteers around the world to locate and incorporate relevant information in the Directory. Given a Web page, one or more volunteers need to read it and understand its subject, and then examine Dmoz's existing Web Directory containing more than 590,000 subjects to find the best fit for the page. Clearly, if we could help the volunteers automate their tasks we would save a lot of time for a number of people.

One way to go about automating the volunteers' task of categorizing pages is to consider it as a classification problem. That is, given an existing hierarchy of subjects (say the Dmoz existing hierarchy) and a number of pages, we can use one of the many machine learning techniques to build a classifier which can potentially assign a subject to every Web page. One problem with this approach however, is that in general it requires a training set. That is, in order to build an effective classifier we need to first train it on a set of pages which has already been marked with a subject from the hierarchy. Typically this is not a big inconvenience if the collection that we need to classify and the hierarchy are static. As a matter of fact, as shown in [13, 15, 19, 22], this approach can be quite effective. However, in a practical situation, neither the Web [24] nor the subject hierarchies are static[1]. Therefore, in the case of the changing Web and subject hierarchy, one would need to recreate the training set and re-train the classifier every time a change was made.

In this paper, we present a novel approach for constructing a Web Directory which does not require a training set of pages and therefore can cope very easily with changes on the Web or the subject hierarchy. Our goal reaches beyond classification per se, and focuses on providing the means via which our hierarchy-based categorization model could be convenient in terms of both time and effort on behalf of Web cataloguers during page categorization. The only input that our method requires is the subject hierarchy from a Web Directory that one would like to use and the Web pages that one would like to assign to the Directory. At a very high level our method proceeds as follows: First we enrich the subject hierarchy of the Web Directory by leveraging a variety of resources created by the Natural Language Processing community and which are freely available. This process is discussed in Section 2. Then, we process the pages one by one and we identify the most important terms inside every page and we link them together, creating "lexical chains" which we will describe in Section 3. Finally, we use the enriched hierarchy and the lexical chains to compute one or more subjects to assign to every page, as shown in Section 4. After applying our method on a real Web Directory's hierarchy and a set of 114,000 Web pages we conclude that, in certain cases, our method has an accuracy of 87% into automatically assigning the Web pages to the same category that was selected by a human. Our results are presented in Section 5 and we conclude our work in Sections 6 and 7.

## 2 Building a Subject Hierarchy for the Web

Form a Web directory perspective, a subject hierarchy organizes a list of subjects, referred to as concepts, in successive ranks with the broadest listed first and with

[1] To see this one can simply compare Dmoz's subject hierarchies (file name: structure.rdf.u8.gz) in http://rdf.dmoz.org/rdf/archive

more specific aspects or subdivisions listed below. Typically, a hierarchy's concepts are depicted as nodes on a graph and the relations between concepts as arcs. Figure 1 shows a fraction of a hierarchy for the subject *Arts*, represented as a directed acyclic graph, where each node denotes a concept that is interconnected to other concepts via a specialization ("is-a") relation, represented by the dashed arcs. Concepts that are associated with a single parent concept are considered disjoint.

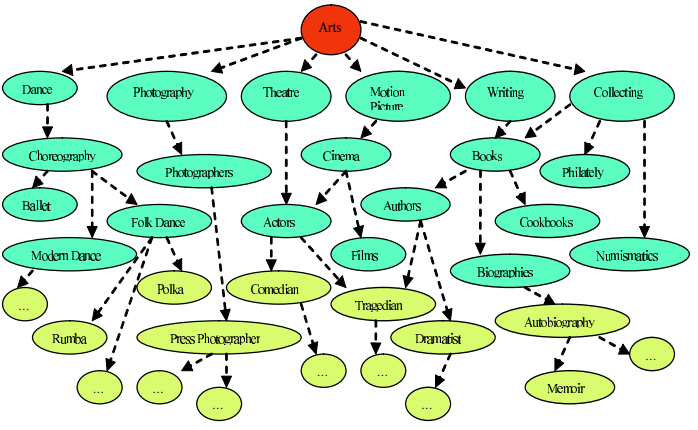



**Fig. 1.** A portion of the hierarchy for the *Arts* topic category (upper level topic) and subcategories (middle level concepts). The lower level nodes correspond to WordNet concepts.

For our purpose of generating a Web directory, we chose to develop a hierarchy of topics that are currently used by Web cataloguers in order to categorize Web pages thematically. To ensure that our hierarchy would both define concepts that are representative of the Web's topical content and be of good quality, we borrowed the hierarchy's top level concepts from the topic categories of the Google Directory and we enriched them with more specific concepts that we leveraged from existing ontological resources that have proved to be richly encoded and useful. The resources that we used for building our hierarchy are: (i) **The Suggested Upper Merged Ontology** (SUMO) [5]. SUMO is a generic ontology of more than 1,000 domain concepts that have been mapped to every WordNet synset that is related to them, (ii) **WordNet 2.0** [7]. WordNet is a lexical network of more than 118K synonym sets (synsets) that are linked to other synsets on the basis of their semantic properties and/or features, (iii) **MultiWordNet Domains** (MWND) [3]. MWND is an augmented version of WordNet; a resource that assigns every WordNet[2] synset a domain label among the total set of 165 hierarchically structured domains it consists of. Part of our hierarchy is illustrated in Figure 1. Our hierarchy has three different layers: the top layer corresponds to topics (Arts in our case); the middle layer to subtopics (e.g. Photography, Dance etc.) and the lower level corresponds to WordNet hierarchies. Our hierarchy can be downloaded from ftp://150.140.4.154/ftproot/.

[2] MWND labels were originally assigned to WordNet 1.6 synsets, but we augmented them to WordNet 2.0 using the mappings from http://www.cogsci.princeton.edu/~wn/links.shtml

### 2.1 Defining the Hierarchy's Concepts

To build our hierarchy we firstly enriched WordNet 2.0 with domain information. Note that a portion of WordNet 2.0 synsets are annotated with domain labels. For assigning domain labels to the remaining synsets, we leveraged domain knowledge from SUMO and MWND. To that end, we automatically appended (using available mappings) to every WordNet synset its corresponding domain label taken from either SUMO or MWND. Synsets of multiple domain assignments (i.e. synsets appended with a SUMO domain and a different MWND domain) were examined in order to select the domain that would best represent the synsets' semantics from a text categorization perspective. In selecting the most representative domain label among multiple domains, we adopted the Specification Marks technique, described in [27], which proceeds as follows. Given a group of terms that pertain to a domain category, we retrieve all their senses and supply them to a WSD module, which disambiguates them on the basis of the concept that is common to all the senses of all the words in this group. Disambiguated words are then supplied to a rules module, which locates the main-concept in WordNet for each of the domains, by using the hyper/hyponymy relation. The domain name used to label the main concept within a group of concepts is selected and propagated down to the WordNet terms that subsume the ISM concept, by following the hyponymy links. Finally, we manually examined the assigned domain labels and corrected any inconsistencies caused by erroneous disambiguation.

Having assigned a domain label to each WordNet synset, the next step we took was to define the hierarchy's top level topics. The hierarchy' top level concepts were chosen manually and they represent topics employed by Web cataloguers to categorize pages by subject. In selecting the topical categories we operated based on the requirement that our topics should be popular (or else useful) among the Web users. To that end, we borrowed 6 first level topics from the Google Directory taxonomy, thus satisfying our popularity requirement. These topics formed the hierarchy's root concepts and are the following: **Sports**, **Society**, **Sciences**, **Health**, **Arts** and **Recreation**. Following on, we integrated the WordNet hierarchies that have been enriched with domain information to their corresponding top level topics. Incorporating WordNet hierarchies into the six top level topics was carried out manually following an iterative process. The first straightforward step that we took was to select those WordNet hierarchies whose parent concept was labeled with a domain name identical to a top level topic (borrowed from the Google Directory) and automatically append them to the respective topic. The remaining hierarchies were manually appended to the hierarchy's top level topics based on their WordNet hypernymy relation. In particular, the WordNet hierarchies whose elements subsumed a top level topic were appended to this topic via an "is-a" relation. This way, those hierarchies' parent nodes become sub-domains in their corresponding 6 topics, denoting a middle level concept in the hierarchy. Following the steps described above, we integrated in our hierarchy's top level topics all WordNet lexical hierarchies for which a matching topic was found. At the end of the merging process, we came down to a total set of 143 middle level concepts, which were manually linked to the 6 top level topics, using their respective WordNet relations. The resulting upper level hierarchy (i.e. top and middle

level concepts) is a directed acyclic graph with maximum depth 4 and maximum branching factor, 28.

## 3 Reducing Pages to Lexical Chains

In this section we show how to leverage our hierarchy in order to detect which of the Web page's words are informative of the page's topic. We start our discussion by making the assumption that we process only Web pages written in English. Having automatically downloaded Web pages, we parse them to remove HTML markup, and we process the pages' contents by applying tokenization, stemming and part-of-speech tagging. Following, we eliminate stop-words from the pages and we compute a set of indexing keywords for every Web page. A driving factor in keywords' selection is to choose terms that express the pages' thematic content. Consequently, we need to account for the pages' lexical cohesion, i.e. the semantic relations that hold between the pages' terms. In selecting keywords, we augment the lexical chaining method introduced in [11, 18, 23], and for every Web page we automatically generate sequences of thematic words, i.e. sequences of semantically related terms.

The computational model we adopted for generating lexical chains is presented in the work of Barzilay [11] and it generates lexical chains in a three steps approach: (i) select a set of candidate terms[3] from the page, (ii) for each candidate term, find an appropriate chain relying on a relatedness criterion among members of the chains, and (iii) if it is found, insert the term in the chain and update accordingly. The relatedness factor in the second step is determined by the type of the links that are used in WordNet for connecting the candidate term to the terms that are already stored in existing lexical chains. Barzilay introduces also a "greedy" disambiguation algorithm that constructs all possible interpretations of the source text, using lexical chains.

However, Soung et al. [26] noted some caveats in this disambiguation formula in avoiding errors, because it does not consider anything about words that make a semantic relation. To surpass this limitation, they propose a new disambiguation formula, which relies on a scoring function *f*, which indicates a possibility that a word relation is a correct one. Given two words, $w_1$ and $w_2$, their scoring function *f* via a relation *r*, is calculated as the product of the words' association score, their depth in WordNet and their respective relation weight. The association score (*Assoc*) of the word pairs ($w_1$, $w_2$) is determined by the words' co-occurrence frequency in a corpus given by:

$$Assoc(w_1, w_2) = \frac{\log (p(w_1, w_2) + 1)}{N_s(w_1) \bullet N_s(w_2)} \mathrm{x} + \mathrm{y} = \mathrm{z} \quad (1)$$

where $p(w_1,w_2)$ is the corpus co-occurrence probability of the word pair $(w_1,w_2)$ and $N_s(w)$ is a normalization factor, which indicates the number of senses that a word w has. The words' ($w_1$, $w_2$) depth (*DepthScore*) expresses the words' position in WordNet hierarchy and demonstrates that the lower a word is in WordNet hierarchy, the more specific meaning it has. Depth score is defined as:

[3] Candidate terms are nouns, verbs, adjectives or adverbs.

$$DepthScore(w_1, w_2) = Depth(w_1)^2 \bullet Depth(w_2)^2 \quad (2)$$

where *Depth* (w) is the depth of word w in WordNet. Semantic relation weights (*RelationWeigh*) have been experimentally fixed to 1 for reiteration, 0.2 for synonymy, hyper/hyponymy, 0.3 for antonymy, 0.4 for mero/holonymy and 0.005 for siblings. Finally, the scoring function *f* of $w_1$ and $w_2$ is defined as:

$$f_s(w_1, w_2, r) = Assoc\ (w_1, w_2) \bullet DepthScore(w_1, w_2) \bullet \mathrm{Re}\, lationWeight(r) \quad (3)$$

The score of lexical chain $C_i$ that comprises $w_1$ and $w_2$, is calculated as the sum of the score of each relation $r_j$ in $C_i$. Formally:

$$Score(C_i) = \sum_{r_j\ in\ C_j} f_s\ (w_{j1}, w_{j2}, r_j) \quad (4)$$

To compute a single lexical chain for every downloaded Web page, we segment the latter into shingles, using the shingling technique, described in [12]. To form a shingle, we group *n* adjacent words of a page, with $n = 50$, which roughly corresponds to the number of words in a typical paragraph. For every shingle, we generate and score lexical chains using the formula described above. In case a shingle produces multiple lexical chains, the chain of the highest score is regarded as the most representative shingle's chain, eliminating hence chain ambiguities. We then compare the overlap between the elements of all shingles' lexical chains consecutively. Elements that are shared across chains are deleted so that lexical chains display no redundancy. The remaining elements are merged together into a single chain, representing the contents of the entire page, and a new Score ($C_i$) for the resulting chain $C_i$ is computed. This way we reassure that the overall score of every page's lexical chain is maximal. The elements of each chain are used as keywords for indexing the underlying pages in subject directories. In the subsequent paragraphs, we introduce a model that automatically categorizes Web pages into topics.

## 4 Assigning Web Pages to Topic Directories

In detecting the Web pages' topics, our model maps the pages' thematic keywords to the hierarchy's concepts, and traverses the hierarchy's matching nodes up to the root nodes. Recall that thematic words are disambiguated upon lexical chains' generation, ensuring that every keyword is mapped to a single node in the hierarchy. Traversal of the hierarchy's nodes accounts to following the hypermymic links of every matching concept until all their corresponding root topics are retrieved. For short documents with very narrow subjects there might be a single matching topic. However, due to both the sparseness of the Web data and the richness of our hierarchy, it is often the case that pages' thematic words correspond to multiple root topics. To accommodate multiple topics assignment, we compute a *Relatedness Score* (*RScore*) of every Web page to each of the hierarchy's matching topics. This relatedness score indicates how expressive is each of the hierarchy's topics for describing the Web pages' contents. Formally, the relatedness score of a page represented by the lexical chain $C_i$ to the hierarchy's topic $D_k$ is defined as the product of the chain's Score($C_i$) and the fraction of the chain's elements that belong to the category $D_k$. The *Relatedness Score* that a page has to each of the hierarchy's matching topics is given by:

$$RScore\ (\mathrm{i}, \mathrm{k}) = \frac{\mathrm{Score}(C_i) \bullet \#\text{of } C_i \text{ elements of } D_k \text{ matched}}{\left|\#\text{of } C_i \text{ elements}\right|} \tag{5}$$

The denominator is used to remove any effect the length of a lexical chain might have on *RScore* and ensures that the final score is normalized so that all values are between zero and one, with 0 corresponding to no relatedness at all and 1 indicating the category that is highly expressive of the page's topic. Finally, a Web page is indexed in the topical category $D_k$ for which it has the highest relatedness score of all its *RScores* above a threshold T, for T = 0.5. The page's indexing score is:

$$IScore\ (\mathrm{i}, \mathrm{k}) = \max RScore\ (\mathrm{i}, \mathrm{k}) \quad \text{where } 1 \le i \le T \tag{6}$$

Pages, whose chains' elements match several topics in the hierarchy, and whose relatedness scores to any of the matching topics are below T, are categorized in all their matching topics. By allowing pages to be indexed in multiple topics, we ensure there is no information loss during the directories' population and that pages with short content are not unquestionably discarded as less informative.

## 5 Experimental Study

To study the efficiency of our approach in populating Web directories, we conducted an experiment in which we supplied our model, named TODE, with 114K Web pages, inquiring that these are categorized in the appropriate topics in the hierarchy.

To ascertain that our perception of TODE's performance would not entail any bias, we elected to experiment with Web pages that had already been listed in topical categories by domain experts. In selecting the experimental data, we downloaded pages from those topics in Google Directory that matched any of the topics represented in our hierarchy, i.e. top level concepts. Downloading took a few days and we fetched only the pages that had been explicitly assigned to one of the six topics in Google Directory, without following the pages' internal links. By selecting pages from the Google Directory, we believe that our sample was popular and of good quality, which is implied by the large number of Google Directory users. In total, we downloaded 114,358 pages that span 91 Google Directory second level topics, which in turn are organized into 6 first level topics. Recall that the 6 first level topics in the Google Directory are the same with the top level topics in our hierarchy. The size of the downloaded data is nearly 5.9GB, which is reduced to 638MB when compressed. Table 1 shows the fraction of experimental pages in each topic in Google Directory.

**Table 1.** Distribution of Google Directory Topics in our Data.

| Topics | Fraction of pages in topic |
|---|---|
| Society | 29% |
| Arts | 25% |
| Sciences | 25% |
| Recreation | 10% |
| Sports | 9% |
| Health | 2% |

We parsed the downloaded pages and generated the shingles for them after removing the HTML markup. Pages were then tokenized, tagged, lemmatized and submit-

ted to TODE, which computed and weighted a single lexical chain for every page. To compute lexical chains, our model relied on a resources index which comprised: (i) the 12.6MB WordNet 2.0 data, (ii) a 0.5GB compressed corpus, which we processed in order to obtain statistical data about words' co-occurrence frequencies, and (iii) the 11MB upper level topics and subtopics in our hierarchy. Using the above data, TODE generated and scored lexical chains for every page; it computed the pages' *RScores* and *IScores* and stored this information in a secondary index. Based on a combined analysis of the data stored in the secondary index, TODE indicates the most appropriate hierarchy's (sub)-topic(s) to categorize each of the pages. At the end of the experiment, we compared the categorizations given by TODE for each of the pages to the categorizations these pages had in the respective Google Directory categories. Our comparison revealed that our model assigned 88,237 out of the 114,358 pages to a category and failed to deliver categorizations for the remaining 26,121 pages. Categorization failure was due to: (i) lack of textual data in the underlying pages; pages comprised lists of links, audiovisual data, etc., (ii) non-existent pages; dead links, redirects, (iii) frames in pages, (iv) downloading time-outs after 10 seconds, and (v) inefficiency in generating lexical chains for pages with very short content. The results presented below are based on the categorizations given for the 88,237 pages.

### 5.1 Directories' Population Performance

To evaluate our system's performance, we used as a comparison testbed the categorizations that our experimental data displayed in the respective Google Directory (sub)-topics. This is primarily because in Google Directory, pages have been manually assigned to topical categories, and secondly because of Google Directory's popularity, which stipulates that the offered categorizations have been found to be useful by many people. We first report the overall performance of our system in categorizing pages into topics, by comparing the fraction of pages that have been assigned to the same topics in both our hierarchy and Google Directory. Figure 2 plots the results.

In Figure 2, the horizontal axis represents the six top-level topics that are common between our hierarchy and the Google Directory. The vertical axis shows the fraction of pages that have been assigned to each of the topics, where 100% corresponds to the total number of pages for which TODE delivered categorizations, i.e. the 88,237 pages. For every topic, the solid/gray bars represent the fraction of pages categorized in that topic in Google Directory and the decorated bars represent the fraction of pages that have been categorized to that topic by TODE. The dark colored part of the decorated bars corresponds to the fraction of TODE's successful categorizations, i.e. the fraction of pages that have been assigned to the same topic by our system as in Google Directory, whereas the light colored part of the decorated bars corresponds to the fraction of pages mis-categorized by TODE. As mis-categorizations, we consider the pages that have been assigned by TODE to a topic, but which they are not assigned to the same topic in Google Directory. From the graph, we can see that in overall our system has a satisfactory performance in detecting a dominant topic of the Web pages, considering that the entire process was fully automated.

In order to have a clearer view on the obtained results, we give percentage values of the delivered categorizations in Table 2, whose first column shows the topics used in our experiment, the second column shows the fraction of the experimental pages that were assigned to each topic in Google Directory, the third column shows the fraction of pages that have been assigned to each topic in TODE and the forth column shows the fraction of the pages that have been "successfully" assigned to each of the topics in TODE, in the sense that these pages are also categorized in the same topics in Google Directory. By quantifying the amount of "successful" categorizations for all of the six topics together, we can see that our system had on average 75.1% categorization accuracy. Note that, categorization accuracy is defined as the fraction of the 88,237 pages that have been assigned to the same topic in our model as in Google Directory. Experimental results, verify that our system has a noticeable potential in assigning pages to topical categories, without imposing any need for human intervention, nor requiring training. Based on our experimental findings, we argue that our system could be employed as a Web cataloguers' assistant and deliver preliminary categorizations for Web pages. These preliminary categorizations could be then further examined by human cataloguers and reordered when necessary.

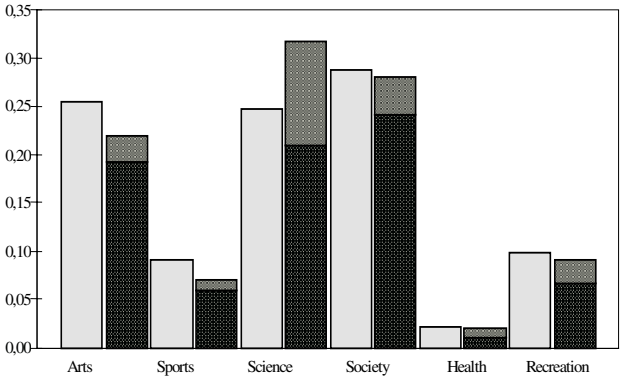



**Fig. 2.** Fraction of pages assigned to topics in Google Directory (solid/gray bars) and in TODE (decorated bars). Dark parts of the decorated bars correspond to the fraction of pages assigned to the same topic in both Google Directory and TODE while light colored parts correspond to the fraction of pages assigned to that topic in TODE but in another topic in Google Directory.

**Table 2.** Categorization Results.

| Topics | Pages in Google Directory | Pages in TODE | TODE pages compatible with Google Directory |
|---|---|---|---|
| Society | 29% | 28% | 86% |
| Arts | 25% | 22% | 87% |
| Sciences | 25% | 32% | 66% |
| Recreation | 10% | 9% | 73% |
| Sports | 9% | 7% | 85% |
| Health | 2% | 2% | 54% |

Our next experiment considered the use of the Google Directory hierarchies for categorizing the same 88,237 pages into the ontology's 6 top level topics. In particular, we appended to each of the ontology's 6 upper level topics their corresponding second and third level concepts in the Google Directory hierarchy, and we enriched the resulting ontology with lower level concepts that we borrowed from WordNet and which correspond to the hyponyms of the Google Directory concepts. We then incorporated this new ontology in TODE and we supplied our system with the 88,237 pages of our first experiment inquiring that these are categorized in the same 6 top level topics. Obtained categorizations were again compared to the categorizations the experimental pages have in the Google Directory. Our results confirm the potential that TODE has in successfully finding a dominant topic for categorizing Web pages, which accounts to an average categorization accuracy of 73%. Although TODE's performance figures might seem somehow low at first glance, however we believe that they are quite promising if we consider that the entire process was fully automatic, without the need of any training or human intervention.

## 6 Related Work

There has been previous work in categorizing Web pages into pre-defined topics [13, 14, 16, 22]. Related work falls into four categories. The first one concerns the hierarchical organization of the Web pages that are retrieved by search queries [15, 21]. This could also be addressed from a meta-search engine perspective, which aims to cluster together the pages retrieved in response to a query, based on either the contents of the pages [6], their links' structure [2], or both [16, 17]. Studies falling into this category rely significantly on the issued queries and the pages retrieved as relevant to the queries at hand. Our work differs from these studies, because we are not dealing with a subset of pages already deemed as relevant to a query (i.e. topic) by some searching mechanism. Instead, we aim at organizing Web pages by relying exclusively on the pages' thematic words and their semantic correlations. The second category concerns the automatic grouping of Web pages into personalized Web directories, based on user-profiling techniques [9, 10]. These approaches employ document clustering methods and usage mining techniques, to automate the process of organizing Web pages into topics. But, these techniques rely on a relatively small and precise set of "interesting" topics that are supplied to the various classification schemes as training paradigms. These training paradigms are determined by the users themselves, either in an explicit manner, by informing the system on their preferences (profiling), or implicitly, by having the system learn the users' profiles from their previous navigational behavior. Our approach for categorizing Web pages by topic is far more generic than personalized classification, in the sense that it is not bound to any particular information preferences and does not undergo any training phase. The third category relies on text classification techniques that group Web pages into pre-existing topics [14, 19]. In this approach, statistical techniques are used to learn a model based on a labeled set of training documents. This model is then applied to unlabeled pages to determine their topics. Again, the distinctive feature of our model from text classification techniques is the lack of a training phase. Finally, the objec-

tive in our work (i.e. directories' population) could be addressed from the agglomerative clustering [20] perspective; a technique that treats the generated agglomerate clusters as a topical hierarchy for clustering documents. The agglomerative clustering methods build the subject hierarchy at the same time as they generate the clusters of the documents. In our work, we preferred to build our hierarchy by using existing resources, rather than to rely on newly generated clusters, for which we would not have enough evidence to support their usefulness for Web pages' categorization.

## 7 Conclusion

We have presented a model that explores a subject hierarchy to automatically categorize Web pages in directory structures. Our approach extends beyond data classification and challenges issues pertaining to the Web pages' organization within directories and the quality of the categorizations delivered. We have experimentally studied the efficiency of our model in categorizing a fraction of Web pages into topical categories, and contrasted its resulting categorizations to the categorizations that the same pages displayed in the corresponding Google Directory categories. Our findings indicate that our model has a promising potential in facilitating current tendencies in editing and maintaining Web directories. It is our hope though, that our approach, will road the map for future improvements in populating Web directories and in handling the proliferating Web data.

## References

1. Google Directory http://dir.google.com.
2. Kartoo http://www.kartoo.com.
3. MultiWordNet Domains http://wndomains.itc.it/.
4. Open Directory Project http://dmoz.com.
5. Sumo Ontology http://ontology.teknowledge.com/.
6. Vivisimo http://www.vivisimo.com/.
7. WordNet 2.0 http://www.cogsci.princeton.edu/~wn/.
8. Yahoo! http://yahoo.com.
9. Yahoo! Inc. *MyYahoo* http://my.yahoo.com
10. Anderson C.R. and Horvitz E. Web montage: A dynamic personalized start page. In Proceedings of the 11th WWW Conference, 2002, 704-712.
11. Barzilay R. and Elhadad M. Lexical chains for text summarization. Master's Thesis, Ben-Gurion University, 1997.
12. Broder A.Z., Glassman S.C., Manasse M. and Zweig G. Syntactic clustering of the web. In Proceedings of the 6th WWW Conference, 1997, 1157-1166.
13. Chakrabarti S., Dom B., Agraval R. and Raghavan P. Scalable feature selection, classification and signature generation for organizing large text databases into hierarchical topic taxonomies. VLDB Journal, *7*, 1998, 163-178.
14. Chekuri C., Goldwasser M., Raghavan P. and Upfal E. Web search using automated classification. In Proceedings of the 6th WWW Conference, 1997.

15. Chen H. and Dumais S. Bringing order to the web: Automatically categorizing search results. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (2000), 145-152.
16. Halkidi M., Nguyen B., Varlamis I. and Vazirgiannis M. THESUS: Organizing web document collections based on link semantics. VLDB Journal, *12*, 2003, 320-332.
17. Haveliwala T. Topic sensitive PageRank. In Proceedings of the 11th WWW Conference, 2002, 517-526.
18. Hirst G. and St-Onge D. Lexical chains as representations of content for the detection and correction of malapropisms. In Fellbaum Ch. (ed.), WordNet: An Electronic Lexical Database. MIT Press, 1998, 305-332.
19. Huang C.C., Chuang S.L. and Chien L.K. LiveClassifier: Creating hierarchical text classifiers through web corpora. In Proceedings of the 13th WWW Conference, 2004, 184-192.
20. Kaufman L. and Rousseeuw P.J. Finding Groups in Data: An Introduction to Cluster Analysis. New York: John Wiley & sons, 1990.
21. Kummumuru K., Lotlikar R., Roy S., Singai K. and Krishnapuram R. A hierarchical monothetic document clustering algorithm for summarization and browsing search results. In Proceedings of the 13th WWW Conference, 2004, 658-655.
22. Mladenic D. Turning Yahoo into an automatic web page classifier. In Proceedings of the 13th European Conference on Artificial Intelligence, 1998, 473-474.
23. Morris J. and Hirst G. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. Computational Linguistics 17, 1, (1991), 21-43.
24. Ntoulas A., Cho J. and Olston Ch. What's new on the web? The evolution of the web from a search engine perspective. In Proceedings of the 13th WWW Conference, 2004, 1-12.
25. Olston Ch. and Chi E. ScentTrails: Intergrading browsing and searching. ACM Transactions on Computer-Human Interaction 10, 3 (Sept. 2003), 1-21.
26. Song Y.I., Han K.S. and Rim H.C. A term weighting method based on lexical chain for automatic summarization. In Proc. of the 5th CICLing Conference, 2004, 636-639.
27. Montoyo, A., Palomar, M., and Rigau, G. WordNet Enrichment with Classification Systems. In Proc. Of NAACL Workshop on WordNet and Other Lexical Resources: Applications, Extensions and Customization, 2001.

# Email Categorization with Tournament Methods

Yunqing Xia[1], Wei Liu[2], and Louise Guthrie[2]

[1] Department of Systems Engineering and Engineering Management,
The Chinese University of Hong Kong, Shatin, Hong Kong
yqxia@se.cuhk.edu.hk
[2] NLP Research Group, Department of Computer Science, University of Sheffield
Regent court, 211 Portobello Street, Sheffield, S10 4DP
{w.liu,l.guthrie}@dcs.shef.ac.uk

**Abstract.** To perform the task of email categorization, the tournament methods are proposed in this article in which the multi-class categorization process is broken down into a set of binary classification tasks. The methods of elimination tournament and Round Robin tournament are implemented and applied to classify emails within 15 folders. Substantial experiments are conducted to compare the effectiveness and robustness of the tournament methods against the n-way classification method. The experimental results prove that the tournament methods outperform the n-way method by 11.7% regarding precision, and the Round Robin performs slightly better than the Elimination tournament on average.

## 1 Introduction

Email is one of the most ubiquitous applications used on a daily basis by millions of people world-wide. Typically, emails are stored in different folders for easy access, imposing some structure on the increasingly unmanageable amount of information received by email. Our work is focused on creating better ways of categorizing email automatically. Email categorization aims to assign the incoming emails appropriate labels which are normally connected to the folders. The technology greatly facilitates email management by content or purpose, which is very helpful for email reading and future searching.

An adaptive email categorization system is implemented in [1] with a probabilistic classifier which assumes that each email folder is modeled so that the probability that email belongs to each folder can be computed. When a new email arrives, we compute the probability of that email coming from each of the folders and classify it to the folder that is most likely (Maximum Likelihood) [2]. This classification scheme is called n-way classification. In this paper we consider the n-way classification method as baseline and lay the tournament methods thereon. The question we ask is whether it is more effective and robust to use a combination of binary classifiers and perform the task of email categorization in tournament manners. We argue that combinations of binary classifiers can improve the classification quality because generally the combinations can decrease classification errors, especially when the training sets hold very different volume.

In this paper, we simulated the elimination tournament (ET) and Round Robin tournament (RRT). We apply the two methods to classify emails within 15 folders. We conduct substantial experiments to compare the tournament methods against the n-way classification method regarding effectiveness and robustness.

A. Montoyo et al. (Eds.): NLDB 2005, LNCS 3513, pp. 150–160, 2005.
© Springer-Verlag Berlin Heidelberg 2005

The remaining sections of this paper are organized as follow. In Section 2 we briefly survey the related works. In Section 3 we present an overview of the n-way probabilistic classification method. We address details of theory and algorithms within the tournament methods in Section 4. We describe our experiments and present results and discussions in Section 5. In Section 6 we conclude and address the future works.

## 2 Related Works

Various approaches towards text classification have been developed in the past few years, ranging from traditional document classification algorithms to newer spam-detection techniques [3]. The most popular approaches to text classification are Naïve Bayesian and related Bayesian learning methods [4, 5], probabilistic classifier[2], clustering techniques such as the k-nearest neighbor algorithm [6] and boosting [7], decision trees like the popular C4.5 algorithm [8], support vector machines [9], neural networks [10], statistical learning and induction methods [11], and voting [12, 13].

Several document classification methods have been applied to email categorization with mixed success [14, 15] and anti-spam filtering by classifying emails into folders [6]. An email categorization system is implemented in [1] with the probabilistic classifier proposed by [2], which assumes that all email folders are modeled so that the probability that the new email belongs to each folder can be computed.

The abovementioned methods vary concerning arrangement of training and classifying. Nevertheless, the common characteristic is that the methods perform classification task based on one round classifying. Thus classification quality can be achieved merely by refining the theory within the method. The voting method is an exception. It takes one of the aforementioned classifiers and a training set as input and trains the classifier multiple times on different versions of training sets [16]. In the classification process, the voting method performs multiple classifying and accumulates vote. The voting method sounds reasonable. However, training process is rather complicated and very time-consuming.

This paper aims to implement the tournament-like classification scheme based on combinations of binary classifiers. The multi-class categorization process is thus broken down into a set of binary classification tasks. We argue that classification quality can be improved with the tournament method because statistically the combinations can decrease classification errors.

## 3 The N-Way Classification

A probabilistic classification method is proposed in [2] which models all classes based on multinomial distribution and uses a maximum likelihood estimator to perform the text classification. As all classes are considered in the training and classifying process, we call this method n-way classification. Theory within the n-way classification method is described in Fig. 1.

Guthrie's theory is employed in [1] to implement an email categorization system. In this paper, based on the n-way classification method, we propose to explore more effective and robust text classification schemes, i.e. the tournament methods.

Given training populations $\{T_i\}_{i=1,2,\ldots,n}$, where population $T_i$ holds label $L_i$; by observing distribution of distinguished words contained in each population, the probabilistic classification model is construct as follows.

$$C(T_1, T_2, \ldots, T_n; W_1, W_2, \ldots, W_n; P_1, P_2, .., P_n)$$

where $W_i$ $(i=1,2,\ldots,n)$ denotes distinguished word set for population $T_i$; $P_j$ $(j=1,2,\ldots,n)$ denotes a distribution vector $(p_{i,j})_{j=1,2,\ldots,n}$ of word set $W_i$ within each population $D_j$.

Given a document $D=\{f_1, f_2, .., f_n\}$, where $f_i$ $(i=1,2,\ldots,n)$ denotes occurrence of words in document $D$ that appear in word set $W_i$, and $f_1 + f_2 + .. + f_n = N$, which is total word number of document $D$. The document will be assign a label that holds the maximum probability calculated by

$$\frac{N!}{f_1!f_2!\ldots f_n!(N-\sum_{k=1}^{n} f_k)!}(p_{i,1}^{f_1} p_{i,2}^{f_2} \ldots p_{i,n}^{f_n} (1-\sum_{j=1}^{n} p_{i,j})^{(N-\sum_{k=1}^{n} f_k)})$$

**Fig. 1.** Theory within the n-way classification method.

## 4 The Tournament Classification

With combinations of the binary classifiers, the tournament classification methods aim to perform classification on multi-class tasks using a tournament-like decision process, in which classes compete against each other to produce the most likely class. By adopting certain rules to combine the candidates within the competition, a multi-class classification process is broken down into a set of binary classification tasks. The most popular tournament classification implementations, inherited their name and rules from sports and games, are elimination tournament and Round Robin tournament. We apply the two tournament methods to perform the task of email categorization.

The tournament methods are different from the voting ones. Firstly, the voting method considers all classes in all training iteration while the tournament methods consider only two classes in each of the training iteration. Secondly, the voting method takes different versions of training set in each of the training iteration, but the tournament methods process the same training sets in all training iteration. However, the two methods do have commons due to the methodology nature. They both take multiple training and employ voting strategy in the classification process.

### 4.1 The Elimination Tournament Method

To describe the ET method, we first brief the competition rules employed in Wimbledon Championship. In this world famous tennis event, every player is required to compete against those players that determined by the sortition before the event kicks off. When one match is over, one of the two players obtains the chance in the next round competition and the unlucky one is eliminated.

The idea within elimination tournament classification is borrowed from the Wimbledon Championship. We take certain rules to perform category elimination. The winner class in the last round is selected to be the optimal class. To show a fair play spirit, the Wimbledon committee requires that every next round competition must not

start before this round finishes. However, this is rather unnecessary in tournament classification since machine always plays fairly. We simplify the competition rules and perform a random competition, which is illustrated in Fig. 2.

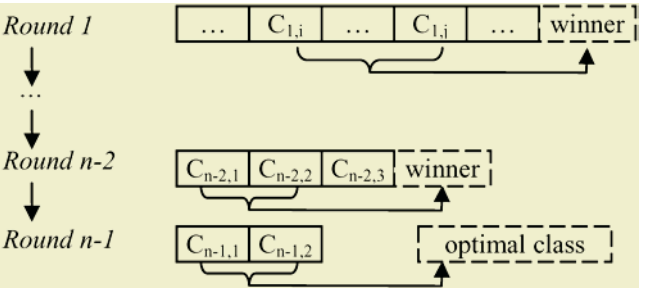



**Fig. 2.** Elimination rules in the elimination tournament method. *n* denotes number of classes within the text classification task . $C_{i,j}$ denotes class *j* that appears within Round *i*. The *winner* node denotes the winner class within the two opponent ones.

We initialize a competition list that contains all classes, and maintain the list after each run. Each time we randomly take two classes out of the list and classify the document with the binary classifier produced with the two classes. The loser class is eliminated and the winner is put back to the end of the list, thus we get a new competition list whose size is decreased by one after this run. We repeat this process until only one class remains in the competition list, which is considered the optimal class that the document most likely belongs to.

### 4.2 The Round Robin Tournament Method

Instead of eliminating the losers, the RRT method assigns scores to both classes after each competition; therefore, every class will accumulate a total score after all the binary classification tasks are over. The one that holds the highest score is selected as the optimal class that the input document most likely belongs to. This operation follows the competitions rules within the English Premiership. As an illustration, every class can be considered as one soccer team, and the team who wins the cup is then considered as the class that the input document most likely belongs to.

In the RRT classification process, we run binary classification processes between every two classes with the binary classifier produced with two classes. A competition table, which is exactly the same one as we see in English Premiership, is maintained by the system to record the winners and losers. The score table is then derived from the competition table by assigning the winner, loser or level classes with different scores. In the RRT experiments, two scoring schemes are designed in Table 1.

**Table 1.** Scoring schemes in the RRT method. The scoring scheme *S1* is exactly the same scoring scheme as English Premiership. In scheme *S2* a punishment score, i.e. -1, is assigned to the loser.

| Scheme | Winner | Loser | Level |
|---|---|---|---|
| *S1* | 3 | 0 | 1 |
| *S2* | 1 | -1 | 0 |

The empirical study shows that *S1* is more accurate in determining the best category which accords the tradition of the English Premiership. Thus we use the scoring scheme *S1* in the later experiments. Suppose we have 5 classes, the score table for one input document is illustrated in Table 2.

**Table 2.** An illustrative score table employed in the RRT method. $C_i(i=1,...,5)$ denotes class *i*. In this table class $C_2$ holds a score of 10 after the RRT competition, thus is selected as the optimal class that the input document most likely belongs to.

| | $C_1$ | $C_2$ | $C_3$ | $C_4$ | $C_5$ | ***Total score*** |
|---|---|---|---|---|---|---|
| $C_1$ | - | 0 | 1 | 3 | 3 | **7** |
| $C_2$ | 3 | - | 3 | 1 | 3 | **10** |
| $C_3$ | 1 | 0 | - | 3 | 0 | **4** |
| $C_4$ | 0 | 1 | 0 | - | 1 | **2** |
| $C_5$ | 0 | 0 | 3 | 1 | - | **4** |

It is worth noting that, under RRT method, a tie occurs very likely in the final score table, that is, two or more classes end up with the same score. One simple approach to resolve the tie problem is "random pick", in which a class is randomly picked out from classes within the tie. This is rather crude in resolving the tie problem. We use a more comprehensive approach. When a tie occurs within the Round Robin tournament, we perform tie-breaking operation as follows. If it is a two-way tie, say, class *i* and class *j*, we then check the score table to find the competition which has ever been performed between the two classes, and the final class will simply be the one whichever wins. If there is a three-way tie or more, we first check the score table to find the tie classes, and the final class is the one holding the most winning rounds. If the tie still exists, we then degenerate the tournament method to n-way method, and output the class whichever wins within the n-way classification.

### 4.3 Comparisons

Firstly, we compare the tournament methods against the n-way method. On one hand, the tournament methods are more considerate than the n-way method in performing the classification task. We believe they hold higher accuracy than the n-way method because error rate can be decreased per combinations of binary classifiers. Most Errors are caused by significantly different volume of training sets from some classes. Since the binary classification is pair-wise [17], when volume of training sets varies greatly, error occurs merely within local competitions. Instead, the n-way method considers all classes in one round training or classifying. When error occurs within n-way method, it must be globally, which seriously decouples the classification quality.

On the other hand, when complexity is concerned, the tournament methods are more complicated, thus require more time and memory than the n-way method in performing a real classification task. In tournament methods, not only the classification requires multiple rounds, but the training program needs construct all possible binary classifiers. Although the training process can be carried out beforehand, classification speed is still lower than that of the n-way method.

Secondly, we observed difference between the two tournament methods. Basically, the RRT method is more reasonable than the ET method. As a comparison, there are

more black horses showing up within Wimbledon than other sport games, say, soccer, because the competition rules within Wimbledon follow elimination tournament. Those who warm up slowly could be likely eliminated in some rigorous competition. But with the Round Robin tournament competition rules, this kind of unlucky tragedy can be avoided to great extent. This is also true for tournament classification. Although text classification is nothing human, rigorous competition does exist, say, two classes holding similar content. Thus we believe the RRT method can outperform the ET method. However, algorithm for the RRT method is more complex that that of the ET method. Thus the RRT method requires more execution time.

We prove all the claims by conducting substantial experiments in performing the talks of email categorization in the following section.

## 5 Experiments

We conduct substantial experiments with both n-way and tournament methods to perform the talk of email categorization.

### 5.1 Corpus Description

To the best of our knowledge, there is no golden standard so far regarding email corpora for categorization evaluation purpose. An email collection is constructed in [1] to evaluate the FASiL email categorization system. 4,512 emails within 50 folders are contributed from ten users in which three of them are university researchers and seven are system evaluation experts. We simply mix all the folders up and extract the biggest 15 folders regarding number of emails the folders contain. It is worth noting that folder *Interno* is contributed from a user working in Portugal. Emails within this folder are most written in Portuguese. The corpus used in our experiment is described in Table 3.

**Table 3.** The corpus used in the experiments.

| No | Folder name | # of emails | # of words |
|---|---|---|---|
| 1 | Recruits | 1,039 | 317,454 |
| 2 | FASiL | 691 | 221,553 |
| 3 | Personal | 432 | 79,517 |
| 4 | Digital | 380 | 134,504 |
| 5 | Books | 339 | 98,298 |
| 6 | UNIV | 337 | 55,098 |
| 7 | Discuss | 335 | 65,975 |
| 8 | Corpora | 218 | 92,371 |
| 9 | Conf | 170 | 90,086 |
| 10 | PhD | 125 | 46,761 |
| 11 | FYA | 122 | 43,256 |
| 12 | FYI | 117 | 40,263 |
| 13 | Speech | 107 | 155,820 |
| 14 | Tech | 81 | 61,904 |
| 15 | Interno (Portuguese) | 19 | 4,111 |
| | **Total** | **4,512** | **1,506,971** |

Email with MIME format often contains additional information which has nothing to do with email content. Fortunately, the noises occur regularly within the email text and a filtering tool is easily developed to delete them.

Most emails are composed by replying to an original email, often including part or whole of the original email together with new content. The first email in the thread will potentially be repeated many times over, which might mislead the training process. A thread-detection filtering tool is used to eliminate unoriginal content in the email. Stylistic features like the presence of a greater than symbol ('>') at the beginning of a line, or the presence of an old email header are also used to determine whether a particular section of the email should be filtered out. An exception is that we keep the original email where segments of the new email are embedded.

### 5.2 Experiment Description

The split-sample validation method is introduced in our experiment to arrange the training and test sets. We split the email collection into training and test sets by random picking up. We use twenty percent emails within every folder as test set and the remaining emails as training set.

In our experiments, the training process involves producing all binary classifiers which combine every two folders. We construct each binary classifier with the n-way classification method. After all the binary classifiers are generated, we then perform categorization process on the test set with the two tournament methods.

We consider n-way classification method as the baseline method, and perform experiments to observe how significant the two tournament methods outperform. We also compare the two tournament methods according to the experiment results.

### 5.3 Evaluation Criteria

We use precision, recall and F-1 measure [11] to evaluate the methods. We also use Micro-average and Macro-average for precision, recall, and $F_1$-measure to present the overall system performance.

Suppose we have *n* folders, and $N_i^u (i = 1,2,...,n)$ emails in folder *i* which are categorized by user manually. The program classify $N_i^p$ emails to folder *i*, in which $N_i^{cp}$ emails that are correctly categorized. We define the evaluation criteria as follow.

$$precision(i) = \frac{N_i^{cp}}{N_i^u} \quad recall(i) = \frac{N_i^{cp}}{N_i^p} \tag{1}$$

$$F_1(i) = \frac{2 \times precision(i) \times recall(i)}{precision(i) + recall(i)} \tag{2}$$

$$Micro\_avg(precision) = \frac{\sum_{i=1}^{m} N_i^{cp}}{\sum_{i=1}^{m} N_i^p} \tag{3}$$

$$Micro\_avg(recall) = \frac{\sum_{i=1}^{m} N_i^{cp}}{\sum_{i=1}^{m} N_i^u} \tag{4}$$

$$Macro\_avg(precision) = \frac{1}{m}\sum_{i=1}^{m} precision(i) \tag{5}$$

$$Macro\_avg(recall) = \frac{1}{m}\sum_{i=1}^{m} recall(i) \tag{6}$$

$$Micro\_avg(F_1) = \frac{2 \times Micro\_avg(precision) \times Micro\_avg(recall)}{Micro\_avg(precision) + Micro\_avg(recall)} \tag{7}$$

$$Macro\_avg(F_1) = \frac{2 \times Macro\_avg(precision) \times Macro\_avg(recall)}{Macro\_avg(precision) + Macro\_avg(recall)} \tag{8}$$

### 5.4 Experimental Results

We first train the n-way method with the training sets and perform classification on the test sets with the n-way classifier. We then run the two tournament methods on the same training and test sets. We present precision, recall and $F_1$-measure for n-way, elimination tournament, and Round Robin tournament in Table 4. Precision and recall are calculated with formula (1). $F_1$-measure is calculated with formula (2). Micro-average precision and recall are calculated with formula (3) and (4). Macro-average precision and recall are calculated with formula (5) and (6). Micro-average and Macro-average $F_1$-measure are calculated with formula (7) and (8).

**Table 4.** Precision, recall and $F_1$-measure for n-way, ET and RRT methods. PRE denotes precision, REC denotes recall, and F1 denotes $F_1$-measure.

| No | Folder name | n-way | | | ET | | | RRT | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | PRE | REC | F1 | PRE | REC | F1 | PRE | REC | F1 |
| 1 | Recruits | 0.911 | 0.740 | 0.817 | 0.860 | 0.918 | 0.888 | 0.864 | 0.918 | 0.890 |
| 2 | FASiL | 0.951 | 0.986 | 0.968 | 0.985 | 0.928 | 0.956 | 0.985 | 0.928 | 0.956 |
| 3 | Personal | 0.945 | 0.989 | 0.966 | 0.988 | 0.943 | 0.965 | 0.988 | 0.943 | 0.965 |
| 4 | Digital | 0.961 | 0.961 | 0.961 | 0.986 | 0.934 | 0.959 | 0.986 | 0.934 | 0.959 |
| 5 | Books | 0.867 | 0.956 | 0.909 | 0.944 | 0.985 | 0.964 | 0.944 | 0.985 | 0.964 |
| 6 | UNIV | 0.905 | 1.000 | 0.950 | 0.984 | 0.940 | 0.962 | 0.955 | 0.955 | 0.955 |
| 7 | Discuss | 0 | 0 | 0 | 1.00 | 0.940 | 0.969 | 1.00 | 0.940 | 0.969 |
| 8 | Corpora | 0.667 | 0.955 | 0.785 | 0.740 | 0.841 | 0.787 | 0.740 | 0.841 | 0.787 |
| 9 | Conf | 0.345 | 0.576 | 0.432 | 0.476 | 0.606 | 0.533 | 0.476 | 0.606 | 0.533 |
| 10 | PhD | 0 | 0 | 0 | 0.857 | 0.480 | 0.615 | 0.857 | 0.480 | 0.615 |
| 11 | FYA | 0.107 | 0.320 | 0.160 | 0.500 | 0.560 | 0.528 | 0.538 | 0.560 | 0.549 |
| 12 | FYI | 0.280 | 0.292 | 0.286 | 0.579 | 0.458 | 0.512 | 0.579 | 0.458 | 0.512 |
| 13 | Speech | 0.583 | 0.955 | 0.724 | 0.808 | 0.955 | 0.875 | 0.808 | 0.955 | 0.875 |
| 14 | Tech | 0.739 | 1.000 | 0.850 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| 15 | Interno | 0 | 0 | 0 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| Micro_Avg | | 0.768 | 0.859 | 0.811 | 0.885 | 0.885 | 0.885 | 0.886 | 0.886 | 0.886 |
| Macro_Avg | | 0.689 | 0.811 | 0.745 | 0.847 | 0.833 | 0.840 | 0.848 | 0.834 | 0.841 |

### 5.5 Discussion 1: N-Way vs. Tournament

According to Table 4, both tournament methods outperform the n-way method by more than 11.7% regarding the Micro-average precision. For most folders the tournament methods outperform the n-way method. This proves that the tournament methods improve classification quality significantly indeed.

We believe the improvement is due to a lower error rate that occurs in tournament methods. Carefully looking into classification log files produced by the programs, we find that the tournament methods maintain the correct classification decisions by n-way method, and correct some errors that occur within n-way method. This proves our argument that the tournament methods are more effective than the n-way method.

According to the precision curves showed in Fig. 3, we find that the n-way method perform rather abnormal within this email collection. For example, precision for class *Discuss*, *PhD* and *Interno* are zero, and three classes, i.e. *Conf*, *FYA* and *FYI*, hold precisions lower than 0.4. In class *Personal*, the n-way method surprisingly outperforms the tournament methods by 5%. The unstable classification quality is caused by nature of the n-way method. When volume of training sets varies significantly, performance of the n-way method is unpredictable.

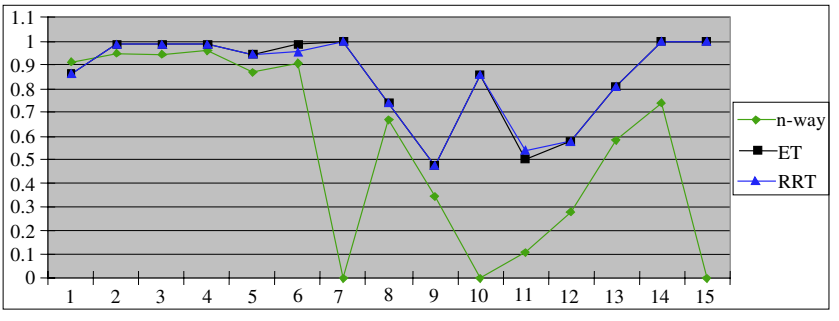



**Fig. 3.** Precision curves for n-way, ET method and RRT method. The X axis denotes the 15 folders, and the Y axis denotes classification precision.

Relatively, the tournament methods yield much more robust classification precision, in which precisions range from 0.47 to 1.0. Thus we conclude that the tournament methods are more robust than the n-way method in performing the task of email categorization.

### 5.6 Discussion 2: Elimination vs. Round Robin

To compare the ET method and RRT method, we enlarge their precision curves in Fig. 4. According to this figure, the two tournament methods perform very close to each other in this email collection. Precisions for most classes are same except that the ET method performs slightly better in class *UNIV* and the RRT method performs slightly better in class *FYA*.

We carefully observe the categorization log file for the elimination tournament. We find some test emails from class *UNIV* happen to be classified with favorable combinations so that other classes are easily eliminated. However, the log file for the Round Robin tournament shows that those emails confuse most binary classifiers and the total score for class *UNIV* is low. This explains why the ET method performs better in class *UNIV*. For emails from class *FYA,* the log files present another story. Some *FYA* emails encounter rigorous combinations, thus suffer tragedies and get eliminated. While in the Round Robin tournament, these emails show clear distinctions, thus accumulate highest scores.

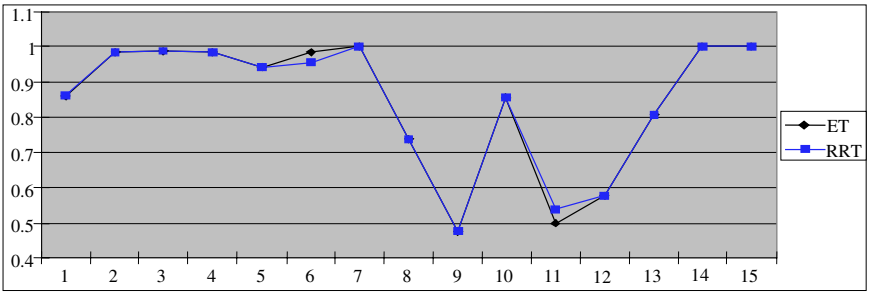



**Fig. 4.** Enlarged precision curves for ET method and RRT method. The X axis denotes the 15 folders, and the Y axis denotes classification precision.

The two cases indicate that combination rules within the elimination tournament are rather critical. Since we use random combination rules, rigorous combinations are inevitable thus errors occur. On the other hand, email that holds indistinct content can confuse the Round Robin tournament, thus some incorrect binary classifications lead to error.

But still, according to the experimental results, the tournament methods perform over 88% Micro-average precision than the n-way method, and the RRT method outperforms the ET method slightly in terms of overall quality according to Table 4.

## 6 Conclusions and Future Works

We propose to deploy the tournament methods to perform the task of email categorization. We conduct substantial experiments to evaluate the two methods. The experimental results reveal that the tournament methods outperform the n-way method by 11.7% regarding precision, thus prove that the tournament methods are much more effective and robust than the n-way method. Another conclusion can be drawn from the experimental results is that the RRT method outperforms ET method slightly.

In this paper we lay the baseline method to be a probabilistic classifier and prove that the tournament methods outperform the n-way method significantly. It is reasonable to assume that classification quality can be improved as well when another classification method, say, Naïve Bayesian, is adopted as the baseline classification method. We will prove this assumption in our future works.

To obtain both better performance and lower execution cost, future works will be carried out to generate the optimal competition rules. Instead of combining two classes randomly, the future method will choose to reject the combination of similar classes unless there is no other category.

Another notable research is the combined tournament methods which are similar to the rules within the World Cup Final. Classes are first grouped and each group contains four classes. Then Round Robin tournament is applied within each groups and produces winner of the group. At last the winner classes from all groups compete against each other with the elimination tournament to produce the final winner. The difficult point exists in class grouping, i.e. producing the optimal grouping scheme, in which a further study of the binary classifiers is indispensable.

## Acknowledgement

Research presented in this article is partially supported by EU FASiL project (IST-2001-38685). We'd also like to thank Dr J. Guthrie for the valuable suggestions on improving tie breaking within RRT method.

## References

1. Xia, Y., Dalli, A., Wilks, Y. and Guthrie, L.: FASiL Adaptive Email Categorization System. Proc. of CICLing-2005, LNCS 3406, Mexico City, Mexico (2005)718–729.
2. Guthrie, L., Walker, E. and Guthrie, J.: Document Classification by machine: Theory and practice. Proc. COLING'94, (1994)1059-1063.
3. Smadja, F. and Tumblin, H.: Automatic Spam Detection as a Text Classification Task. Elron Software (2003).
4. Lewis, D.: Naive Bayes at forty: The independence assumption in information retrieval. Proc. ECML-98, Chemnitz. (1998)4-15.
5. McCallum, A. and Nigam, K.: A comparison of event models for naive bayes text classification. AAAI-98 Workshop on Text Categorization (1998).
6. Androutsopoulos, Koutsias, I., J. Chandrinos, Paliouras, G. K. V. and Spyropoulos, C. D.: An Evaluation of Naive Bayesian Anti-Spam Filtering. Proc. of the workshop on Machine Learning in the New Information Age (2000).
7. Carrerras, X. and Marquez, L.: Boosting Trees for Anti-Spam Email Filtering. Proc. RANLP-2001 (2001).
8. Quinlan, J.R.:C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo (1993).
9. Thorsten, J.: A Statistical Learning Model of Text Classification with Support Vector Machines. Proc. of SIGIR-01, New Orleans. ACM Press, New York (2001).
10. Wiener, Pederson, E. J.O. and Weigend, A.S.: A neural network approach to topic spotting. Proc. SDAIR-95, Nevada, Las Vegas (1995)317-332.
11. Yang, Y.: An evaluation of statistical approaches to text categorization. Journal of Information Retrieval. 1(1/2)(1999)67-88.
12. Breiman, B.: Bagging predictors, Machine Learning, v.24 n.2, (1996 )123-140, Aug.
13. Freund, Y. and Schapire, R. E.: Experiments with a New Boosting Algorithm, in Proceedings of the 13th International Conference on Machine Learning, (1996) 325-332.
14. Cohen, W.: Learning Rules that Classify EMail. Proc. AAAI Spring Symposium on Machine Learning in Information Access, Stanford, California (1996).
15. Payne, T. and Edwards, P.: Interface Agents that Learn: An Investigation of Learning Issues in a Mail Agent Interface. Applied Artificial Intelligence Journal, AUCS/TR9508 (1997).
16. Aas, L. and Eikvil, L. Text categorisation: A survey. Norwegian Computing Center, Raport NR 941 (1999).
17. Fürnkranz, J.: Round Robin Classification. Journal of Machine Learning Research 2 (2002) 21-747.

# Knowledge-Based Information Extraction: A Case Study of Recognizing Emails of Nigerian Frauds

Yanbin Gao[1] and Gang Zhao[2]

[1] Advanced System Development Co, Ltd, Beijing, China
yanbin_gao@asdc.com.cn
[2] STARLab, Computer Science Department, Vrije Universiteit Brussel, Belgium
gang.zhao@vub.ac.be

**Abstract.** This paper describes the methodology, process and results of developing an application ontology as software specification of the semantics of forensics in the email suspicious of Nigerian frauds. Real life examples of fraud emails are analyzed for evidence and red flags to capture the underlying domain semantics with an application ontology of frauds. A model of the natural language structure in regular expressions is developed in the light of the ontology and applied to emails to extract linguistic evidences of frauds. The evaluation of the initial results shows a satisfactory recognition as an automatic fraud alert system. It also demonstrates a methodological significance: the methodical conceptual modeling and specific purpose-driven linguistic modeling are effective in encapsulating and managing their respective needs, perspectives and variability in real life linguistic processing applications.

## 1 Introduction

This paper presents a study of developing an ontology as a specification of application semantics for linguistic engineering and processing. The application domain is the detection of the emails suspicious of Nigerian frauds. The ontology is a knowledge model of the fraud forensics. It serves as conceptual basis for specifying linguistic rules in regular expressions and information extraction to recognize fraud evidences from texts, samples of Nigerian fraud emails.

The paper introduces the ontology development methodology (Section 2), describes its development process (Section 3) and the linguistic engineering with respect to the resultant ontology (Section 4), analyses the experiment results of evidence recognition (Section 5) and concludes with a summary (Section 6).

## 2 Ontology Modeling Approach

We make a methodological distinction between two viewpoints in knowledge engineering for ontology-based information extraction: application semantic model vs. linguistic model. The application semantics is described as a conceptual model of basic concepts and relationships in the problem domain. Though the intended solution of information extraction is a software artifact of natural language processing, we recognize the necessity for a problem-oriented semantic model, independent of linguistic considerations, and the solution-oriented natural language models built thereupon. Besides the engineering need for a modular approach and complexity manage-

A. Montoyo et al. (Eds.): NLDB 2005, LNCS 3513, pp. 161–172, 2005.
© Springer-Verlag Berlin Heidelberg 2005

ment, it is important to capture the recurrent essential qualities of frauds underlying highly varied and dynamic linguistic expressions, in order to encapsulate and keep up with changes and evolutions efficiently.

The ontology to create is not an upper or base or foundation ontology, neither is it a domain ontology. It is an application specific ontology of the fraud forensics for recognizing Nigerian advance fee frauds. It is a part of or extension of a topical ontology of fraud forensics. It is specific in conceptual perspectives and relevance across multiple domain ontologies. Its purpose at this point is to describe what rather than how.

### 2.1 Ontology Representative Framework

The DOGMA (**D**eveloping **O**ntology-**G**uided **M**ediation for **A**gent) ontology representation framework defines an ontology as a set of *lexons* and their commitments in particular applications. A lexon is defined as 5-tuples, <*Context, Term*$_1$*, Role*$_1$*, Term*$_2$*, Role*$_2$ >. It represents a fact type: a relationship type between two object types (*Term*$_1$ and *Term*$_2$ playing *Role*$_1$, and *Role*$_2$) [3] [5]. While lexons capture the underlying concepts and relationships, the commitment ground them to a particular application or task requirement [2] [9] with specific constraints and instantiations.

### 2.2 Ontology Engineering Methodology

AKEM (**A**pplication **K**nowledge **E**ngineering **M**ethodology) is devoted to ontology-based knowledge engineering [8] [9] [10]. Based on the DOGMA ontology representation framework, it is designed for multi-disciplinary geographically distributed team of knowledge engineering. It follows a lifecycle model similar to RUP [4] through the activities of *scoping*, *analysis*, *development* and *deployment*. It emphasizes semantic scoping and traceability of decision making in modeling with specific deliverables of particular formats.

Scoping is to identify a part of the universe of discourse for modeling and development. *Stories* are used to convey business case, scenarios and their semantic scope to the knowledge or ontology engineers. Analysis is to create a knowledge constituent model to describe how the application semantics can be decomposed and how each constituent is elaborated in the description of business logic. Development is a process of creating ontologies to capture the meta knowledge: the semantic relationship underpinning the domain knowledge. It has three main tasks: *extraction*, *abstraction* and *organization* of lexons. Deployment is to specify commitments: a set of instantiated and constrained lexons with respect to specific applications.

## 3 An Ontology of Fraud Forensics About Nigerian Frauds

The Nigerian fraud is a type of advance fee frauds. The perpetrator seeks to rob the victim of financial resources by luring him into paying fees for fictitious administrative or financial operations in the promise of a considerable share of a fictitious capital or fortune. The potential victim is approached nowadays through unsolicited emails. One common bait is the fortune left behind by the dead without heirs. We shall illustrate our ontology modeling with this “fortune from the dead” case (FFD). It

involves a fictitious need to transfer large sums of capital into an overseas bank account. The author of emails typically claims to be a bank official or a relative of the dead and offers up to 30 percent of capital for the victim to assist the transfer. The forms of Nigerian frauds vary greatly. The underlying scheme and pattern, however, are stable and recurrent. In view of this fact, we propose to capture the underlying conceptions in terms of ontology and handle the surface variations on the basis of the conceptual model.

### 3.1 Knowledge Scoping and Analysis

The process follows the main stages of the AKEM methodology of scoping and analysing the application knowledge in the application domain.

#### 3.1.1 Scoping with Stories

The semantic scope under consideration at a given time of knowledge engineering is specified and documented by a story. It not only identifies the focus or boundary of attention, but also conveys the semantic context in which it stands [8]. It is the deliverable of the scoping with a structured presentation of information. The following figure shows parts of the story and its structure.

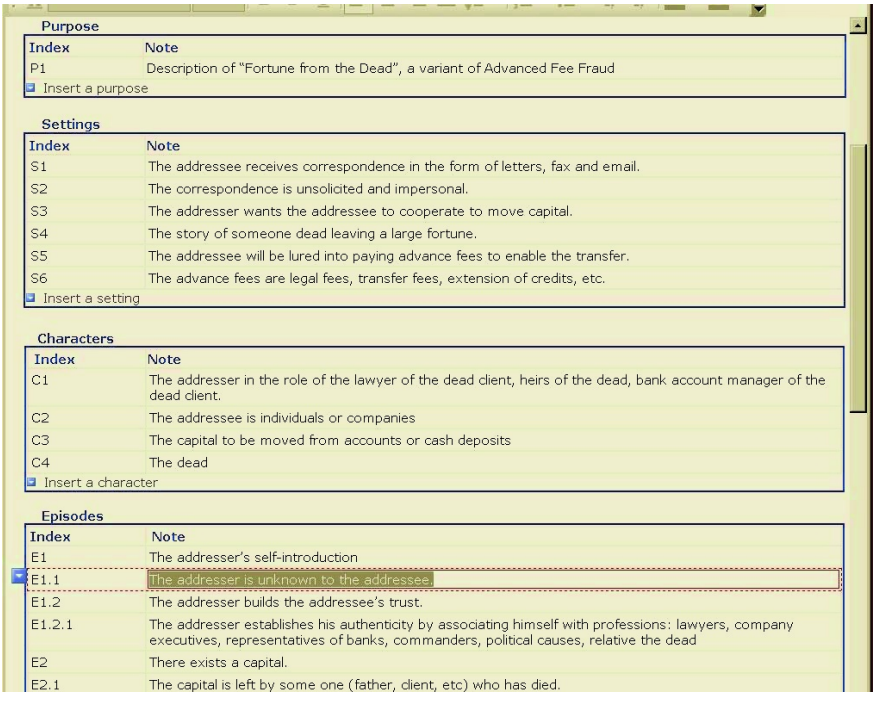



**Fig. 1.** Semantic scoping with the AKEM story editor.

The *Settings* of the story describe the background information whereas the *Characters* the actors or objects involved. The *Episodes* describe either sets of relationships

in hierarchy or a sequence of events or their composition. They are the starting points of the knowledge decomposition at the analysis stage of AKEM.

### 3.1.2 Knowledge Constituent Analysis

The analysis activity of AKEM is to create the knowledge constituent model within the semantic space defined by the story. It consists of knowledge decomposition and elaboration.

*Knowledge Decomposition.* It is a hierarchical structure of knowledge constituents, which is a four-layer detective model derived from Wigmore chart [7]. Fig. 2 is the knowledge decomposition of FFD. The top layer is the hypothesis of the existence of the fraud. The second layer consists of supporting postulates to the hypothesis. There are six basic ingredients of the FFD fraud scheme: medium, fraudster, victim, bait, offer and follow-up action. The third layer specifies evidences in the postulates and the fourth the facts that expresses or embodies the evidences.

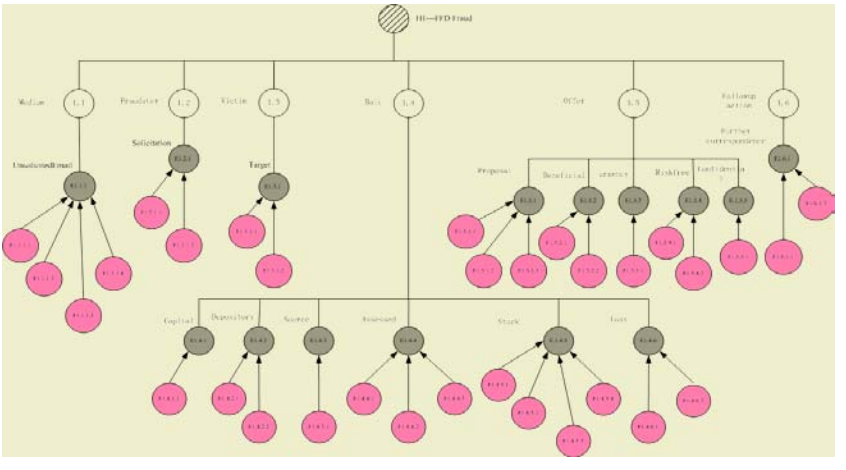


**Fig. 2.** Knowledge constituent model of FFD.

*Knowledge Elaboration.* The knowledge elaboration are the logical statements of about each constituent in a controlled language. They are a set of rules which represent the semantic connection among the knowledge constituents as well as description of the constituent. They also indicate the heuristics involved in the fraud investigation.

Fig. 3. describes the knowledge elaboration concerning Postulation Node 1.1, Evidence Node 1.1.1 and Fact Nodes 1.1.1.1, 1.1.1.2, 1.1.1.3, 1.1.1.4 and 1.1.1.5. It describes the unsolicited emails is used as medium for FFD.

## 3.2 Ontology Development

The ontology development seeks to define the concepts and relationships underlying the knowledge elaboration. Three tasks are performed to produce a set of lexons: extraction, abstraction and organization [10]. Extraction is to spot the key words and phrases of the elementary semantic conceptions (the left part of Fig. 4).

**IF** Fact (1.1.1.1) the addressee surprised by the email
**OR IF** Fact (1.1.1.2) the addressee is unknown addressor
**OR IF** Fact (1.1.1.3) the addressor introduce self
**OR IF** Fact (1.1.1.4) the addressor impersonal reference to addresser
**OR IF** Fact (1.1.1.5) the addressor describe the acquirement of addresses
**THEN** Evidence (1.1.1) the addressor send unsolicited email to addressee

**IF** Evidence (1.1.1) the addressor send unsolicited email to addressee
**THEN** Postulate (1.1) email is unsolicited as the medium of the fraud scheme

**Fig. 3.** An example of the knowledge elaboration.


**Fig. 4.** Extraction and abstraction of lexons in AKEM Ontology Editor.

The abstraction is a process of identifying the objects and relationships expressed the highlighted key words and phrases and formalizing them into lexons. The purpose is to model only those concepts and relationships explicitly verbalized in the deliverable of knowledge elaboration. The result of the abstraction is a set of lexons (the right part of Fig. 4). The organization is devoted to the introduction of conceptions that are presumed or implied in the story and knowledge elaboration, such as subtyping relationship. Table 1 shows some lexons produced during the ontology development phase.

**Table 1.** A subset of lexons concerning FFD.

| Context | $Term_1$ | $Role_1$ | $Term_2$ | $Role_2$ |
|---|---|---|---|---|
| FFD | Fraud | CharacterisedBy | FraudType | Characterise |
| FFD | FraudScheme | ConsistOf | Mediums | |
| FFD | FraudScheme | ConsistOf | Bait | |
| FFD | FraudScheme | ConsistOf | Offer | |
| FFD | FraudScheme | ConsistOf | FollowingupAction | |

# 4 Linguistic Modeling

The linguistic model is treated as deployment of the ontology of fraud forensics resulting from the previous activities of knowledge analysis and development. At the stage of deployment, the knowledge constituent analysis and ontology serve as development specification documents to organize and manage the linguistic engineering. The phased knowledge and language modelling encourages the formal identification of the application semantics about recurrent patterns of frauds and linguistic structures that express fraud evidences in texts, and, more importantly, the explicit representation of the mappings between two models. The layered modelling encapsulates the variation and changes in the linguistic expressions of evidences, which facilitates responsive model adaptation to keep up with the evolution of frauds.

## 4.1 Ontology Commitments for Recognizing FFD

The ontology model of FFD is produced without considering how and where the basic concepts and relationships are used or deployed. It focuses on the problem space rather than the solution space. The application of recognizing the “red flags” of the Nigerian frauds in emails, highlighting words and phrases expressing concepts of the fraud model.

From the deliverables from knowledge scoping and analysis, 15 red flags can be identified:

- Unsolicited email
- Targeted cooperator
- Solicitation
- The Dead
- Capital
  - Depository
  - Access
- Obstacle
  - Capital stuck
  - Capital to loose
- Proposal
  - Benefits
  - Minimal risks
  - Confidentiality
  - Urgency
- Follow-up
  - Further Correspondence

The relevant lexons are selected to create the conceptual model of the FFD. For example, the lexon, <*FFD, Addressor, Solicit, Addressee, SolicitedBy*>, shows that it is important to establish from the text or its pragmatic context that the intention of the email is to solicit. Fig. 5 is the graphic representation of some key lexons needed to produce linguistic model for red flag recognition. The conceptual model of fraud evidences in terms of relevant lexons specifies what to model in the language model.

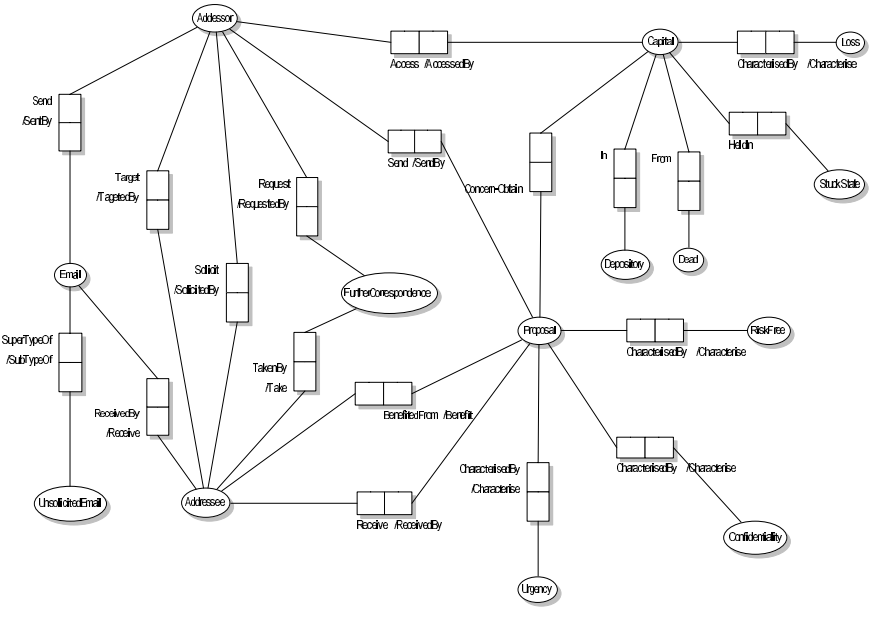



**Fig. 5.** Some key ontological terms and relationships in FFD.

## 4.2 Creating Natural Language Model

Linguistic modeling produces regular expressions of words and phrases as the linguistic evidences of frauds in emails. Since the fraud evidence is actually embodied in texts in natural language, the regular expressions are used to specify the lexical and syntactic pattern of the natural language expressions. In this study, we did not explore the use of any natural language processing/understanding engine. The popular regular expression engine was considered sufficient to explore the ontology-based natural language engineering for fraud detection at this preliminary stage. A previous similar approach of ontology-based information extraction was explored by Data Extraction Group at Brigham Young University [1].

The development process takes three steps:

- Define a semantic model based on the deliverables from the activities of knowledge analysis and ontology development)
- Describe syntactic and lexical structures that express the semantic model in the light of the representative samples
- Formalize the syntactic and lexical description in regular expressions

The linguistic model developed for the current study is based on 20 email samples of the Nigerian frauds. Table 2 shows three key intermediate results of linguistic modelling: ontological proposition to extract, linguistic tokens and the underlying linguistic structure in regular expression.

**Table 2.** Example deliverables in language modeling.

| Lexon commitments | Natural language structures | Regular expression |
|---|---|---|
| "addressor seek foreigner", "addressor seek overseas firm", "addressor seek assistant", "addressor seek partnership". | need\|seeking\|asking\|soliciting\|appear\|lookingfor\|search\|desire …<br><br>foreign partner \| person \| assistant \| help \| partnership \| overseas firm \| cooperation \|… | `(?:(?:look\|seek\|solicit\|ask\|appeal\|need\|desire\|search\|like\|request)(?:ing\|s){0,1}\s(?:\w*\s){0,6}(?:help\|assistance\|partner\|participant\|foreigner\|person\|account\|permission\|relationship\|partnership\|co{0,1}o{0,1}peration\|oversea\sfirm))` |

## 5 Detecting Fraud Red Flags

RegExTest [6] is used to extract red flags from suspicious emails against the ontology model. Fig. 6 shows how to RegExTest is used to extract the fraud evidence the suspicious email and to list and group extracted fraud evidences.

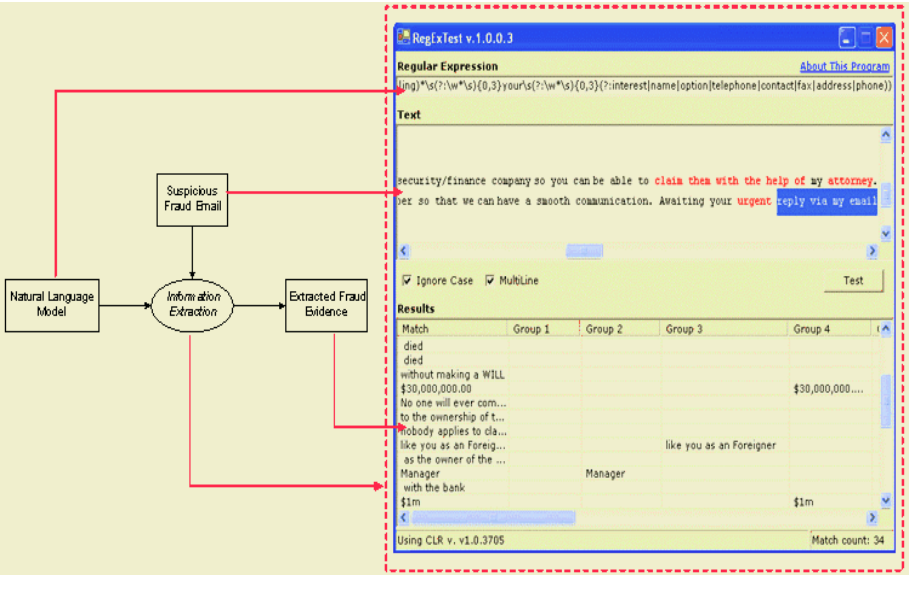



**Fig. 6.** Using RegExTest tool to detect fraud evidence of FFD.

### 5.1 Test and Evaluation

The 50 emails of FFD other than the 20 used for conceptual and linguistic modeling have been tested on the RegExTest. Table 3 shows the processing log of one email of FFD in terms of 15 *Categories of Evidence*. The *Linguistic Facts* are the linguistic tokens extracted rightly or wrongly or should have been extracted. They can be interpreted as instance one or more categories of evidence by the fraud model FFD.

**Table 3.** A Processing Log of An Email of FFD.

| Linguistic Facts | Categories of Evidence | | | | | | | | | | | | | | | Total |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | |
| Extracted | 5 | 2 | 2 | 6 | 6 | 2 | 11 | | 1 | 3 | | | 2 | | | 40 |
| Mistaken | 1 | | | | | | 1 | | | 2 | | | | | | 4 |
| Missed | | | 1 | | | 1 | | | | | | | | | 1 | 3 |

Table 4 shows a summary of the statistics of the 50 emails of FFD examined. The *Extracted* column is the total of linguistic facts extracted according to the fraud evidence and linguistic model. The *Ideal* column is the linguistic facts identified by the human investigator with respect to the fraud model. The average rate of mistaken recognition is 1.66. An average of 2.72 linguistic tokens are missed per email out of the fifty. The *Precision* counts the correct extraction in all the extraction whereas the *Recall* counts the correct recognition by the existing fraud evidence in one suspicious email. The average precision and recall of evidence recognition is 94% and 91%.

**Table 4.** Evaluation of 50 emails of FFD.

| Email | Extracted | Correct | Mistaken | Missed | Ideal | Precision | Recall |
|---|---|---|---|---|---|---|---|
| 1 | 40 | 36 | 4 | 3 | 39 | 90% | 92% |
| 2 | 48 | 47 | 1 | 4 | 51 | 98% | 92% |
| 3 | 44 | 42 | 2 | 2 | 44 | 95% | 95% |
| 4 | 50 | 49 | 1 | 1 | 50 | 98% | 98% |
| 5 | 53 | 52 | 1 | 0 | 52 | 98% | 100% |
| 6 | 36 | 35 | 1 | 3 | 38 | 97% | 92% |
| …… | | | | | | | |
| 46 | 19 | 19 | 0 | 2 | 21 | 100% | 90% |
| 47 | 26 | 24 | 2 | 1 | 25 | 92% | 96% |
| 48 | 24 | 24 | 0 | 2 | 26 | 100% | 92% |
| 49 | 39 | 34 | 5 | 0 | 34 | 87% | 100% |
| 50 | 27 | 26 | 1 | 1 | 27 | 96% | 96% |
| Average | 28.54 | 26.88 | 1.66 | 2.72 | 29.6 | 94% | 91% |

### 5.2 Tests on Cases of "Over-Invoicing"

Though the ontology model and linguistic model is produced on FFD, we use it to extract red flags from emails suspicious of other forms of Nigerian frauds to test the model applicability and robustness. The application ontology of frauds is intended to be generic over a family of similar applications and solutions.

The case of over-invoicing is different from the FFD in that the fictitious fortune comes from over-invoicing in business contracts. 50 suspicious emails with over-invoicing cases are processed using the same ontology and language models. Table 5. summarises the statistics of the results. The average rates of mistaken and missed recognition are 1.58 and 4.32. The precision and recall of evidence recognition is 92% and 80%. The precision almost the same as the results on the FFD cases. The recall is lower, due to the lacks in the over-invoicing specific details in conceptual and linguistic model. The overall result demonstrates that the application ontology of the Nigerian fraud forensics as knowledge specification covers sufficiently the problem space and serves as a basic layer of reusable knowledge representation.

**Table 5.** Evaluation of 50 emails of the Over-invoicing Case.

| Email | Extracted | Correct | Mistakes | Missed | Ideal | Precision | Recall |
|---|---|---|---|---|---|---|---|
| 1 | 21 | 20 | 1 | 4 | 24 | 95% | 83% |
| 2 | 16 | 16 | 0 | 5 | 21 | 100% | 76% |
| 3 | 17 | 13 | 4 | 4 | 17 | 76% | 76% |
| 4 | 39 | 39 | 0 | 1 | 40 | 100% | 98% |
| 5 | 15 | 14 | 1 | 4 | 18 | 93% | 78% |
| 6 | 23 | 21 | 2 | 4 | 25 | 91% | 84% |
| …… | | | | | | | |
| 45 | 16 | 14 | 2 | 4 | 18 | 88% | 78% |
| 46 | 11 | 10 | 1 | 7 | 17 | 91% | 59% |
| 47 | 28 | 25 | 3 | 2 | 27 | 89% | 93% |
| 48 | 27 | 25 | 2 | 3 | 28 | 93% | 89% |
| 49 | 27 | 21 | 6 | 2 | 23 | 78% | 91% |
| 50 | 24 | 22 | 2 | 4 | 26 | 92% | 85% |
| Average | 20.68 | 19.1 | 1.58 | 4.32 | 23.42 | 92% | 80% |

### 5.3 Tests on Normal Emails

The conceptual and language models are also tested on 50 non-fraudulent emails. Their subject of the emails varies from private to business or technical correspondences. The purpose of the test is to assess the differentiability between fraudulent and non-fraudulent emails by the fraud model and their corresponding linguistic model. Our assumption is that the bigger the difference is between the red flag recognition on the two sets of emails, the more performative of the model in recognizing emails suspicious of the Nigerian fraud will be.

Most linguistic tokens extracted from normal emails express the concepts of the unsolicited email, solicitation, capital and further correspondence. Expressions of the concepts relevant to the fraud model are fewer than 4 for every email in comparison with more than 10 expressions from suspicious ones.

Figure 7 shows the general trends or distribution of the three sets of tests. The percentage is the categories of evidence present out of the 15 categories in the fraud model. The linguistic tokens recognized include both the correct and mistaken recognition. The missed recognition is not considered. In other words, the distribution re-

flects the unedited evidence profile produced automatically of the suspicious and normal emails. It is however worth noting that each category of evidence carries equal weight in the percentage calculation. This is counter-intuitive, since some categories of red flags are more important than the others. Weighting on the red flags will be taken into account in the future research in order to capture experts' heuristics and intuitions.

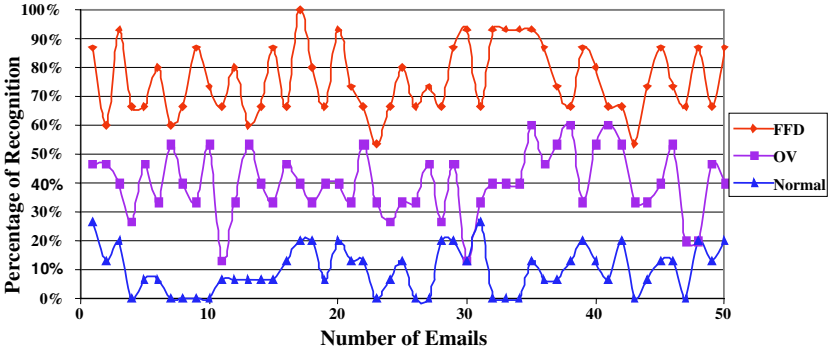



**Fig. 7.** Distribution of fraudulent and non-fraudulent emails.

## 6 Conclusion

This study focuses on the use of ontology for semantic text mining. It uses an ontology-based knowledge engineering approach to define the model of domain semantics and explores its use as software specification to guide linguistic modeling in regular expressions for information extraction. Though the solution of the automatic recognition of Nigerian frauds is linguistic processing, it is not approached as a task of linguistic engineering in the first place. Instead, the problem space is modeled as ontology-based application knowledge engineering. The decoupling of linguistic and conceptual modelling in development serves an fundamental requirement in machine assisted fraud detection. It is to capture the underlying recurrent essence of frauds while allowing for flexibility and adaptability to their creative and variable forms by distinguishing between models of problem and solution spaces and modeling knowledge in layers.

From the semantic modeling to linguistic rule specification, six major tasks are performed: problem definition, knowledge resources collection, knowledge scoping, analysis, application ontology development, and ontology deployment: natural language modeling. The whole process is purpose-driven and leads to a high precision rate.

The model of application semantics is not deployed on a natural language processing engine. Though its theoretical and potential issues are not explored, the experiment proves the pragmatic value of the dedicated and methodical domain modeling and the popular and easily available regular expression engine. A large and less structured domain can prove a serious challenge to the approach in the complexity of linguistic engineering with regular expressions. The use of easily customisable the natural language engine can be instrumental to manage the semantic and linguistic complexity.

The study also explores a methodology of development of the information extraction technology as fraud-alert expert system. The initial experiment results shows a good promise of such technology to assist monitoring and supervision of emails contents in proactive fraud detection and prevention.

## Acknowledgement

The authors wish to thank Robert Meersman for his advice and support to the research on ontology. This study is partially funded by the EU 5th framework program, IST 2001-38248, the project of FF POIROT (www.ffpoirot.org).

## References

1. Data Extraction Research Group at BYU. http://www.deg.byu.edu/
2. Deray, T.,Verheyden, P.: Towards a Semantic Integration of Medical Relational Databases by Using Ontologies: a Case Study. On the Move to Meaningful Internet Systems 2003: OTM 2003 Workshops, Lecture Notes in Computer Science, Vol. 2889/2003. Springer-Verlag, Heidelberg, (2003) 137-150.
3. Jarrar, M., Demey, J., Meersman, R.: On Reusing Conceptual Data Modeling for Ontology Engineering. Journal on Data Semantics, 1(1) (2003) 185 – 207
4. Kruchten, P.: The Rational Unified Process: An Introduction, Boston, Addison Wesley (2000).
5. Meersman, R.: Reusing Certain Database Design Principles, Methods and Techniques for Ontology Theory, Construction and Methodology. STARLab Technical Report 01 (2000)
6. RegExTest Project at: http://sourceforge.net/projects/regextest
7. Wigmore, J.H.: The Science of Judicial Proof as given by Logic, Psychology and General Experience. Boston, Little Brown (1937)
8. Zhao Gang: DOGMA-AKEM in FFpoirot. Draft report in WP6 of FFpoirot. (2003)
9. Zhao, G., Gao, Y., Meersman, R.: An Ontology-based Approach to Business Modeling. In: Proceedings of the International Conference of Knowledge Engineering and Decision Support (2004) 213 – 221
10. Zhao, G., Kingston, J., Kerremans, K., Coppens, F., Verlinden, R., Temmerman, R., Meersman, R.: Engineering an Ontology of Financial Securities Fraud, OTM 2004 Workshops, Lecture Notes in Computer Science, Vol. 3292. Springer-Verlag, Heidelberg, (2004) 605-620.

# Extended Tagging and Interpretation Tools for Mapping Requirements Texts to Conceptual (Predesign) Models

Günther Fliedl[1], Christian Kop[1], Heinrich C. Mayr[1], Martin Hölbling[1], Thomas Horn[1], Georg Weber[1], and Christian Winkler[2]

[1] Institute of Business Informatics and Application Systems, University of Klagenfurt, Austria
[2] Institute of Computational Linguistics, University of Klagenfurt, Austria

**Abstract.** This paper discusses an advanced tagging concept which supplies information that allows for a tool supported step by step mapping of natural language requirements specification to a conceptual (predesign) model. The focus lies on sentences containing conditions as used for describing alternatives in business process specifications. It is shown, how the tagging results are interpreted systematically such allowing for a stepwise model generation.

## 1 Introduction

Requirements texts describing dynamic aspects of a Universe of Discourse (UoD) mostly contain conditions, i.e., sentences of the type "if-then". These and other variants of sentence patterns decoding implicational relations often prove hard to be analyzed by the methods of deep parsing, because deep parsing rules are not flexible enough to deal with many types of syntactic variance, ambiguity and other sorts of linguistic complexity existing in this context, including phenomena like conjunction reduction. Requirements specifications for information systems mainly live from such "flexible" implicational sentence constructs describing all kinds of (business) processes. We, therefore, had to find a flexible solution for tackling such kind of sentence constructs in our NIBA project[1], which aims at mapping requirements texts into conceptual models.

Semantic tagging, e.g. labeling words by semantic classifiers ("sem-tags" like "tvag2 see [3]) seems to be a promising approach for extracting general patterns out from conditional sentence constructs which, in a subsequent step, can be interpreted and mapped to a conceptual model pattern.

Classical tagging approaches use standardized (POS) tag sets. This kind of standardized tagging (e.g. Treetagger, Brilltagger, Q-tag etc.[1], [10], [13]), however, seems to have the following three weaknesses:

- The tags provide merely lexical-categorial information,
- Ambiguity can be made explicit only in a restricted sense,
- Only limited chunking (normally chunking of simple NPs) is possible.

Furthermore, up to now POS-tagging mainly focused on the English language [13]. However, problems soon emerged when taggers were developed for "foreign" languages. With respect to German, big problems arise from its morphological "rich-

[1] The NIBA project is founded by the Klaus Tschira Stiftung Heidelberg. The aim of the project is natural language processing to support requirements engineering and conceptual modeling.

A. Montoyo et al. (Eds.): NLDB 2005, LNCS 3513, pp. 173–180, 2005.
© Springer-Verlag Berlin Heidelberg 2005

ness" and its (relatively) free word order. The complexity of words, the ambiguity of word endings and the big amount of serialization variants rarely allow for a straight forward interpretation of the input material.

To overcome these linguistic problems a tagging and interpretation approach has been elaborated [3] and implemented within the NIBA framework. In this paper we introduce, as a new key concept, a four-level interpretation of XML presented Tagger-Output. For the purpose of pragmatic interpretation we additionally assume that in the context of typical scenario texts even simple sentences may be linguistic encodings of if-then relations (the if-part can be empty). The tools used for tagging and interpretation as well as their underlying strategies are presented within this paper. In section 2 the underlying basic linguistic theory is outlined as well as some notions of the modeling language we use as interlingua between natural language and common conceptual modeling languages (like, e.g. UML [11]). Section 3 introduces the extended tagging concept and its implementation in NIBA-TAG. Section 4 discusses the interpretation strategies of the tagging outputs. In Section 5 we demonstrate how the tagging and interpretation components may be embedded into a stepwise mapping process. The paper ends with some comments of further research and development to be done.

## 2 Underlying Theories

### The Basic Linguistic Theory

As has been described in a series of precedent publications (see, e.g., [3]), NTMS (**N**atürlichkeits-**T**heoretische **M**orpho**S**yntax) proved to be a flexible fundament for the linguistic subtasks of the NIBA project. NTMS is a grammar model based on Generative Syntax in the Chomsky style [2]. Sentence related phenomena are represented by trees expressing constituency and dependency. These trees are projections of lexical base-categories. Each tree has just one lexical head and, accordingly, just one dominant heritage line. Given our framework, we tag the root elements with NTMS-defined morpho-syntactic and (domain-)semantic features for categories, types and subtypes of lexical base elements. Most of the used features have a pure semantic function (e.g. [countable], [animate]….etc).

Our focus of linguistic investigation lies on deep analysis of the morpho-syntactic relevance of verbal features like ["tvag2"] which decodes bivalency of the verbal head and agentivity of the involved subject. Features like these are related to specific configurations of Theta-roles, which can be presented as Predicate-Argument-Structures (PAS). Verbclass-specific PAS are core components of the typical NTMS lexicon entries for verbs (up to now we classified about 16.000 german verb entries). Lexicon elements bear a code valence and the argument position of the indicated theta-roles. Thematic role configurations can be derived from the sem-tags [4].

### The Conceptual Predesign Model (KCPM) – An Interlingua for Model Mapping

Within the context of mapping requirements specifications to conceptual models an interlingua approach, which applies a lean semantic model as an interface between natural languages texts and conventional conceptual models (like the ER-model,

UML), turned out be advantageous for both: the business owners (end users) being enabled to validate requirements models on a transparent and tolerable level of abstraction, and the mapping process itself the complexity of which may be reduced substantially by the use of an Interlingua. The model we propose is called KCPM (Klagenfurt Conceptual Predesign model) and described in detail in e.g. [7].

KCPM consists of a small set of modeling notions for static (structural) and dynamic UoD aspects. The latter also cover the needs of business process modeling. The main notions are **thing-type**, **cooperation-type**, **operation-type**, **pre**- and **post condition**. Thing-types represent important domain concepts. A thing-type is a generalization of the classical conceptual notions class and value type, so that, e.g., both *customer* and *customer name* are modeled as (different) thing-types. Typical things (instances of thing-types) are natural or juristic persons, material or immaterial objects, abstract notions. In textual requirements specifications they usually are referred to by noun phrases. Operation-types are used to model functional services that can be called via messages (service calls, [6]) between objects that are seen as systems. As such they may be perceived of as a generalization of the notions use-case, activity, action, method, service etc. Each operation-type is characterized by references to so-called thing-types which model the **actors** (acting and calling actors of the respective operation-type) and service parameters.

An acting actor is the thing-type which a given operation-type has been assigned to as a service. It is capable of carrying out instances of the operation-type. A caller is a thing-type that can initiate the execution of an operation-type instance by calling up the corresponding service of the respective acting actor. In the case of business process modeling a *customer* can initiate a "service" of an enterprise (e.g. *to answer to a letter of complaint*). In that case on the top level the *enterprise* is the acting actor (the thing type which is responsible for executing the operation type "*answer to a letter of complaint*"). The *letter of complaint* and *the answer* are the incoming and outgoing service parameters.

To sum up, UoD dynamics emerge from (acting) actors performing operations invoked by (calling) actors under certain circumstances (**pre-conditions**) and thus creating new circumstances (**post-conditions**). This is captured by the KCPM concept of **cooperation-type**, a term we adopted from object orientated business process modeling [8]. A particular cooperation in that sense is an elementary step of a business process to which one or more actors contribute by (concurrently) executing operations (services). Thus for our complaint letter example a cooperation type could have the following pre-condition, operation-types and post-conditions:

- pre-condition: *letter of complaint about a product failure comes in,*
- involved operation-types with contributing (acting) actors: *secretary checks customer status, clerk searches for amount of loss caused by the product failure,*
- post-conditions: *costumer is important and amount of loss > 10.000 Euro*.

## 3 Sophisticated Tagging

The tagging approach we propose includes [5], [3]

- the assignment of POS and semantically motivated subclass tags,
- morphological parsing with suffix identification, stemming, lemmatizing and composita splitting,

- chunking of phrases (different types of word grouping),
- relating phrases to syntactic and semantic “default”-functions.

Consequently, much effort had to be spent into the definition of context rules that act as disambiguation triggers during the tagging process. Based here-on, a tagging tool called NIBA-TAG has been developed and integrated into the NIBA Toolset.

NTMS based Part-of-speech tags divide words into morpho-syntactically and semantically motivated categories, based on how they can be combined to form sentences. For example, some types of auxiliaries can combine with participles and not with infinitives; articles can combine with nouns, but not verbs. Part-of-speech tags also supply information about the semantic content of a word. For example, nouns typically express "things," and prepositions express relationships between things. In addition, verbs encode different concepts of meaning. Sometimes they refer to features of an entity (in the case of ergative verbs), in other cases they describe an action carried out by the respective subject. The largest verb class is represented by bivalent relational agentive verbs.

A merely morphological classification of verb tags as is commonly practiced by most taggers proves unsuitable for conceptual orientation which is crucial in requirements engineering. Therefore, a sub-classification of verb tags from a semanto-logical point of view turns out to be indispensable. The example output below contains different sorts of parameter allocation and is derived from the following example sentence:

*“Ein einlangender Beschwerdebrief wird an die Beschwerdeabteilung weitergeleitet. (An incoming letter of complaint is forwarded to the complaints department)”*

```
<n3 position="0">
          <q0 number="1" numeral="Ein" position="0" type="numeral" id="0"
          lowerCase="ein">Ein
          </q0>
          <a0 position="1" id="1" lowerCase="einlangender">einlangender
          </a0>
          <n0 position="2" id="2" lowerCase="beschwerdebrief"
          lastcomponent="1">Beschwerdebrief
          </n0>
    </n3>
    <v0 referTo="werden" mode="pass" form="ind" position="1"
          verbclass-number="1" tense="present" person="3"
          lowerCase="wird" id="3" verbclass="AUX">wird
    </v0>
    ....
    <v0 referTo="weiterleiten" form="partizip" position="3" lastposition="1"
          verbclass-number="7" lastcomponent="1" tense="perfect" partikel="weiter"   lower-
Case="weitergeleitet" id="7" verbclass="tVag/2">weitergeleitet
    </v0>
```

In this XML-output the input words are related to wordclass-specific features like Verbclass= AUX, person= 3 or mode= pass(iv) for the auxiliary element “wird”; “verbclass= “tvag2”, partikel= weiter, tense= perfect for “weitergeleitet”, a german participle of the particle verb “weiterleiten”.

## 4 Interpretation and Tool Support

NIBA-TAG outputs are processed by the NIBA "Dynamics" interpreter which derives "operation-types", "conditions" and "cooperation-types". Fig.1 shows the main window of the interpreter with its four different views.

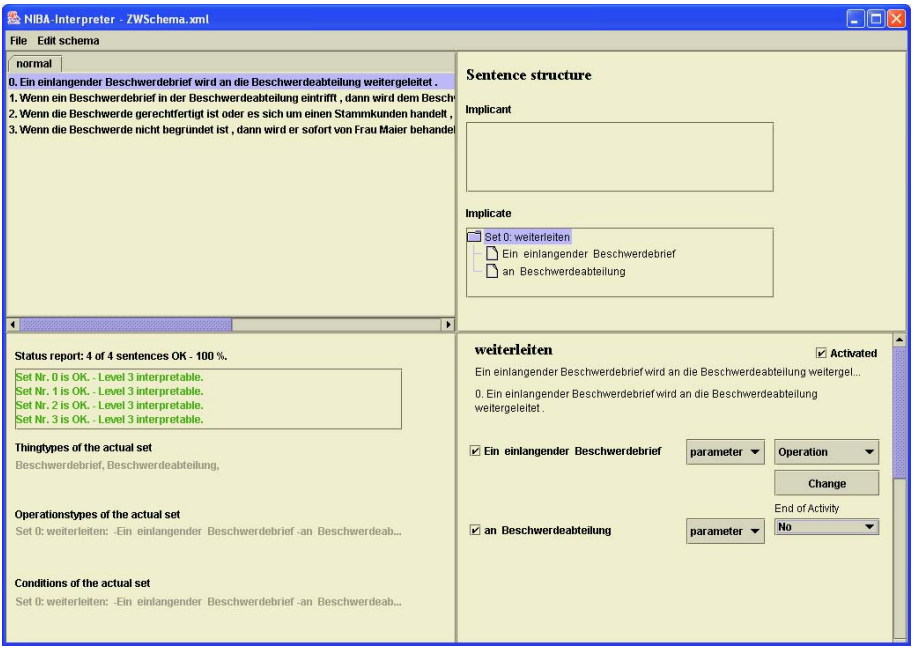



**Fig. 1.** Main window of the interpreter.

The left upper part of the window lists all the sentences of a text, the right hand upper part presents a structural template for each of these sentences based on the assumption that each sentence can be defined as consisting of a condition part and its implicational consequence. If the condition part is not stated explicitly then it is empty. In the condition part as well as in the implication part the verbs are collected together with those noun and prepositional phrases that could be candidates for verb arguments of the resp. verb. The identification of the verb arguments by the interpreter is based on the resp. NTMS verbclass.

The interpreter assigns syntactic functions and semantic roles to those word groups which are identified as syntactic phrases by NIBA-TAG (see section 3). This is a quite difficult task for German sentences because of the mentioned (morpho-) syntactic problems in German (free word order etc.; see section 3). The interpreter makes linguistically motivated default decisions. The assignments of the syntactic functions and semantic roles are used for the interpretation process. The results are shown in the right upper and lower part of the window. The interpretation process presupposes decisions about the interpretability of the tagged sentences. Currently a first and simple model of four levels (level 0 – level 3) was introduced:

**Level 0:** No interpretation of the sentence is possible at all. This means that the sentence is written in such a way that the interpreter cannot find any structure within the sentence which can be used for interpretation (see level 1).
**Level 1:** At least one of the two parts *implication* and/or *condition* is found.
**Level 2:** Verbs are found, but cannot be associated with arguments.
**Level 3:** Candidates for verb arguments are recognized for the verb (if there is only one in the sentence), or for at least one verb (if there are more of them in the sentence).

If a sentence can be interpreted on level 3 then it is possible to derive KCPM notions. This can be controlled in the right lower corner of the window. For each verb and its arguments in the implication section (upper right part) the end user receives a default interpretation. If the verb is an agentive verb then it is mapped to an operation-type. The noun which is the syntactic subject of the sentence is mapped to the acting actor. The nouns which could be the syntactic objects are mapped to parameters. If the verb is not an agentive verb then it is mapped to a condition. All the arguments of the verbs in the sentence are listed as candidates for involved thing-types. If there is more than one, the user has to select one of them to be the involved thing-type. For example, if the sentence "a person owns a car" is taken as a condition, then both "person" and "car" are candidates for the involved thing-type of that condition but only one of these nouns can be chosen as an involved thing-type (e.g. person). The rest of the sentence "owns a car" is then treated as the property of that thing-type.

All interpreter results are understood to be "default", i.e., the user can always overrule default decisions. E.g., the user can change the kind of each thing-type (parameter, calling actor, acting actor) if he thinks that the default assumption does not fit well.

Furthermore, currently, the tool distinguishes between sentences (sentence parts) that are useful for interpretation and sentences which are not ("fill-ins"). In fact, the tool assumes that every sentence should be interpreted. However, there is a check box "Activated" which is "on" by default. If the user disables this check box, then the interpretation result will not be transferred into in the final schema.

In some cases (where the sentence fits with some given implicational sentence patterns e.g. if/then constructs) the tool can also derive cooperation-types. The user then has the possibility to relate conditions and operation-types to logical operators (or, and, xor). Where possible, the tool gives hints about which of these operators should be chosen, otherwise, the user has to do this manually.

## 5 NIBA Workflow

*Tagging* and *Interpretation* as described in the previous sections are embedded as the first two steps (components) into a workflow supporting the derivation of KCPM entries and, in a subsequent step, supporting the mapping of these entries to a conceptual model (actually activity diagrams).

Basically, the NIBA Workflow consists of the following phases, and each of these phases is supported by at least one tool or a tool component.

1. Sentence analysis using the NTMS based NIBA-TAG (see section 3).
2. The NIBA Interpreter Tool which derives KCPM notions (thing-type, operation-type condition and cooperation-type) using the tagger output.

3. KCPM schema editor which is used for schema verification and modifications by the end users and their consultants. Currently two different representations of the schema are implemented. One tool works with a graphical, the other one with glossary like schema representation. The latter also manages the links between the schema elements (e.g. specific thing-types, cooperation-types, operation-types and conditions) and the corresponding sentences in the requirements documents.
4. The transformation of the pre-design schema into a conceptual model (here UML Activity Diagrams) is realized by a special component [12] of the graphical editor.

All these tools are integrated in the 'NIBA-Manager', allowing the user to draw up and manage projects in an easy way. Within the NIBA Manager, documents can be uploaded, the working area (universe of discourse) can be defined, and tools necessary for the workflow can be added and removed.

## 6 Conclusion

The approach presented here uses combined tagging and interpretation for processing requirements text. We call this a "step by step generation" of the predesign notions "thing-type", "operation-type", "condition" as well as "cooperation-type" based on linguistic patterns. The advantage of this method is a multilevel representation of the information which allows for user feedback on all stages of requirements text processing and a subsequent automatic predesign (KCPM) schema construction.

## Acknowledgment

The authors like to thank Alex Salbrechter, Jürgen Vöhringer, and Christian Irrasch for their substantial implementation work on the other tools of the NIBA Workflow (*NIBA Manager*, *NIBA Graphic Dynamic Editor* and *KCPM Web-Interface*).

## References

1. Brill, E.: A simple rule-based part of speech tagger In: Proceedings of the Third Conference on Applied Natural Language Processing, ACL, 1992.
2. Fliedl, G.: Natürlichkeitstheoretische Morphosyntax, Aspekte der Theorie und Implementierung, Habilitationsschrift, Gunter Narr Verlag, Tübingen, 1999.
3. Fliedl, G.; Kop, Ch.; Mayerthaler, W.; Mayr, H.C.; Winkler, Ch.; Weber, G.; Salbrechter, A.: Semantic Tagging and Chunk-Parsing in Dynamic Modeling. In: Meziane F.; Metais E.; (eds.) Proceedings of the 9th International Conference on Applications of Natural Language Processing and Information Systems, NLDB2004, Salford UK, Springer LNCS 3316 pp. 421 – 426.
4. Fillmore, Ch.; Petruck, J.M.; Ruppenhofer J.; Wright, A.: FrameNet in Action: the case of Attaching. International Journal of Lexicography, 2003.
5. Fliedl, G., Weber, G.: Niba-Tag - A Tool for Analyzing and Preparing German Texts. In: Zanasi, A.; Brebbia C.A.; Ebecken, N.F.F.; Melli, P. (Hrsg.): Data Mining 2002 Bologna: Wittpress September 2002 (Management Information Systems, Vol 6), pp. 331 – 337.
6. Hesse, W.; Mayr, H.C.: Highlights of the SAMMOA Framework for Objekct Oriented Application Modelling, In: Quirchmayr, G.; Schweighofer, E.; Bench-Capon, J.M. eds.: Proceedings of the 9th International Conference of Database and Expert Systems Applications(DEXA'98), Lecture Notes in Computer Science, Springer Verlag, 1998, pp.353 -373.

7. Kop C., Mayr H.C.: An Interlingua based Approach to Derive State Charts form Natural Language Requirements In: Hamza M.H. (Hrsg.): Proceedings of the 7th IASTED International Conference on Software Engineering and Applications, Acta Press, 2003, pp. 538 - 543.
8. Kaschek, R.; Kohl, C.; Mayr, H.C.: Cooperations – An Abstraction Concept Suitable for Business Process Reengineering. In Györkös, J.; Krisper, M.; Mayr, H.C. eds.: Conference Proceedings ReTIS'95, Re-Technologies for Information Systems, R. Oldenbourg Verlag, Wien, München, 1995, pp. 161 – 172.
9. Marcus, M.; Santorini, B. and Marcienkiewicz, M.: Building a large annotated corpus of English: the Penn Treebank., Computational Linguistics, 1993.
10. Schmid, H.: Probabilistic Part-of-Speech Tagging using Decision Trees.¨ http://www.ims.uni-stuttgart.de/ftp/pub/corpora/tree-tagger1.pdf, 1994.
11. Schach, Stephen R.: An introduction to object-oriented analysis and design with UML and the unified process, McGraw Hill, Boston, Mass., 2004.
12. Salbrechter, A.; Mayr H.C.; Kop C.: Mapping Pre-designed Business Process Models to UML In: Hamza M.H. (Hrsg.): Proc. of the 8th IASTED International Conference on Softeare Engineering and Applications Cambrigde USA: ACTA Press 2004 (400-405).
13. Tufis, Dan; Mason, Oliver.: Tagging Romanian Texts: a Case Study for QTAG, a Language Independent Probabilistic Tagger. Proceedings of the First International Conference on Language Resources & Evaluation (LREC), 1998, pp.589-596.

# Improving Question Answering Using Named Entity Recognition

Antonio Toral, Elisa Noguera, Fernando Llopis, and Rafael Muñoz

Grupo de investigación en Procesamiento del Lenguaje Natural y Sistemas de Información
Departamento de Lenguajes y Sistemas Informáticos
University of Alicante, Spain
{atoral,elisa,llopis,rafael}@dlsi.ua.es

**Abstract.** This paper studies the use of Named Entity Recognition (NER) for the Question Anwering (QA) task in Spanish texts. NER applied as a preprocessing step not only helps to detect the answer to the question but also decreases the amount of data to be considered by QA. Our proposal reduces a 26% the quantity of data and moreover increases a 9% the efficiency of the system.

## 1 Introduction

Question Anwering (QA) systems extract the answer to a user query from a set of documents. They receive as input the relevant documents or passages that Information Retrieval (IR) or Passage Retrieval (PR) systems respectively return.

IR systems [6] provide a sorted list of documents according to an input query. Thus, they play a very important role when looking for specific information in huge collections of text. Although there are different algorithms to carry out this task, there is a general consensus about some of the important variables to consider: number of times that each query term appears in the document, number of query terms that appear in the document and size of the document.

A drawback of these systems is that they do not take into account the proximity of query terms inside the document. An improvement of these systems consists of considering fragments of documents instead of the documents as a whole. This way, the proximity of query terms in the documents has an influence in order to estimate the relevance. This is the basis of Passage Retrieval (PR) systems [5].

Moreover, we may desire to get the answer to a query instead of the relevant fragments or documents. This is the goal of QA. While IR/PR systems usually use simple statistical techniques and thus are very efficient, QA are based on complex Natural Language Processing (NLP) algorithms and structures. That is why the later is not applied to the whole collection of documents but to the more relevant ones, that are ranked by using an IR/PR algorithm.

Following this tendency, our new approach consists of applying a Named Entity Recognition (NER) algorithm in order to reduce the number of QA input

A. Montoyo et al. (Eds.): NLDB 2005, LNCS 3513, pp. 181–191, 2005.
© Springer-Verlag Berlin Heidelberg 2005

passages. This involves preprocessing queries in order to find the entity type of the answer and applying NER to the PR output in order to discard the relevant passages in which no entities of this kind are found.

This paper is structured as follows. In the next section the theory behind PR, NER, and QA is outlined. The third section describes the architecture of our system. The fourth section presents the evaluation results. Finally, section five details conclusions and future research.

## 2 Background

### 2.1 Passage Retrieval as an Improvement of Information Retrieval

Passage Retrieval systems boost IR field by proposing a set of solutions to traditional IR systems common problems. The most important of these are:

- It is possible to locate the most relevant fragments of documents.
- It adds the concept of proximity to calculate the relevance.
- It avoids normalization of documents and the problems it entails.
- When applied as the input of QA systems, PR reduces the amount of text QA has to process.

A passage may be defined as a contiguous fragment of text in a document. It is the basic unit to Passage Retrieval processing. Its structure can vary according to the different approaches used to PR [2].

- Discourse model. It uses structural properties of the documents in order to define the passages, i.e. sentences, paragraphs.
- Semantic model. A document is divided into semantic pieces depending on the appearence of different topics within the document.
- Window-based model. The passages size is constant. Usually this size is measured in bytes or words.

To determine the relevance of a passage to a query, these systems measure the similarity between the query and the passage. There are different mathematical functions to compute this similarity, like cosine [11], pivoted cosine [12] and okapi [10].

### 2.2 Named Entity Recognition

NER is a subtask of Information Extraction whose aim is to detect and classify named entities in documents such as person, organization and localization names or dates, quantities, measures, etc. The term Named Entity was introduced in the 6th Message Understanding Conference (MUC6) [3]. In fact, the MUC conferences were the events that have contributed in a decisive way to the research of this area.

Mainly, there are two approaches to NER [1]: one based on hand-written rules and dictionaries and another involving supervised learning techniques. Although

the supervised learning approach is where more research effort focus nowadays, the rule-based model has some advantages. For example, this kind of model has better results for restricted domains, is capable of detecting complex entities that learning models have difficults with and can be adapted to deal with different types of entities by executing it with different configurations.

### 2.3 Question Answering

QA is the automatic task whose aim is to find specific answers, or at least small fragments of text, to unrestricted user queries in large collections of data.

Although being a very recent matter, Question Answering is one of the most important tasks of NLP and where numerous research efforts focus nowadays. This is mainly due to the importance of the conferences in which this subject is treated; Cross Language Evaluation Forum (CLEF) [4] and Text RETrieval Conference(TREC) [13]. One of the purposes of these conferences is to evaluate and compare the performance of QA systems.

To provide a specific answer, QA systems need to understand text at a minimum level. In order to accomplish this, it is necessary to carry out different techniques of natural language processing, like lexical, morphological, syntactic and semantic analysis.

Another fact about QA systems is that they should provide an answer within a short period of time if they were to be part of a computer-human real time interaction system. Because NLP techniques are expensive to compute and because of the time constraint, QA usually is not applied to a whole collection of documents but to the most relevant ones, which are usually provided by a IR or PR system. Therefore, our goal is to reduce the quantity of information that QA will consider.

## 3 System Architecture

Our system is made up of two modules: IR-n [7] and DRAMNERI. The first is a PR system while the later performs NER. A comprehensible diagram of the whole system is shown in figure 1.

### 3.1 IR-n

IR-n is a passage retrieval system whose main characteristics are:

- Passages are compounded of a fixed number of sentences.
- Passages overlap. In spite of increasing the processing time it has been proved that it obtains better results [8]
- It uses the cosine formula adapted to passages as the similarity measure:

$$Passage\_similarity = \sum_{t \in p \wedge q} W_{p,t} \cdot W_{q,t} \quad (1)$$

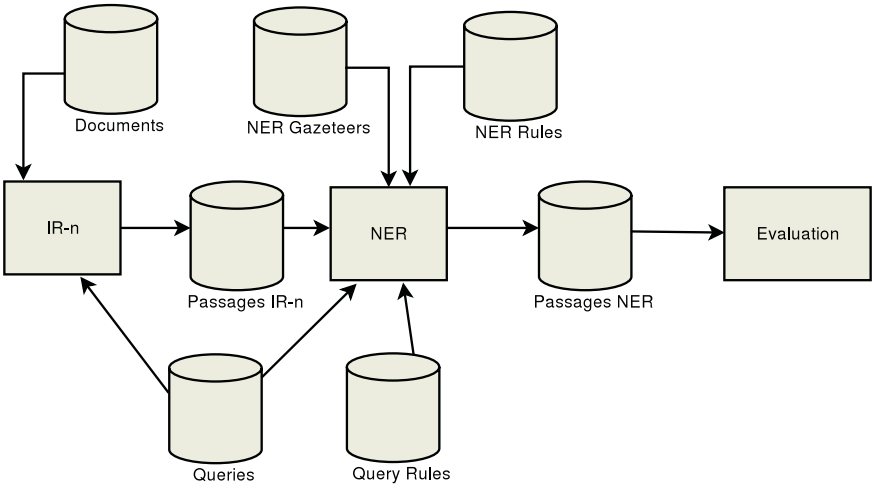



**Fig. 1.** System architecture.

Where
$W_{p,t} = log_e(f_{p,t} + 1)$,
$f_{p,t}$ is the number of appearances of term $t$ in passage $p$,
$W_{q,t} = log_e(f_{q,t} + 1) \cdot idf$,
$f_{q,t}$ represents the number of appearances of term $t$ in question $q$,
$idf = log_e(n/f_t + 1)$,
$n$ is the number of documents of the collection and
$f_t$ is the number of documents the term $t$ appears in.

Once the characteristics are noted, we outline the architecture of this passage retrieval system. IR-n is structured as a set of modules: indexing and passage retrieval. Their description follows.

**Indexing Module.** This module transforms the input documents in a set of structures to speed up the searching time. These structures contain information about the words which appear in the documents:

- Number of documents that contain each word
- Number of times each word appears in each document
- Sentence number in which each word appears in each document

Before building these structures we have to bear in mind that different forms of the same word (i.e plant and plants) should be stored as the same term. For this to be accomplished a stemmer is applied [9]. On the other hand, there are words that should not be stored because their frecuency of appearence is very high and they are not relevant. They are called stop-words.

**Passage Retrieval Module.** The aim of this module is to sort the passages according to its relevance to a given query. To accomplish this, the following subtasks are applied:

- Selection of passages. Determines the passages in which the terms of the query appear.
- Relevance computation of each passage. For each passage selected in the previous step, its relevance is calculated by applying the similarity measure (see equation 1).
- Visualization of results. The passages sorted by relevance are displayed. The format of this output may be adapted to specific needs.

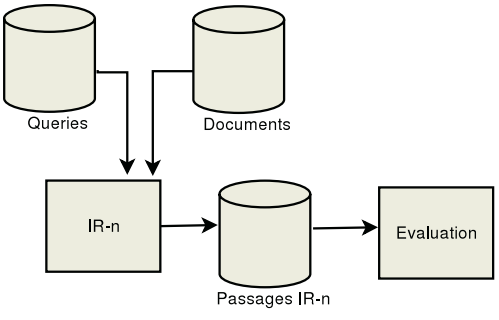



**Fig. 2.** IR-n architecture.

### 3.2 DRAMNERI

DRAMNERI (Dictionary, Rule-based and Multilingual Named Entity Recognition Implementation) is a system that classifies named entities. In this research named entity recognition is modified to detect only one entity class each time, according to the answer type.

This NER system is organized as a sequential set of modules with a high degree of flexibility, meaning that some modules may be used or not depending on the input. Also, most of the actions it performs, and the dictionaries and rules it uses are configurable by using parameter files. The main modules are briefly outlined in the following subsections.

**Answer Type Recognition.** To obtain the entity type that matches each answer we developed a module which, although not being part of the core NER system, has been adapted to this system in order to deal with this task. We did this recognition by using the interrogative particles (see table 1) and WordNet to get the hyponyms of given words, i.e. person.

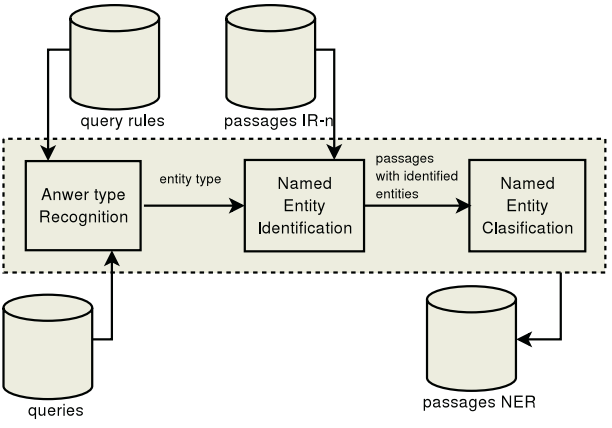



**Fig. 3.** DRAMNERI architecture.

**Table 1.** Rules to match answer entity types.

| Interrogative Particle | Matching entity |
|---|---|
| cuantos/cuantas (how many) | QUANTITY |
| cuando (when) | DATE |
| donde (where) | LOCALIZATION |
| quien (who) | PERSON |
| que (what/which) | PERSON, LOCALIZATION or ORGANIZATION |

As it can be seen in table 1, in some cases the interrogative particles is not enough to get the answer entity type. For these cases we use aswell WordNet hyponyms. As an example, if the interrogative particle is que (what), we use the rules using hyponyms that are shown in table 2.

**Table 2.** Rules to match answer entity types.

| Rule | Matching entity |
|---|---|
| que (what/which) + hyponym(person) | PERSON |
| que (what/which) + hyponym(localization) | LOCALIZATION |
| que(who) + hyponym(organization) | ORGANIZATION |

Some examples showing how these rules are applied to identify real questions follows:

- Quien (who) → ¿Quien es el presidente de Irlanda? (Who is the president of Ireland?) → PERSON
- Que (what) + hyponym(localization) → ¿Que país europeo es el mayor consumidor de alcohol? (Which european country consumes more alcohol?) → LOCALIZATION

**Named Entity Identification.** This task is applied on each sentence in the given text. Groups of capitalised words and/or words with numbers jointed by prepositions are detected and identified as generic entities. The maximum number of prepositions between two capitalised words and the list of prepositions to consider are configurable.

For example, if we have 'of' and 'the' in the preposition list and the maximum number of prepositions between capitalised words is 1, then the string "in the University of Alicante" would be identified as "in the <ENTITY> University of Alicante </ENTITY>" but "Lilly of the Valley" would be identified as "<ENTITY> Lilly </ENTITY> of the <ENTITY> Valley </ENTITY>" instead of "<ENTITY> Lilly of the Valley </ENTITY>" because 1 is the maximum number of prepositions between capitalised words.

**Named Entity Recognition.** The goal of this phase is to assign a category to each of the entities detected in the previous step. For this to be accomplished, rules, dictionaries and triggers are used. The boundaries of the identified entitities can be altered in this phase. This module is applied in two steps in a sequential manner:

*Classification using triggers.* For trigger driven classification length-configurable left and right context of the identified entity are considered. Within these contexts front triggers and back triggers dictionaries are applied respectively. If any happens to be found then the entity is classified with the category of the dictionary that the matching trigger belongs to.

For example, if we have the string "Mr. <ENTITY>Smith</ENTITY>" and mr. is a person trigger, then Smith is classified as a person entity. The output string would be "<ENTITY type=PERSON>Mr. Smith</ENTITY>".

*Classification using rules.* Dictionaries and rules are combined to perform entity classification. Rules follow the standard regular expression syntax and may contain elements that refer to dictionaries. Each rule is linked to an entity category. This way, if a rule matches a string of text then the category assigned is the one that is linked to the rule. An example follows:

rule: PER PREP PREP PER
entity: PER

This rule matches and entity that consists of a token which is in the Person dictionary (PER), followed by a token present in the preposition dictionary (PREP), etc. If a string of text matches then it is assigned the category PER. An example of string that would match is "Jorge de la Varga".

## 4 Evaluation

The aim of our evaluation is to determine whether or not the application of NER to the output of a PR system reduces the amount of data and if increases the

performance of the output data. In this section we present the data collection used. Then, we check if the entity type detected as the corresponding to the answer is the correct one. Later on, the evaluation measure is outlined and finally, the evaluation results of our system are presented.

### 4.1 Data Collection

We have used the EFE1994 document collection which was used at QA CLEF 2004. This is made up of 215,738 documents in Spanish. We have used also the collection of queries that was used at the QA track (Spanish) in CLEF-2004. It has 200 queries.

### 4.2 Anwering Entity Type Detection

Regarding our answer type recognition, we classified correctly 150 out of 200 queries. From the remaining 50 queries, 46 were not classified, and 4 were classified into an incorrect entity type. Therefore the recall is 75% and the precission 97.4%. For the 46 non classified NER just returns all the input passages.

### 4.3 Evaluation Measure

We use the Mean Reciprocal Rank (MRR) [7] to evaluate our system. The value assigned to each question is the inverse value of the first passage in which the answer is found or zero if the answer is not found. The final value is the average of the values for all the questions. This is obtained with the following formula:

$$MRR = (\sum_{i=1}^{Q} 1/far(i))/Q \tag{2}$$

Where

$Q$ is the number of queries,

$far(i)$ refers to the position of the first passage in which the correct answer is found to the query $i$,

$1/far(i)$ will be zero if the answer is not found in any passage.

### 4.4 PR vs. PR-NER

We have carried out an experiment consisting of comparing the results of PR (see figure 2) and PR when applying NER to its output (PR-NER, see figure 1) in order to test the performance of our system. This means that on one hand we evaluate the system with the passages that PR returns. On the other hand we evaluate only the passages that contain at least an entity of the same type of the answer.

To do this we applied PR to the data collection. Once we had the most relevant passages we used them as the input of our NER algorithm. NER returns the

passages that contain at least an entity of the answer type. Finally we compared the output of the NER system with the output of the PR.

We have tested the system with different passage size configurations. The aim of doing this is to find the model that optimizes MRR and where NER produces a maximum reduction of output data. The results of testing the system with different passage sizes are presented in table 3.

**Table 3.** Results using different passage sizes.

| Sentences per Passage | MRR PR | MRR PR-NER | output improvement | output reduction |
|---|---|---|---|---|
| 2 | 0.15 | 0.17 | 13% | 42% |
| 4 | 0.19 | 0.21 | 11% | 32% |
| 6 | 0.21 | 0.23 | 10% | 28% |
| 8 | 0.22 | 0.24 | 9% | 26% |
| 10 | 0.22 | 0.24 | 4% | 24% |
| 12 | 0.22 | 0.24 | 4% | 23% |
| 14 | 0.23 | 0.24 | 4% | 22% |
| 16 | 0.23 | 0.24 | 4% | 21% |
| 18 | 0.23 | 0.24 | 4% | 21% |
| 20 | 0.23 | 0.24 | 4% | 21% |
| 30 | 0.24 | 0.25 | 4% | 21% |
| 40 | 0.24 | 0.25 | 4% | 20% |

From the results obtained, we found that it is not worth to use a passage size greater than 8 because MRR growth is slowed down, the output data reduction achieved by NER decreases while the passage size is greater. For size 8, we obtained a significant reduction of the number of passages (26%) and a 9% increment of the MRR.

Aside from this, we provide more in-depth results for the 8 sentences per passage configuration. In table 4, MRR and output reduction are presented for each entity type. It should be pointed out that the reduction of data is significant for some kinds of entities, being Percentage (80%), Place_Capital (45%), Place_City (38%), Person (24%) the most important. Besides, there is a increase of MRR for some entities such as Place_Country, Person, Quantity among others.

## 5 Conclusions and Future Work

NER plays an important role in QA systems that work on the output of a PR module for it has proved to reduce substantially the amount of output data (26% for the optimal configuration) and even though, it improves PR output results (MRR) until a 13%. Moreover, DRAMNERI processing time is very low compared to IR-n module. If we consider 100% the total processing time then, on average, IR-n would take 94.8% and DRAMNERI the remaining 5.2%.

**Table 4.** In-depth results when using 8 as passage size.

| Entity class | MRR PR | MRR PR-NER | output reduction |
|---|---|---|---|
| Definition | 0.03 | 0.03 | 0% |
| Abbreviation | 0.2 | 0.2 | 0% |
| Event | 0.32 | 0.3 | 3% |
| Group | 0.17 | 0.17 | 4% |
| Place | 0.37 | 0.37 | 11% |
| Place_Capital | 0.05 | 0.05 | 45% |
| Place_City | 0.15 | 0.15 | 38% |
| Place_Country | 0.23 | 0.25 | 14% |
| Object | 0.57 | 0.57 | 0% |
| Person | 0.28 | 0.29 | 24% |
| Quantity | 0.21 | 0.23 | 21% |
| Economic | 0 | 0 | 0% |
| Age | 0.5 | 0.5 | 10% |
| Measure | 0 | 0 | 44% |
| Period | 1 | 1 | 0% |
| Percentage | 0 | 0 | 80% |
| Year | 0.23 | 0.24 | 36% |
| Date | 0.08 | 0.08 | 27% |
| Month | 0 | 0 | 10% |
| Non found | 0 | 0 | 0% |

It is important to mention that in most of the cases one of the returned entities is the answer to the question. This could take to an interesting line of future research, because it can lead to a drastic reduction in QA processing time.

We consider two other possible lines of future research. Firstly, to embed Named Entity Classification into the indexed PR data to increase the system speed. Secondly, to apply an hybrid NER, in a way that the rule based one does the task as in this article and we use the supervised learning to try to recognise non-classified entities.

## Acknowledgements

This research has been partially funded by the Spanish Government under project CICyT number TIC-2003-7180 and under project PROFIT number FIT-340100-2004-14 and by the Valencia Government under project number GV04B-268.

## References

1. A. Borthwick. *A Maximum Entropy Approach to Named Entity Recognition.* PhD thesis, New York University, September 1999.
2. J.P. Callan. Passage-level evidence in document retrieval. In *Proceedings of the 17th annual conference on Research and Development in Information Retrieval, London, UK*, pages 302–310. Springer Verlag, 1994.

3. N. Chinchor. Overview of muc-7. In *Proceedings of the Seventh Message Understanding Conference (MUC-7)*, 1998.
4. CLEF. Workshop of cross-language evaluation forum (clef 2003). In *Workshop of Cross-Language Evaluation Forum (CLEF 2003)*, Lecture notes in Computer Science, Trondheim, Nordway, 2003. Springer-Verlag.
5. M. Kaskziel and J. Zobel. Passage retrieval revisited. In *Proceedings of the 20th annual International ACM Philadelphia SIGIR*, pages 178–185, 1997.
6. F. W. Lancaster. *Information Retrieval Systems: Characteristics, Testing and Evaluation*. John Wiley and Sons, New York, 1979.
7. F. Llopis. *IR-n: Un Sistema de Recuperación de Información Basado en Pasajes*. PhD thesis, University of Alicante, 2003.
8. F. Llopis, J. L. Vicedo, and A. Ferrández. Passage selection to improve question answering. In *Proceedings of the Workshop on Multilingual Summarization and Question Answering, Taipei, Taiwan*, pages 11–16. Colling, 2002.
9. M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–130, 1980.
10. S. Roberston, S. Walker, and M. Beaulieu. Okapi at trec-7. In *Seventh Text RETrieval Conference, volume 500-242*, pages 253–264. National Institute of Standard and Technology. Gaithersburg, USA, 1998.
11. G. Salton. Automatic text processing: The transformation, analysis, and retrieval of information by computer. 1989.
12. A. Singhal, C. Buckley, and M. Mitra. Pivoted document length normalization. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Developement in Information Retrieval, Experimental Studies*, pages 21–29, 1996.
13. TREC-10. Tenth Text REtrieval Conference. In *Tenth Text REtrieval Conference*, volume 500-250 of *NIST Special Publication*, Gaithersburg, USA, nov 2002. National Institute of Standards and Technology.

# Using Semantic Roles in Information Retrieval Systems

Paloma Moreda, Borja Navarro, and Manuel Palomar

Grupo de investigación del Procesamiento del Lenguaje y Sistemas de Información
Departamento de Lenguajes y Sistemas Informáticos. Universidad de Alicante
Alicante, Spain
{moreda,borja,mpalomar}@dlsi.ua.es

**Abstract.** It is well known that Information Retrieval Systems based entirely on syntactic contents have serious limitations. In order to achieve high precision Information Retrieval Systems the incorporation of Natural Language Processing techniques that provide semantic information is needed. For this reason, in this paper a method to determine the semantic role for the constituents of a sentence is presented. The goal of this is to integrate this method in an Information Retrieval System.

## 1 Introduction

It is well known that Information Retrieval (IR) Systems based entirely on syntactic contents have serious limitations. One of the challenges of these applications is to develop high quality or high precision systems. In order to do this, it is necessary to involve Natural Language Processing (NLP) techniques in this kind of systems. These techniques provide semantic information to IR systems. Among the different NLP techniques which would improve IR systems, Semantic Role Labelling (SRL) is found . In this paper an extension of a IR system making use of a Semantic Role Labelling method is presented. Such method improves retrieval performance by reducing the number of non-relevant documents retrieved. This research is integrated in the project R2D2[1].

A semantic role is the relationship between a syntactic constituent and a predicate. For instance, in the next sentence

(E0) The executives gave the chefs a standing ovation

*The executives* has the Agent role, *the chefs* the Recipient role and *a standing ovation* the Theme role.

To achieve high precision IR systems, recognizing and labelling semantic arguments is a key task for answering "Who", "When", "What", "Where", "Why",

[1] This paper has been supported by the Spanish Government under project "R2D2: Recuperación de Respuestas de Documentos Digitalizados" (TIC2003-07158-C04-01). Besides, it has been partially funded by the Valencia Government under project number GV04B-276

A. Montoyo et al. (Eds.): NLDB 2005, LNCS 3513, pp. 192–202, 2005.
© Springer-Verlag Berlin Heidelberg 2005

etc. For instance, the following questions could be answered with the sentence (E0). The Agent role answers the question (E3) and the Theme role answers the question (E4).

(E1) Who gave the chefs a standing ovation?

(E2) What did the executives give the chefs?

These examples show the importance of semantic roles in applications such as Information Retrieval.

Currently, several works have tried using Semantic Role Labelling in IR systems, unsuccessfully. Mainly, it is due to two reasons:

1. The lower precision achieved in these tasks.
2. The lower portability of these methods.

It is easy to find methods of Semantic Role Labelling that work with high precision for a specific task or specific domain. Nevertheless, this precision drops when the domain or the task are changed. For these reasons, this paper is about the problem of Semantic Role Labelling integrated with a IR system.

The remaining paper is organized as follows: section 2 gives an idea about the state-of-art in IR systems using semantic information. Afterwards, the Semantic Role Labelling method is presented in section 3. Then, how this method improves the performance of a IR system is presented in section 4. Finally, 5 concludes.

## 2 Using Semantic Information in IR: Background

In several IR systems the meaning of documents resides solely in the words that are contained within them. So, these systems, based on mathematical models such as the Boolean model, the vector-space model, the probabilistic model and their variants [2], represent the meanings of documents and queries as bags of words. Even though they are well established, from the user's perspective, it is difficult to use these IR systems. Users frequently have problems expressing their information needs and translating those needs into queries.

For instances [21], consider the sentence *Harry loves Sally*. If it is considered as a query in a keyword matching system, the system would look for documents containing the terms *Harry*, *Sally* and *love*, and would not be able to distinguish among the following sentences (E3), (E4), (E5), (E6) and (E7).

(E3) Harry loves Sally

(E4) Sally loves Harry, but Harry hates Sally

(E5) Harry's best friend loves Sally's best friend

(E6) Harry and Sally loves pizza

(E7) Harry's love for Sally is beyond doubt

Several methods have been proposed to help users to choose searching terms and articulate queries making use of semantic information. Most of them work for a specific domain and use domain specific thesaurus. For instance, some systems use concepts[2]. So, the system presented in [19] first assigns a syntactic analysis to input from either a query or document about medical domain. The heart of the approach is a mapping of the phrases to concepts in UMLS domain model [1]. Then, the semantic interpretation specifies the relationship in a semantic case role form, which it is obtained between the concepts and the input phrases.

In [6] y [7] a method implemented on Digital Library using a hierarchical perspective based on a concept system is presented. The EDR [8] electronic dictionary was used as a thesaurus. The meanings of words extracted from queries and text are represented by concepts and are used for retrieval. The "concept of a word" indicates the concept which represents the shared synonyms for the meaning of a word.

Complex nominal sequences must undergo a specific semantic treatment in order to increase the performance of IR systems [5]. This work defines three objectives using semantic information on English compounds: determination of the conditions under which the concept expressed by a compound is presented in a text, recognition of equivalent reformulations of the compounds and a weighting of the words of the compounds proportional to their importance. An extension of this work is proposed in [4] adding an objective of disambiguation of polysemous words. The work shows, by using concrete examples from an experimentation conducted on a French system of telematic services, how a rich semantic model for binomial sequences can be used in order to increase both the recall and precision rates of an IR system. This system, named CNET, is composed of three modules: the linguistic analyzer, which generates a structured representation of a text in which each word is replaced by the list of its meanings; the indexing module, which generates the list of the indexes, and its weights according to their frequencies in the text, that represent the contents of a text; and the matching module, which valuates the relevance of a text for a given question.

On the other hand, several methods have investigated IR systems making use of relationships for specific domains or specific tasks. For instance, the use of relation matching in IR is discussed in [21]. Terms and relationships between terms expressed in the query are matching with terms and relationships found in the documents. In this method, non-domain specific knowledge was used. Nevertheless, the cause-effect relation was the only one studied.

The work of [23] is based on an algorithmic approach of concept discovery and association. Concepts are discovered using an algorithm based on an automated thesaurus generation process. Subsequently, similarities among terms are computed using the cosine measure, and the relationships among terms are established using a method known as *max-min* distance clustering.

NLP techniques with the structured domain knowledge provided by the UMLS, were applied to texts concerning to the coronary arteries in order to ex-

[2] "Concept" and "term" words are used according to the terminology used by respective authors

tract arterial branching relationships from cardiac catheterization reports [20]. First, the coronary artery terminology occurring in the sentence is identified. Next, the processing constructs a complete branching predication where a correspondence between a syntactic entity and a semantic predicate is established.

Besides the semantic relations are explored and evaluated in cross-language IR in the medical domain making use UMLS as the primary semantic resource and a corpus of English and German medical abstracts, in [22]. A method for selecting relevant relations from those proposed by UMLS and a method for extracting new instances of relations based on statistical and NLP techniques are described. First the specialist lexicon provides lexical information. Second, the metathesaurus is the core vocabulary component used for assigning a identifier for each term. Third, the semantic network provides a grouping of concepts according to their meaning into a semantic type and specifies potential relations between those semantic types.

Other researchers have studied general methods for extracting semantic relations for IR. In response, Lu [12] investigated the use of case relation matching using a small test database of abstracts. Using a tree-matching method for matching relations, he obtained worse results than from vector-based keyword matching. The tree-matching method used is probably not optimal for IR and the results may not reflect the potential of relation matching [21].

Liu [9] tried to match individual concepts together with the semantic role that the concept has in the sentence. Instead of trying to find matches for *term1-relation-term2*, his system sought to find matches for *term1-relation* and *relation-term2* separately. Liu used case roles and the vector-space retrieval model, and was able to obtain positive results only for long queries (abstracts that are use as queries).

The DR-LINK project attempted to use general methods for extracting semantic relations for IR. Non-domain specific resources were used. However, preliminary results found few relation matches between queries and documents.

## 3 The SemRol Method

In this section, the Semantic Role method, named SemRol, is presented.

The problem of the Semantic Role Labelling is not trivial. In order to identify the semantic role of the arguments of a verb, two phases have to be solved, previously. Firstly, the sense of the verb is disambiguated. Secondly, the argument boundaries of the disambiguated verb are identified.

First, the sense of the verb has to be obtained. Why is it necessary to disambiguate the verb? Following, an example shows the reason for doing so.

(E8) John gives out lots of candy on Halloween to the kids on his block

(E9) The radiator gives off a lot of heat

Depending on the sense of the verb a different set of roles must be considered. For instance, Figure 1 shows three senses of verb *give* (give.01, give.04,

and give.06)) and the set of roles of each sense. So, sentence (E0) matches with sense give.01. Therefore, roles *giver*, *thing given* and *entity given to* are considered. Nevertheless, sentence (E1) matches with sense give.06 and sentence (E2) matches with sense give.04. Then, the sets of roles are (*distributor*, *thing distributed*, *distributed*) and (*emitter*, *thing emitted*), respectively. In sentence (E1), *John* has the distributor role, *lots of candy* the thing distributed role, *the kids on his block* the distributed role and *on Halloween* the temporal role. In sentence (E2), *the radiator* has the emitter role and *a lot of heat* the thing emitted role. These examples show the relevance of WSD in the process of assignment of semantic roles.

```
<roleset id="give.01" name="transfer"> <roles>
  <role n="0"  descr="giver"  vntheta="Agent"/>
  <role n="1"  descr="thing given"  vntheta="Theme"/>
  <role n="2"  descr="entity given  vntheta="Recipient""/>
</roles>

<roleset id="give.04" name="emit"> <roles>
  <role n="0"  descr="emitter"/>
  <role n="1"  descr="thing emitted"/>
</roles>

<roleset id="give.06" name="transfer"> <roles>
  <role n="0"  descr="distributor"/>
  <role n="1"  descr="thing distributed"/>
  <role n="2"  descr="distributed"/>
</roles>
```

**Fig. 1.** Some senses and roles of the frame *give* in PropBank [17].

In the second phase, the argument boundaries are determined. For instance, in the sentence (E0), the argument boundaries recognized are

[The executives] gave [the chefs] [a standing ovation]

Once these two phases are applied, the assignment of semantic roles can be carried out.

So, our method, named SemRol, presented in this section consists of three phases:

1. Verb Sense Disambiguation phase (VSD)
2. Argument Boundaries Disambiguation phase (ABD)
3. Semantic Role Disambiguation phase (SRD)

These phases are related since the output of VSD phase is the input of ABD phase, and the output of ABD phase is the input of SRD phase. First, the process to obtain the semantic role needs the sense of the target verb. After that, several heuristics are applied in order to obtain the argument boundaries of the sentence. And finally, the semantic roles that fill these arguments are obtained. So, the success of the method depends on the success of the three phases.

Both, Verb Sense Disambiguation phase and Semantic Role Disambiguation phase are based on conditional Maximum Entropy (ME) Probability Models

**Table 1.** Results of the SRD phase.

| | Precision | Recall | $F_{\beta=1}$ | | Precision | Recall | $F_{\beta=1}$ |
|---|---|---|---|---|---|---|---|
| A0[a] | 92.17% | 90.50% | 91.33 | AM-MNR | 99.70% | 97.90% | 98.79 |
| A1 | 83.17% | 96.31% | 89.26 | AM-MOD | 100.00% | 100.00% | 100.00 |
| A2 | 98.14% | 88.26% | 92.94 | AM-NEG | 100.00% | 98.47% | 99.23 |
| A3 | 99.08% | 72.48% | 83.72 | AM-PNC | 100.00% | 99.00% | 99.50 |
| A4 | 92.86% | 35.37% | 51.23 | AM-PRD | 100.00% | 100.00% | 100.00 |
| A5 | 100.00% | 50.00% | 66.67 | AM-PRP | 0.00% | 0.00% | 0.00 |
| AM-ADV | 99.71% | 98.58% | 99.14 | AM-REC | 0.00% | 0.00% | 0.00 |
| AM-CAU | 98.11% | 98.11% | 98.110 | AM-TMP | 99.04% | 94.99% | 96.99 |
| AM-DIR | 96.30% | 86.67% | 91.23 | R-A0 | 0.00% | 0.00% | 0.00 |
| AM-DIS | 97.50% | 76.47% | 85.71 | R-A1 | 0.00% | 0.00% | 0.00 |
| AM-EXT | 96.00% | 48.98% | 64.86 | R-A2 | 0.00% | 0.00% | 0.00 |
| AM-LOC | 100.00% | 98.70% | 99.34 | R-AM-LOC | 100.00% | 75.00% | 85.71 |
| R-AM-TMP | 85.71% | 100.00% | 92.31 | V | 97.44% | 97.44% | 97.44 |
| | | | | **all** | **92.46%** | **92.38%** | **92.42** |
| | | | | **all–{V}** | **90.53%** | **90.41%** | **90.47** |

[a] The semantic roles considered in PropBank are the following [3]: Numbered arguments (A0-A5, AA):, arguments defining verb-specific roles; adjuncts (AM-), general arguments that any verb may take optionally, for instance, AM-LOC is location or AM-CAU: cause; references (R-), arguments representing arguments realized in other parts of the sentence; and verbs (V), participant realizing the verb of the proposition.

[18]. It has been implemented using a supervised learning method that consists of building classifiers using a tagged corpus [15]. Argument Boundaries Disambiguation and Semantic Role Disambiguation phases take care of recognition and labelling of arguments, respectively. VSD module means a new phase in the task. It disambiguates the sense of the target verbs. So, the task turns more straightforward because semantic roles are assigned to sense level. A more detalied approximation of this method is presented in [14].

Results about SRD phase are shown in table 1. In order to evaluate it, right senses of the verbs and right argument boundaries have been presumed. The results have been obtained using the PropBank corpus [17]. There are 26 different kinds of roles in this corpus. One of theme, the *V* role, refers verbs. In this case, the precision is about 97%. In most of cases, the precision is over 83%, being over 96% in sixteen of them (only four cases are below 96%) and 100% in seven of them.

## 4 IR System Extended with SemRol

The goal of this paper is to integrate the method presented in the previous section (section 3) in an IR system. In this particular case, in the IR-n system[11], [10].

The architecture of an IR system extended with the SemRol method is shown in the figure 2. The architecture presented consist of four modules: IR system, selection module, annotation module and module of heuristics.

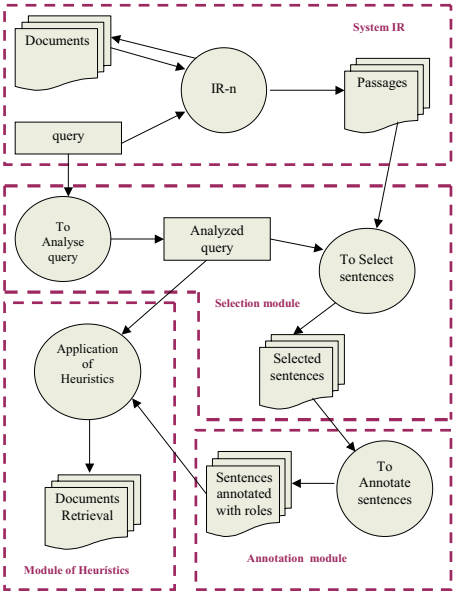



**Fig. 2.** The architecture of an IR system extended with the SemRol method.

When a query is done, the IR system IR-n retrieves a set of passages. It is supposed that these passages contain the answer of the query. Then the verbs of the sentences of these passages are compared with the verb of the query, and a list of verbs related with it, in order to select only the sentences containing a verb of this list. Next, the selected sentences are annotated with semantic roles making use of SemRol method. Finally, a set of heuristics are applied. These heuristics establish a relation between queries and semantic roles. So, only the sentences contained the right semantic roles are selected and the number of passages retrieved is reduced.

### 4.1 IR System IR-n

Passage Retrieval is an alternative to traditional document-oriented Information Retrieval. IR-n system is a passage retrieval system. These systems use contiguous text fragments (or passages), instead of full documents, as basic unit of information. So, IR-n system uses the sentences as atoms with the aim to define the passages. Thus each passage is composed by a specific number of sentences. This number depends in a great measure of the collection used. For this reason, the system requires a training phase to improve its results. IR-n system uses overlapping passages in order to avoid that some documents can be considered not relevant if words of the question appear in adjacent passages.

First, the system calculates the similarity between the passages and the user query. Next, the system determines the similarity of the documents that contain these passages making use of the best passage similarity measure. This approach

is based on the fact that if a passage is relevant then the document is also relevant.

As most of IR systems, IR-n system uses also techniques of query expansion. In current version, the most frequent terms in the documents are added.

### 4.2 The Extension

The new modules needed to extend the IR system are analyzed below.

- **Selection module.** First, a list of verbs related to the verb in the query is obtained. In order to do this, an electronic lexical database has been used, WordNet [13]. In it, nouns, verbs, and adjectives are organized into synonym sets, each representing underlying lexical concepts. To create WordNet several kinds of semantic relations were used, such as synonymy, hyponymy, meronymy and antonymy. In the case of verbs, this semantic relations have been adapted to fit the semantics of them (for instance, troponymy is the adaptation of hyponymy).
  In our system, the list of related verbs is extracted making use of synonymy and troponymy relations.
  Secondly, the verbs of the sentences of the passages retrieval by IR-n are compared with this list of verbs. So, only passages containing sentences with one of these verbs are selected and those sentences are marked.
- **Annotation module.** The sentences marked in the previous module are annotated with semantic information by using the SemRol method. So, the argument boundaries of the sentences are recognized and the semantic roles that fill this arguments are identified.
  As a result, a set of annotated sentences with the roles of the arguments of the verbs is obtained.

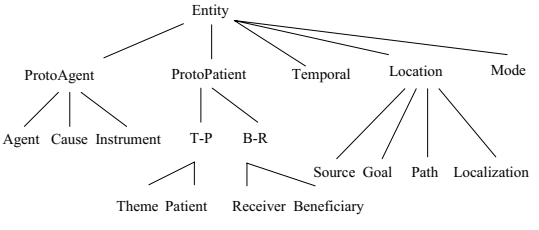



**Fig. 3.** Set of roles.

The set of roles [16] used for the annotation process is shown in the figure 3. This set of roles is different to the set of roles used for testing the SemRol method (See section 3). It is not a problem because a mapping between both set of roles can be done easily.

– **Module of heuristics.** Depending on the kind of question a different set of roles could be considered. So, it is possible to define a set of heuristics in order to establish a relationship between questions and semantic roles. For instance, questions such as "When", "What + time expression" or "In what + time expression" must be answered with the Temporal semantic role and must not be answered with the Agent, Patient, Location, Cause or Mode semantic role; and "Where", "In where + location expression" or "In what + location expression" must be answered with the Location semantic role and must not be answered with the Agent, Patient, Temporal, Cause or Mode semantic role. A summary of these heuristics is shown in figure 4.

| Question | Role | | No role |
|---|---|---|---|
| Where<br>In where<br>In what + exp<br>At what + exp | Location | | ProtoAgent<br>Mode<br>Temporal<br>Cause<br>ProtoPatient |
| When<br>In what + exp<br>What + exp | Temporal | | ProtoAgent<br>Mode<br>Location<br>Cause<br>ProtoPatient |
| How | Mode<br>Theme (if it is a diction verb) | | ProtoAgent<br>Location<br>Temporal<br>Cause<br>Patient<br>Beneficiary |
| Who | Agent - ProtoAgent<br>Patient - ProtoPatient | | Mode<br>Temporal<br>Location<br>Theme<br>Beneficiary |
| What | Cause<br>Theme | | |
| Whose | Receiver<br>Beneficiary<br>Patient | ProtoPatient | Agent<br>Location<br>Mode<br>Temporal<br>Theme<br>Cause |

**Fig. 4.** Set of heuristics.

Then, making use of these rules only the sentences containing the right semantic roles are selected and the number of passages retrieval is reduced.

## 5 Conclusions and Working in Progress

In this paper, an extension of a IR system using semantic role information is presented. When a query is done, the IR system IR-n retrieves a set of passages. Then the verbs of the sentences of these passages are compared with a list of verbs related with the verb of the query. Next, the sentences containing a verb of this list are annotated with semantic roles by using the SemRol method. Finally, several heuristics are applied in order to establish a relation between queries and semantic roles. So, only the sentences containing with the right semantic roles are selected and the number of passages retrieved is reduced.

The SemRol method is used to annotate selected sentences with semantic information. This method, based on conditional Maximum Entropy (ME) Probability Models, identifies and labels the constituents of a sentence with semantic roles. It consists of three phases. First, the process to obtain the semantic role needs the sense of the target verb. After that, several heuristics are applied in order to obtain the argument boundaries of the sentence. And finally, the semantic roles that fill these arguments are obtained.

Currently, we are developing the extension modules. Shortly, we will show results about this IR system extended with semantic role information and will evaluate them in appropriate forum.

On the other hand, it is important to say that the current version of the SemRol method only works with English corpus. In order to overcome this limitations, some kind of adaption must be done.

## References


1. UMLS Unified Medical Language System (2005AA release). http://www.nlm.nih.gov/research/umls/umlsdoc.html, January 2005.
2. R.A. Baeza-Yates and B.A. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press / Addison-Wesley, 1999.
3. X. Carreras and L. Màrquez. Introduction to the CoNLL-2004 Shared Task: Semantic Role Labeling. In *Proceedings of the Eighth Conference on Natural Language Learning (CoNLL-2004)*, Boston, MA, USA, Mayo 2004.
4. C. Fabre and P. Sebillot. Semantic interpretation of binomial sequences and information retrieval. In International ICSC Congress on Computational Intelligence: Methods and Applications (CIMA). Symposium on Advances in Intelligent Data Analisys (AID), editors, *Proceedings of the Symposium on Advances in Intelligent Data Analisys*, Rochester, USA, 1999.
5. L.S. Gay and W.B. Croft. Interpreting nominal compounds for information retrieval. *Information Processing and Management*, 26(1):21–38, 1990.
6. C. Horii, M. Imai, and K. Chihara. An information retrieval using conceptual index term for technical paper on digital library. In *Proceedings of International Symposium on Research, Development and Practice in Digital Libreries*, volume 97, pages 205–208, Tsukuba, November 1997. International Symposium on Research, Development and Practice in Digital Libreries (ISDL).
7. C. Horii, M. Imai, and K. Chihara. Information retrieval using conceptual index terms for technical papers in a digital library. *Systems and Computer in Japan*, 32(8), 2001.
8. Japan Electronic Dictionary Laboratory. EDR electronic dictionary technology guide (2nd edition, revised). http://www.iijnet.or.jp/edr, 1995.
9. G.Z. Liu. Semantic vector space model: Implementation and evaluation. *Journal of American Society for Information Science*, 48(5):395–417, 1997.
10. F. Llopis and R. Muñoz. Cross Language experiments with IR-n system. In *Proceedings of Workshop of Cross-Language Evaluation Forum (CLEF 2003)*, Trondheim, Norway, 2003.
11. F. Llopis and J.L. Vicedo. IR-n system, a passage retrieval system at CLEF 2001. In *Proceedings of Workshop of Cross-Language Evaluation Forum (CLEF 2001)*, pages 244–252, Darmstadt, Germany, 2001.

12. X. Lu. *An application of case relations to document retrieval*. PhD thesis, University of Western, Ontario, 1990.
13. G. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller. Five papers on wordnet. csl report 43. Technical report, Cognitive Science Laboratory, Princeton University, 1990.
14. P. Moreda, M. Palomar, and A. Suárez. Assignment of semantic roles based on word sense disambiguation. In *Proceedings of the 9TH Ibero-American Conference on AI*, Puebla, Mexico, Noviembre 2004.
15. P. Moreda, M. Palomar, and A. Suárez. Identifying semantic roles using maximum entropy models. In *Proceedings of the International Conference Text Speech and Dialogue*, Lecture Notes in Artificial Intelligence, Brno, Czech Republic, 2004. Springer-Verlag.
16. B. Navarro, P. Moreda, B. Fernández, R. Marcos, and M. Palomar. Anotación de roles semánticos en el corpus 3lb. In *Proceedings of the Workshop Herramientas y Recursos Linguísticos para el Español y el Portugués*, Tonantzintla, México, November 2004. Workshop Herramientas y Recursos Linguísticos para el Español y el Portugués. The 9TH Ibero-American Conference on Artificial Intelligence (IBERAMIA 2004).
17. M. Palmer, D. Gildea, and P. Kingsbury. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 2004. Submitted.
18. A. Ratnaparkhi. *Maximum Entropy Models for Natural Language Ambiguity Resolution*. PhD thesis, University of Pennsylvania, 1998.
19. T.C. Rindflesch and A.R. Aronson. Semantic processing in information retrieval. In *Proceedings of the 17th Annual Symposium on Computer Applications in Medical Care*, pages 611–615, 1993.
20. T.C. Rindflesch, C.A. Bean, and C.A. Sneiderman. Argument identification for arterial branching predications asserted in cardiac catheterization reports. In *Proceedings of the AMIA*, 2000.
21. C. Soo and G. Kho. The use of relation matching in information retrieval. *LIBRES: Library and Information Science Research*, 7(2), September 1997.
22. S. Vintar, P. Buitelaar, and M. Volk. Semantic relations in concept-based cross-language medical information retrieval. In *Inproceedings of the Workshop on Adaptive Text Extraction and Mining*. Workshop on Adaptive Text Extraction and Mining (ECML/PKDD), 2003.
23. J. Zhang, J. Mostafa, and Himansu Tripathy. Information Retrieval by Semantic Analysis and Visualization of the Concept Space of D-lib Magazine. *D-lib Magazine*, 8(10), October 2002.

# Text Categorization Based on Subtopic Clusters

Francis C.Y. Chik, Robert W.P. Luk, and Korris F.L. Chung

Department of Computing, Hong Kong Polytechnic University
{cscychik,csrluk,cskchung}@comp.polyu.edu.hk

**Abstract.** The distribution of the number of documents in topic classes is typically highly skewed. This leads to good micro-average performance but not so desirable macro-average performance. By viewing topics as clusters in a high dimensional space, we propose the use of clustering to determine subtopic clusters for large topic classes by assuming that large topic clusters are in general a mixture of a number of subtopic clusters. We used the Reuters News articles and support vector machines to evaluate whether using subtopic cluster can lead to better macro-average performance.

## 1 Introduction

Since document classification involves high-dimensional feature space, the effects of different feature reduction techniques were examined in order to improve recognition performance [1]. It is a well-known fact that the size of different text categories can vary significantly in text corpora. The Reuters-21578 collection is a common benchmark for comparing methods of text categorization [2-8]. The documents in the Reuters collection were collected from Reuters newswire in 1987. Over one third of the text classes are having less than 10 documents in the Reuters-21578 [2-3]. The skewness problem cannot be eliminated by replacing with a larger data set corpora like the Reuters Corpus Volume 1 (CRV1) [24], i.e. the uneven distribution of document sizes of topics within a data set will always occur, and may subsequently introducing problems for text categorization.

An unresolved problem for research on Text Categorization (TC) is how robust the methods are used to tackle problems with a skewed category distribution. Since categories typically have an extremely non-uniform distribution in practice [2], it would be meaningful to compare the performance of different classifiers with respect to category frequencies. Most commonly, methods are compared using a single score, such as the accuracy, error occurrence rate, or averaged F1 measure [2] over all category assignments to documents. A single-valued performance measure can be either dominated by the classifier's performance on common categories or rare categories, depending on how the average performance is computed. Two conventional methods are used to evaluate the performance average across categories. Micro averaging assigns equal weight to every document, while macro averaging assigns equal weight to each category [3]. Inevitably, skewed category distribution often leads to good micro-average performance but not so desirable macro-average performance.

To improve the macro-average performance, our approach is to break the large topic classes into subtopic classes, similar to the idea of passage-based retrieval [21], because large topics may have been generated by more than one term distribution [22]. The subtopic classes should have a significant amount of terms that occur in

A. Montoyo et al. (Eds.): NLDB 2005, LNCS 3513, pp. 203–214, 2005.
© Springer-Verlag Berlin Heidelberg 2005

documents of the subtopic but not in the other subtopic. We propose to use clustering [23] to find these subtopics of a large topic class as shown in Fig. 1. One important issue is to determine which topic classes are larger. This will be addressed by examining the performance with different thresholds to define large topic classes. By comparing the micro-average performance and macro-average performance before and after clustering, it is possible to identify if subtopic clustering has generated any positive result on the macro-average performance.

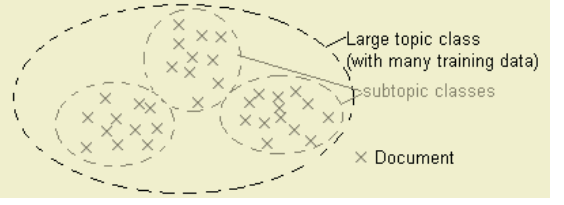



**Fig. 1.** Visual representation of a large topic class consists of a mixture of a number of subtopic clusters.

In Section 2, we shall briefly describe the methodology for experimental setup and performance measurements. This will be followed by results and discussion in Section 3. Lastly, conclusion and future work will be drawn in Section 4.

## 2 Methodology

### 2.1 Experimental Setup

**Data Set.** The Reuters-21578 document set has previously been regarded as a standard real-world benchmarking corpus for the Information Retrieval (IR) community. The ModApte split (training data set: 9,603 documents, test data set: 3,299 documents, unused: 8,676 documents) of Reuters-21578 document set is used for our experiments.

Except two large topics, including “acq” (1,488 training documents) and “earn” (2,709 training documents), the rest of the training topics have the number of documents below 500 (ranging from 1 to 460). Test documents can be assigned to more than one topic; therefore, 3,299 single-label test documents are expanded to 3,409 test documents which are used for the evaluation exercise.

The distribution of the number of training documents in a topic class is typically highly skewed. The number of terms in a topic increases logarithmically with an increase in the number of training documents. They are shown in Fig. 2.

**Preprocessing.** Preprocessing involves removing SGML tags, punctuation marks, stop words and performing word stemming to reduce the feature vector size. Bag-of-words [12] document representation (vector space model) scheme is used for feature representation. Term importance is assumed to be inversely proportional to the number of documents a particular term appears in. The term frequency (*tf*) and inverse document frequency (*idf*) are used to assign weights to terms. The inverse document frequency for term *t* is defined as [15]:

$$idf(t) = \log(N / n(t)) \quad . \tag{1}$$

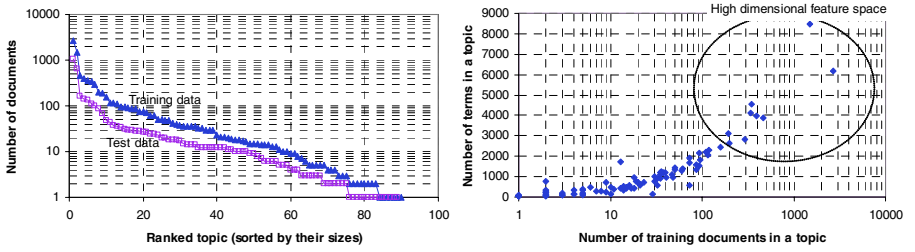



**Fig. 2.** The number of training/test documents plotted against ranked topic sorted by their sizes (left). The number of terms in a topic plotted against the number of training documents in its topic (right).

The common non-content words are removed to reduce possible interference in classification results. It is assumed that the importance of a term increases with its use-frequency. Combining these two assumptions lead to *tfidf*:

$$tfidf(t) = tf(t) \times idf(t) \quad . \tag{2}$$

Cosine normalization is used. Every document vector is divided by its Euclidean length, $((w_1)^2 + (w_2)^2 + \ldots + (w_n)^2)^{1/2}$, where $w_i$ is the *tfidf* weight of the *i*-th term in the document. The final weight for a term hence becomes:

$$\frac{tfidf \text{ weight}}{\text{Euclidean length of the document vector}} \quad . \tag{3}$$

**Classifier.** Instead of implementing a classifier, we use Rainbow/Libbow software package [16-17] to perform text classification. The classifier utilizes machine learning methods such as Naïve Bayes (NB), Support Vector Machine (SVM) and K-Nearest Neighbor (KNN) for text classification [2], [6-7]. As the major focus of this paper is not about the performance of classifier algorithms, only SVM classifier for single-label classification was selected for the following experiments. Scores of performance measurements generated by the classifier will be shown in the following section.

## 2.2 Performance Measurements

**Recall, Precision and F1.** Classification performance is measured by both recall and precision. For evaluating the performance, three quantities are of interest for each topic. They are: $a$ = the number of documents correctly assigned to this topic.

$b$ = the number of documents incorrectly assigned to this topic.

$c$ = the number of documents incorrectly rejected from this topic.

From these quantities, we define the following performance measures:

$$\text{recall} = a/(a+c) \quad . \tag{4}$$

$$\text{precision} = a/(a+b) \quad . \tag{5}$$

In addition, we use F1 measure [18], combining recall and precision with equal weighting, to compare the overall results of the algorithms:

$$\text{F1} = (2 \times \text{recall} \times \text{precision})/(\text{recall} + \text{precision}) \quad . \tag{6}$$

Macro-average performance scores are determined by first computing the performance measures per topic and then averaging these to compute the global means. Mi-

cro-average performance scores are determined by first computing the totals of *a*, *b* and *c* for all topics and then these totals are used to compute the performance measures. There is an important distinction between the two types of averaging. Micro averaging gives equal weight to every document, while macro averaging gives equal weight to each topic.

For a sample test data set containing 3,409 test documents, the measurements of recall, precision and F1 plotted against the training document number of 90 topics and against ranked topic (sorted by their scores from the smallest value to the largest) are shown in Fig. 3. It is observed that 61 out of 90 topics are having both recall and precision zero. The percentage of topics not classified correctly is 67.78%.

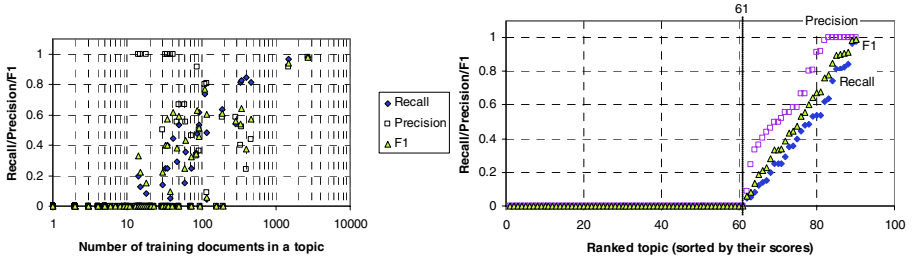



**Fig. 3.** The distribution of recall/precision/F1 measurement plotted against the number of training documents in a topic (left). The distribution of recall/precision/F1 measurement plotted against ranked topic sorted by their scores (right).

Recall, precision and F1 measurement of the 90 topics in the experimental data set are unevenly distributed. The uneven distribution is due to the fact that the distribution of the number of documents in the data set is highly skewed in nature. The results of macro-average and micro-average are shown in Table 1. From the result, the macro-average recall is 14.84%, macro-average precision is 22.35% and macro-average F1 is 17.84%. The reason for this low score is due to the fact that more than half of the topics (67.78%) in the data set are zero in both recall and precision.

**Table 1.** The macro-average and micro-average performance calculated by a sample test data set containing 3,409 test documents.

| Macro-average | | | Micro-average |
|---|---|---|---|
| Recall | Precision | F1 | Recall/Precision/F1 |
| 14.84% | 22.35% | 17.84% | 69.26% |

**Skewness.** Skewness is measured against the number of test data sets. Each test data set (consists of test documents) has the skewness, and its own scores (such as recall and precision) are calculated by the classifier.

The skewness is calculated by Kullback-Leibler (KL) distance [19]. Suppose two variables of the same type characterized by their probability distribution *f* and *f'*. The skew distance (KL distance) can be derived using as:

$$\text{skew distance} = \sum_{i=1}^{t} f_i(x) \times \log \frac{f_i(x)}{f_i'(x)} \ , \tag{7}$$

where $t$ is the number of topics, $f$ is the probability distribution of test documents of the topics and $f'$ is the equal probability distribution of test documents of the topics. For a data set containing of 90 topics, the skew distance is calculated as:

$$\text{skew distance} = \sum_{i=1}^{90} f_i(x) \times \log \frac{f_i(x)}{\frac{1}{90}} \quad . \tag{8}$$

$$f_i(x) = \frac{\text{number of test documents from topic } (i) \text{ in the test data set}}{\text{number of test documents from all topics in the test data set}} \quad . \tag{9}$$

For skewness measurement, we use 925 test data sets where 100 test documents in each test data set are selected randomly from 3,409 test documents. Each test data set has it own skew distance. Fig. 4 shows the histogram of skew distance of the 925 test data sets.

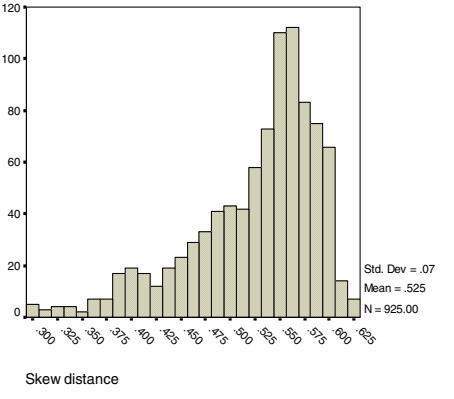



**Fig. 4.** The histogram of skew distance of 925 test data sets. 100 test documents in each test data set are selected randomly from 3,409 test documents.

For the 925 test data sets (925 skew distances), the scores of recall, precision and F1 are plotted against the skew distance. The scatter plots are shown in Fig. 5. On these plots, linear regression lines are drawn to predict the values at different skew distances. Zero skew distance is used as the reference point. The results at zero skew distance are shown in Table 2.

**Table 2.** The macro-average and micro-average performance at zero skew distance.

| Macro-average | | | Micro-average |
|---|---|---|---|
| Recall | Precision | F1 | Recall/Precision/F1 |
| 48.17% | 33.49% | 40.89% | 67.01% |

## 2.3 Clustering

By viewing topics as clusters in a high dimensional space, we propose the use of clustering to determine subtopic clusters for large topic classes by assuming that large topic clusters are in general a mixture of a number of subtopic clusters.

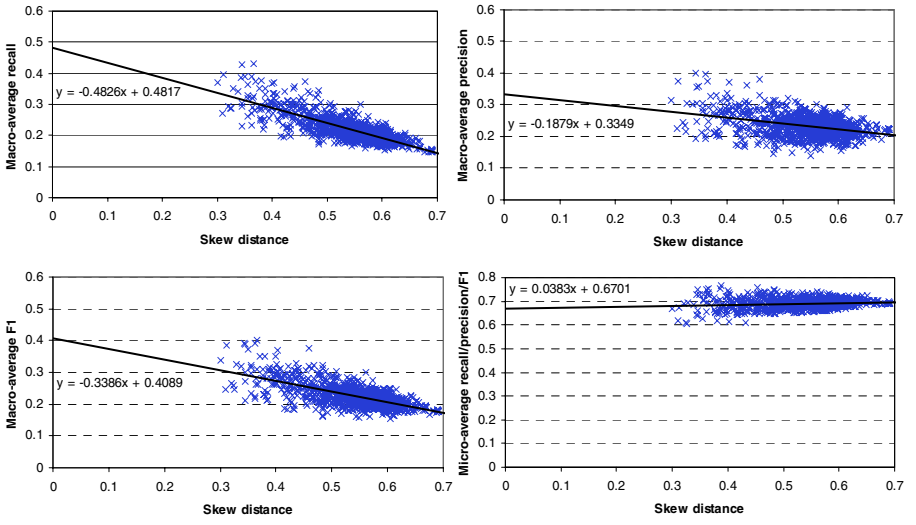



**Fig. 5.** Macro-average recall plotted against skew distance (top left). Macro-average precision plotted against skew distance (top right). Macro-average F1 plotted against skew distance (bottom left). Micro-average recall/precision/F1 plotted against skew distance (bottom right).

The cluster analyses (hierarchical and non-hierarchical clustering) in this paper are conducted by SPSS [20]. For each topic to be clustered into subtopics, all document vectors are initially grouped together to form a document-by-word matrix with size $m$ by $n$ ($m$ is the number of documents and $n$ is the size of document vector).

Topics with topic size which generates optimal macro-average performance (in Section 3.1) are selected for our experiment. For demonstration purpose, topics with topic size exceeding 100 are selected for clustering. Within the 90-topic data set, 77 topics have the number of training documents less than or equal to 100. Hence, only 13 topics meet our experimental criteria are selected for subtopic clustering. By means of complete linkage hierarchical clustering, 13 topics are clustered into 1,148 subtopics. The total number of topics and subtopics are 1,225 (77+1,148). By means of k-means non-hierarchical clustering, 13 topics have been clustered into 701 subtopics. The total number of topics and subtopics are 778 (77+701). The classifier is trained on these topics for performance evaluation. The clustered scores are compared with the previous result without subtopic clustering, by mapping clustered subtopics onto previous non-clustered topics after classification.

**Hierarchical Clustering.** The scores of recall, precision and F1 are plotted against the skew distance. The scatter plots are shown in Fig. 6. On these plots, linear regression lines are drawn to predict the values at different skew distances (zero skew distances are used as the reference point). The dotted lines are linear regressions showing the projected trends of micro-average and macro-average performance at different skew distances before subtopic clustering. Hence, the differences between the dotted and the solid lines in the graphs below demonstrate the difference in macro-average and micro-average performance before and after hierarchical clustering. Table 3 demonstrates the performance at zero skew distance.

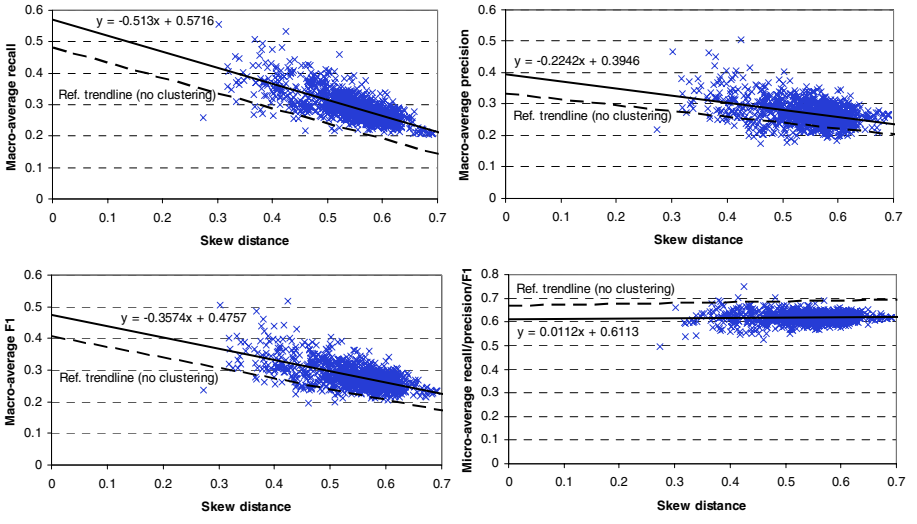



**Fig. 6.** Macro-average recall plotted against skew distance for hierarchical clustering (top left). Macro-average precision plotted against skew distance for hierarchical clustering (top right). Macro-average F1 plotted against skew distance for hierarchical clustering (bottom left). Micro-average recall/precision/F1 plotted against skew distance for hierarchical clustering (bottom right).

**Table 3.** The macro-average and micro-average performance at zero skew distance from the 925 test data sets (using subtopics by complete-linkage clustering to build the classifier).

| Macro-average | | | Micro-average |
|---|---|---|---|
| Recall | Precision | F1 | Recall/Precision/F1 |
| 57.16% | 39.46% | 47.57% | 61.13% |

**Non-hierarchical Clustering.** Non-hierarchical Clustering is conducted following the same procedure as Hierarchical Clustering. The scatter plots are shown in Fig. 7 and Table 4 demonstrates the performance at zero skew distance.

**Table 4.** The macro-average and micro-average performance at zero skew distance from the 925 test data sets (using subtopics by k-means clustering to build the classifier).

| Macro-average | | | Micro-average |
|---|---|---|---|
| Recall | Precision | F1 | Recall/Precision/F1 |
| 55.26% | 37.05% | 45.56% | 60.89% |

## 3 Experimental Results and Discussion

The comparison results of macro averaging and micro averaging at different cluster sizes by complete-linkage clustering are discussed in Section 3.1. They are calculated from the 925 test data sets at skew distance equals to 0. For macro-average performance, the optimal result is obtained when the maximum subtopic class size is set to 100.

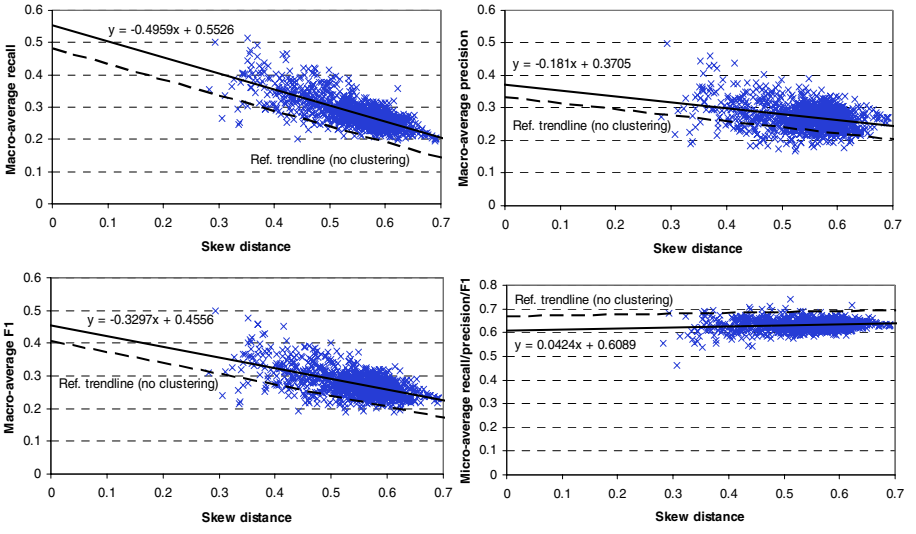



**Fig. 7.** Macro-average recall plotted against skew distance for non-hierarchical clustering (top left). Macro-average precision plotted against skew distance for non-hierarchical clustering (top right). Macro-average F1 plotted against skew distance for non-hierarchical clustering (bottom left). Micro-average recall/precision/F1 plotted against skew distance for non-hierarchical clustering (bottom right).

We have also evaluated whether the complete-linkage clustering is better than k-means clustering. In Section 3.2, the macro-average and the micro-average result with clustering and without clustering are summarized and compared. The results are also calculated from the 925 test data sets at skew distance equals to 0.

In Section 3.3, the percentages of topics never be classified correctly are summarized with subtopic clustered by complete-linkage clustering and k-means clustering. The scores are calculated from the sample test data set containing 3,409 test documents.

### 3.1 Comparison of Macro-averaging and Micro-averaging at Different Cluster Sizes by Complete-Linkage Clustering

To investigate the effect of topic/subtopic size, training documents with cluster-sizes limited to 5, 10, 25, 50, 100, 200 and 500 are classified by complete-linkage clustering. Fig. 8 shows the scatter plots and Table 5 shows the performance of the classifier with subtopic clustering for different maximum subtopic class sizes.

In general, the optimal macro-average performance (F1 measurement is 47.57%) is attained when the topic size is 100. However, at a certain point when the topic size is below 100, the macro-average performance and the micro-average performance nearly coincides (i.e. theirs scores are almost the same). Under such circumstance, over-clustering is likely to occur and adversely affect the macro-average and micro-average performance.

The best micro-average performance is achieved by using the classifier without subtopic clustering, mainly due to the benefit of large topics.

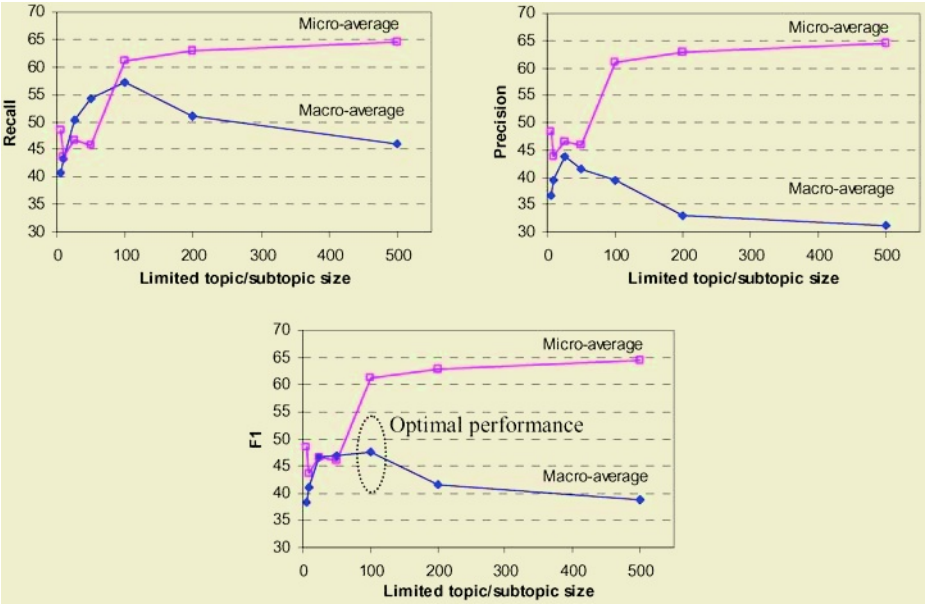



**Fig. 8.** Macro-average recall and micro-average recall plotted against limited topic/subtopic size by using complete-linkage method (top left). Macro-average precision and micro-average precision plotted against limited topic/subtopic size by using complete-linkage method (top right). Macro-average F1 and micro-average F1 plotted against limited topic/subtopic size by using complete-linkage method (bottom).

**Table 5.** The results from the 925 test data sets (at skew distance = 0) using complete-linkage clustering with topic/subtopic size limited to 5, 10, 25, 50, 100, 200 and 500 are summarized.

| Subtopic size limited to | Macro-average | | | Micro-average |
|---|---|---|---|---|
| | Recall | Precision | F1 | Recall/Precision/F1 |
| 5 | 40.64% | 36.72% | 38.35% | 48.43% |
| 10 | 43.31% | 39.48% | 41.04% | 43.70% |
| 25 | 50.36% | 43.90% | 46.71% | 46.59% |
| 50 | 54.16% | 41.42% | 46.91% | 45.88% |
| 100 | 57.16% | 39.46% | 47.57% | 61.13% |
| 200 | 51.00% | 32.98% | 41.64% | 62.82% |
| 500 | 46.02% | 31.09% | 38.73% | 64.44% |
| No clustering | 48.17% | 33.49% | 40.89% | 67.01% |

### 3.2 Comparison of Macro-averaging and Micro-averaging by Complete-Linkage Clustering and K-Means Clustering

The macro-average and micro-average result calculated from the 925 test data sets at zero skew distance using complete-linkage and k-means clustering with topic/subtopic size limited to 100 are summarized in Table 6. It shows that complete-linkage clustering performs better regardless of all performance measures. While we have to

accept that hierarchical clustering, such as complete-linkage, provides better performance than non-hierarchical clustering, as it is able to locate the cluster boundaries more accurately and create a higher performance in text categorization.

**Table 6.** The results from the 925 test data sets (at skew distance = 0) using complete-linkage clustering and k-means clustering with topic/subtopic size limited to 100 are summarized.

| Clustering method | Macro-average | | | Micro-average |
|---|---|---|---|---|
| | Recall | Precision | F1 | Recall/Precision/F1 |
| No clustering | 48.17% | 33.49% | 40.89% | 67.01% |
| Complete-linkage | 57.16% | 39.46% | 47.57% | 61.13% |
| K-means | 55.26% | 37.05% | 45.56% | 60.89% |

### 3.3 Comparison of Percentage of Topics with Zero Recall and Precision

The scores are calculated from a sample test data set containing 3,409 test documents. The measurements of recall, precision and F1 plotted against ranked topic (sorted by their scores from the smallest value to the largest) using complete-linkage clustering and k-means clustering are shown in Fig. 9. The results are summarized in Table 7 and show that the classifier with subtopic clustering by complete-linkage method has 18.03% improvement while the result by k-means method has 16.39% improvement. Again it shows that complete-linkage clustering performs better than k-means clustering.

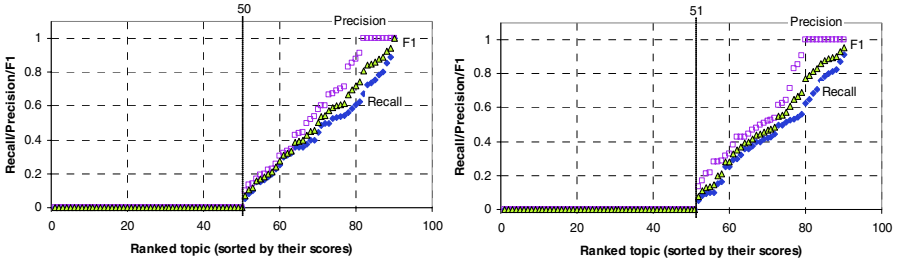



**Fig. 9.** The distribution of recall/precision/F1 measurement plotted against ranked topic sorted by their scores using complete-linkage clustering with topic/subtopic size limited to 100 (left). The distribution of recall/precision/F1 measurement plotted against ranked topic sorted by their scores using k-means clustering with topic/subtopic size limited to 100 (right).

**Table 7.** The percentages of topics that have never been classified correctly are summarized (without subtopic, with subtopic clustered by complete-linkage clustering and with subtopic clustered by k-means clustering).

| Clustering method | Topics that have never been classified correctly | Improvement |
|---|---|---|
| No clustering | 67.78% (61 out of 90) | - |
| Complete-linkage | 55.56% (50 out of 90) | 18.03% |
| K-means | 56.67% (51 out of 90) | 16.39% |

## 4 Conclusion and Future Work

We have shown that subtopic clustering of large topic classes can improve the macro-average performance consistently across different skewness of the test data set distribution. The optimal result shows that there is 16.34% improvement in macro-average performance (by F1 measurement) when the maximum subtopic size equals to 100 by using complete-linkage clustering.

This experiment shows that 100 is a useful threshold value which indicates whether there is a need to divide the large topic classes into subtopic classes (i.e. subtopic clustering) in order to increase macro-average performance. However, there is a slight decrease in the micro-average performance and more research is needed to enhance the use of subtopic clustering for text categorization. We will further explore how can the best size of the subtopic clusters can be determined analytically or automatically.

By comparing hierarchical clustering and non-hierarchical clustering, it shows that complete-linkage clustering performs better for recall, precision and F1 performances when the maximum subtopic size is at 100. For topics with zero recall and precision, there is an 18.03% improvement by complete-linkage clustering.

## References

1. Y. H. Li and A. K. Jain, Classification of text documents, The Computer Journal, 41(8), 537-546, 1998.
2. Y. Yang and X. Liu, A re-examination of text categorization methods, Proc. 22nd ACM SIGIR Conf., pp. 42-49, 1999.
3. K. Aas. and L. Eikvil, Text Categorisation: a survey, Technical Report #941, Norwegian Computing Center, 1999.
4. D. Lewis, Reuters-21578 text categorization test collection distribution 1.0, http://www.daviddlewis.com/resources/testcollections/reuters21578/.
5. S. Dumais, J. Platt, D. Heckerman, and M. Sahami, Inductive learning algorithms and representations for text categorization, Technical Report, Microsoft Research, 1998.
6. T. Joachims, Text categorization with support vector machines: Learning with many relevant features, Proc. European Conference on Machine Learning, pp. 137-142, 1998.
7. Y. Yang, An evaluation of statistical approaches to text categorization, Technical Report CMU-CS-97127, Computer Science Department, Carnegie Mellon University, 1997.
8. R. Schapire and Y. Singer, Boostexter: a boosting-based system for text categorization, Machine Learning, 39(2),135-168, 2000.
9. H. Schütze, Single-link, complete-link & average-link clustering, NLP and Text Mining, http://www-csli.stanford.edu/~schuetze/.
10. C. Nicholas, J. Kogan and M. Teboulle, Tutorial on clustering large and high-dimensional data, http://www.csee.umbc.edu/~nicholas/clustering/.
11. A. Jain, M. Murty, and P. Flynn, Data clustering: a review, ACM Computing Surveys, 31(3), 263-323, 1999.
12. T. Mitchell, Machine Learning, McGraw-Hill, 1997.
13. G. Salton, A. Wong and C. S. Yang, A vector space model for automatic text retrieval, Communications of the ACM, 18(11), 613-620, 1975.
14. M. F. Porter, An algorithm for suffix stripping, Program, 14 (3) 130-7, July 1980.
15. G. Salton and C. Buckley, Term weighting approaches in automatic text retrieval, Information Processing and Management, 24(5), 513-523, 1988.
16. A. McCallum, Rainbow, http://www.cs.cmu.edu/~mccallum/bow/rainbow/.
17. Website: http://www-2.cs.cmu.edu/~mccallum/bow/.

18. C. J. van Rijsbergen. Information Retrieval, Butterworths, London, 1979.
19. S. Kullback and R. Leibler, On information and sufficiency, Annals of Mathematical Statistics, 22, 79–86, 1951.
20. Website: http://www.spss.com/.
21. J.P. Callan, Passage-level evidence in document retrieval. Proc. 17th ACM SIGIR Conf., pp. 302-310, 1994.
22. H. Takamura and Y. Matsumoto, Two-dimensional clustering for text categorization, Proc. of CoNLL-2002, pp. 29-35, 2002.
23. V. Hatzivassiloglou, L. Gravano, and A. Maganti, An investigation of linguistic features and clustering algorithms for topical document clustering. Proc. 23rd ACM SIGIR Conf., pp. 224-231, 2000.
24. Reuters Corpus, Volume 1, English language (Release date 2000-11-03, Format version 1, correction level 0), http://about.reuters.com/researchandstandards/corpus/.

# Interpretation of Implicit Parallel Structures. A Case Study with "vice-versa"

Helmut Horacek[1] and Magdalena Wolska[2]

[1] Fachrichtung Informatik
Universität des Saarlandes, Postfach 15 11 50, D-66041 Saarbrücken, Germany
horacek@ags.uni-sb.de
[2] Fachrichtung Computerlinguistik
Universität des Saarlandes, Postfach 15 11 50, D-66041 Saarbrücken, Germany
magda@coli.uni-sb.de

**Abstract.** Successful participation in task-oriented, inference-rich dialogs requires, among other things, understanding of specifications implicitly conveyed through the exploitation of parallel structures. Several linguistic operators create specifications of this kind, including "the other way (a)round", "vice-versa", and "analogously"; unfortunately, automatic reconstruction of the intended specification is difficult due to the inherent dependence on given context and domain. We address this problem by a well-informed reasoning process. The techniques applied include building deep semantic representations, application of categories of patterns underlying a formal reconstruction, and using pragmatically-motivated and domain-justified preferences. Our approach is not only suitable for improving the understanding in everyday discourse, but it specifically aims at extending capabilities in a tutorial dialog system, where stressing generalities and analogies is a major concern.

## 1 Introduction

Specifications implicitly conveyed through the exploitation of parallel structures are an effective means of human communication. Unfortunately, handling these utterances adequately is problematic for a machine, since a formal reconstruction of the fully explicit representations may be associated with ambiguities in a number of cases. Their correct interpretation typically requires some degree of context understanding and domain knowledge, which are notorious weaknesses of automated methods. Consider, for example, the following statement made by a student in an experiment with a simulated tutorial dialog about proving theorems in elementary set theory [2]: "If all $A$ are contained in $K(B)$ and this also holds *vice-versa*, these must be identical sets" ("$K$" stands for "complement" here). The domain-adequate interpretation of the operator "vice-versa" is ambiguous here in that it may operate on immediate dependent relations or on the embedded relations. In precise terms, the implicit specification "and this also holds vice-versa" may be interpreted as "all $K(B)$ are contained in $A$" or as "all $B$ are contained in $K(A)$". The fact that the *Containment* relation is asymmetric and the context of the task – which is proving that "If $A \subseteq K(B)$,

A. Montoyo et al. (Eds.): NLDB 2005, LNCS 3513, pp. 215–226, 2005.
© Springer-Verlag Berlin Heidelberg 2005

then $B \subseteq K(A)$" holds – suggest that the second interpretation is meant. In the context of a tutorial session, it would be a suitable strategy to assume the more plausible interpretation to be the intended one. This enables the tutorial system to focus the response on the incorrect conclusion made by the student, that is, about the identity of the sets referred, rather than starting a boring clarification subdialog, the usual strategy in task-oriented human-machine dialog.

As shown in this example, successful participation in task-oriented, inference-rich dialogs requires understanding of specifications implicitly conveyed by parallel structures. There are several lexical devices that create such specifications, e.g. "the other way (a)round", "vice-versa", "analogously". Due to the problem complexity, in this paper, we concentrate on "vice-versa". We address the problem by a well-informed reasoning process. Applied techniques include building deep semantic representations, application of patterns underlying formal reconstruction, and using pragmatically-motivated and domain-justified preferences.

The outline of this paper is as follows. We describe phenomena observed. Then we illustrate our natural language analysis techniques. Subsequently, we categorize the underlying interpretation patterns, and we describe an algorithm for using these patterns in context. Finally, we discuss the embedding of the interpretation techniques in the tutorial system we are currently developing.

## 2 Data Collected from Corpora

In order to learn about regularities in reconstructions of the underlying explicit form of propositions specified by "vice-versa" or similar operators, we looked at several corpora. They include Negra and Frankfurter Rundschau corpora, our own corpus of tutorial dialogs [10], and results of several internet searches. We looked at the German phrasings "andersrum" and "umgekehrt", and their English equivalents "vice-versa" and "the other way (a)round". We only consider instances where the parallel structure with some pair of items swapped is not stated explicitly. Moreover, we do not deal with "andersrum" as expressing a purely physical change. Such a change can be either *intrinsic*, in terms of orientation of the object referred to by "andersrum", or *extrinsic*, i.e. with respect to another object related to the target object (see the Bielefeld corpus [3]).

The classification of "vice-versa" utterances presented in Fig. 1, reflects the role of the text portions that must be swapped for building the parallel proposition conveyed implicitly. The examples demonstrate that the task of reconstructing the proposition left implicit in the text may be tricky.

The first and structurally simplest category concerns swapping two *case role* fillers. This may be applied to Agent and Patient roles, as in (1), or to two directional roles as in (2). In the last example in this category, complications arise due to the fact that one of the arguments is missing on the surface and needs to be contextually inserted prior to building the assertions with exchanged directional arguments. Moreover, the case role swap can also work across clauses, when a parallel structure is the target for exchanging items, as in example (3). Complex interrelations may occur when the fillers themselves are composed structures, is

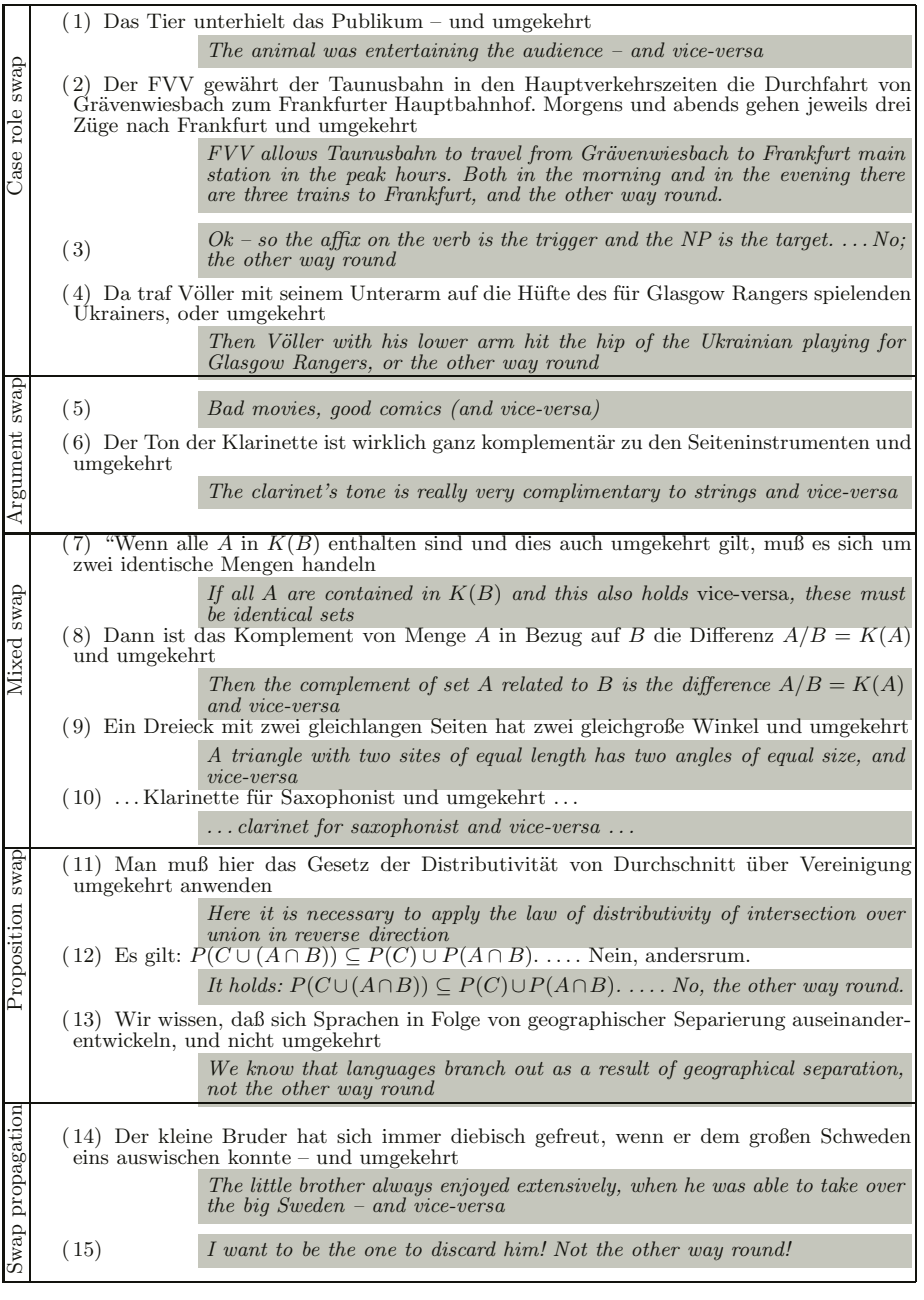



**Fig. 1.** Examples of utterances with "vice-versa" or similar operators.

in (4), which also makes swapping other pairs of items structurally possible. In this example, the need for exchanging the persons including their body parts mentioned rather than the mere body parts or just the persons requires some

degree of physical understanding and is associated with complications in building the intended proposition with exchanged arguments.

The second category comprises examples of swapping applied to *arguments* of two constituents rather than to the constituents themselves. A simple example is (5); this utterance, however, is ambiguous, since it could also be categorized as a *case role* swap. Similarly to (3), a contextually-motivated enhancement may be required prior to applying a swapping operation; in (6), this involves a metonymic extension, i.e. expanding the "strings" to "the strings' tones".

The third category comprises occurrences of a "mixed" form of the first two, with one argument substituted for a constituent which, in turn, takes the position of this argument in the reconstructed form. The first example in this category, (7), has already been discussed in the Introduction. The next one, (8), also from our own domain, shows a case with repeated occurrences of the items to be swapped. Moreover, swapping the items $A$ and $B$ must be propagated to the included formula. The following example, (9), is handled easier when applying the exchange on basis of the surface structure, thereby swapping the properties of a triangle for the reconstructed assertion. If a deeper structure of the sentence's meaning is built, this would amount to an implication, expressing that a triangle having two sides of equal length implies that this triangle has two equal sides. For such a structure, the reconstruction would fall into the next category, exchange of the order of two propositions: reversing an implication in this case. In (10), the lexeme "Saxophonist" needs to be expanded into "Saxophone" and "Spieler" ("player"), prior to performing the exchange.

The fourth category constitutes a swap of entire "propositions"; in the domain of mathematics, this may pertain to formulas. In (11), swapping applies to the sides of the equation referred to descriptively by the distributivity law. In (12), this applies to the arguments of the set inclusion relation, when the arguments are interpreted as propositions. The last example, (13), requires a structural recasting in order to apply the appropriate swapping operation. When the utterance is rebuilt around the $RESULT$ relation, expressed as an optional case role on the surface, swapping the two propositions related – "branching out of languages" and "geographical separation" – yields the desired result.

Examples in the last category involve a "propagation" of a swapping operation, e.g. an exchange performed on an embedded clause may be percolated to the embedding one. Candidates for such a propagation are expressions of propositional attitudes, as in (14) and (15). However, these examples demonstrate that the context determines whether propagation is required or not. While in (14), the joy of the Danes is complemented by the choice of the Swedes regarding competitions with opposite result, it is the contrast to the unwanted negated fact stressing the intention of the speaker in (15).

## 3 The Interpretation Procedure

In this section, we illustrate our technical contribution. It consists of three parts, each dealt with in a separate subsection: (1) the linguistic/semantic analysis, (2)

definitions of rules that support building parallel structures, and (3) the proper algorithm to make use of these rules.

### 3.1 Linguistic Analysis

The linguistic analysis consists of semantic parsing followed by contextually motivated embedding and enhancements.

We use a deep parser to construct a linguistic meaning representation of the input and a set of knowledge sources to assign the linguistic meaning a domain-specific interpretation ([9], [5]). The output of the deep parser is a relational structure representing a dependency-based deep semantics of the utterance in the sense of Prague School sentence meaning, as employed in the Functional Generative Description (FGD) at the tectogrammatical level [7]. In FGD, the central frame unit of a clause is the head verb which specifies the *tectogrammatical relations* (TRs) of its dependents (*participants/modifications*). A distinction is drawn between *inner participants*: Actor, Patient, Addressee, Effect, Origin, and *free modifications* (adjuncts), such as Location, Means, Direction (From, Where-To), Criterion, Time (From-When, To-When) [4]. Moreover, every valency frame also specifies which modifications are *obligatory* and which *optional*.

For example, the utterance (7) in Fig. 1 obtains the following interpretation[1]:

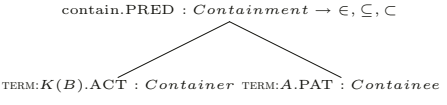



**Fig. 2.** Interpreted representation of the utterance "all $A$ are contained in $K(B)$".

which, in the context of an informal verbalization of a step in a naive set theory proof, translates into the following formal statement: "$\forall x, x \in A \Rightarrow x \in K(B)$".

The meaning representations are embedded within discourse context and discourse relations between adjacent utterances are inferred where possible based on the linguistic indicators (discourse markers). The nodes (heads) and dependency relations of the interpreted dependency structures as well as discourse-level relations serve as input to instantiate the reconstruction patterns. Moreover, at the sentence level, we also use the information about the status of a dependent (inner participant vs. free modification, obligatory vs. optional). Finally, contextual enhancements are carried out that are driven by needs for reconstruction, but can also serve other interpretation purposes (e.g. lexical or metonymic extension).

Based on analysis of our corpus of "vice-versa" examples, we have identified combinations of dependency relations that commonly participate in the swapping operation called for by "vice-versa" phrases. Examples of pairs of such relations at sentence-level are shown in Fig. 3. Similarly, in the larger discourse context, for example, arguments of $CAUSE$, $RESULT$, $CONDITION$,

[1] We present a simplified schematic representation of the tectogrammatical representations. Where necessary, for space reasons, irrelevant parts are omitted.

| |
|---|
| *Exchangeable*(ACTOR, PATIENT) |
| *Exchangeable*(DIRECTION-WHERE-FROM, DIRECTION-WHERE-TO) |
| *Exchangeable*(TIME-TILL-WHEN, TIME-FROM-WHEN) |
| *Exchangeable*(CAUSE, PRED[a]) |
| *Exchangeable*(CONDITION, PRED) |

[a] *PRED* is the immediate predicate head of the corresponding relation.

**Fig. 3.** Excerpt from the table of exchangeable relations.

*SEQUENCE* or *LIST* are likely candidates for a swapping operation. During processing, we use the association table as a preference criterion for selecting candidate relations to instantiate patterns. If one of the elements of a candidate pair is an *optional argument* that is not realized in the given sentence, we look at the preceding context to find the first instance of the missing element. Such is, for example, the case with the utterance (2), schematically presented in Fig. 4, which does contain a likely candidate for a swap according to our association table ($Direction - Where - To$), but where the corresponding pair ([ von Grävenwiesbach ].DIR-WHERE-FROM) has to be retrieved from the preceding utterance. Additionally, the utterance (10) would call for more complex procedures to identify the required metonymic expansion, however, so far we have concentrated on resolving metonymic references solely in the context of our specific domain.

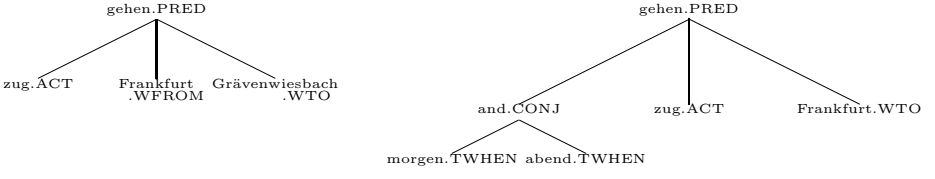



**Fig. 4.** Tectogrammatical representation of the discourse (2) in Fig. 1.

### 3.2 Interpretation Patterns

In order to accomplish the formal reconstruction task, we define rules that encapsulate specifications for building the implicit parallel text on the basis of the corresponding co-text. These rules consist of a pattern and an action part. The patterns are intended to be matched against the output of parser on a text portion in question, by identifying relevant case roles, and giving access to their fillers. Moreover, the patterns test constraints on the compatibility of candidates for swapping operations. The actions apply recasting operations on the items identified by the patterns, thereby building the implicit parallel text.

Within patterns, we perform category membership tests on the representation in question. Assuming $x$ referring to a semantic representation, $Pred(x)$

1a. *Case role* swap within the same clause

$$Pred(x) \wedge Case_1(x, y) \wedge Case_2(x, z) \wedge$$
$$Type - compatible(y, z) \wedge Exchangeable(Case_1, Case_2) \rightarrow Swap(x, y, z, x_p)$$

1b. *Case role* swap across two clauses

$$Conj(x) \wedge Case_1(x, y) \wedge Case(y, u) \wedge Case_2(x, z) \wedge Case(z, v) \rightarrow Swap(x, u, v, x_p)$$

2. *Argument* swap

$$Pred(x) \wedge Case_1(x, y) \wedge Case_{11}(y, u) \wedge Case_2(x, z) \wedge Case_{21}(z, v) \wedge$$
$$\neg(Case_1 = Case_2) \wedge Type - compatible(u, v) \rightarrow Swap(x, u, v, x_p)$$

3. *Mixed* swap

$$Pred(x) \wedge Case_1(x, y) \wedge Case_{11}(y, u) \wedge Case_2(x, z) \wedge$$
$$\neg(Case_1 = Case_2) \wedge Type - compatible(u, z) \rightarrow Swap(x, u, z, x_p)$$

4. *Proposition* swap

$$Subord(x) \wedge Case_1(x, y) \wedge Case_2(x, z) \wedge \neg(Case_1 = Case_2) \rightarrow Swap(x, y, z, x_p)$$

**Fig. 5.** Patterns for reconstruction of implicitly specified propositions.

is a logical function that checks if $x$ has a *Pred*-feature, i.e., it is an atomic proposition. Similarly, $Conj(x)$ and $Subord(x)$ perform more specific tests for complex propositions: coordination or subordination, respectively. Moreover, $Pred_1(x, x_1)$ accesses the first proposition and binds it to $x_1$, while $Pred_2(x, x_2)$ does this analogously for the second one. Within a proposition, case fillers and modifiers, are accessed by $Case(x, y)$, where $y$ specifies the filler of $Case$ in $x$, and indices are used to express constraints on identity or distinctiveness of case roles. In addition to access predicates, there are test predicates that express constraints on the identified items. The most basic one is $Type\text{-}compatible(x, y)$, which tests whether the types of $x$ and $y$ are compatible according to the underlying domain ontology. A more specific test is performed by $Exchangeable(Case_1, Case_2)$ to access the associations specified in the previous section.

The action part of the patterns is realized by operation $Swap(x, y, z, x_p)$ which replaces all occurrences of $x$ in $z$ by $y$ and vice-versa, binding the result to $x_p$. Different uses of this operation manifest themselves in different instantiations of $y$ and $z$ with respect to the overarching structure $x$.

There are patterns for each category introduced in Sec. 2, except for *Swap propagation* (Figure 5). All patterns are tested on a structure $x$ and, if successful, the result bound is to a structure $x_p$. For the first category, *case role* swap, there are two patterns. If the scope of the swap is a single clause (1a), two case roles identified as exchangeable are picked and their fillers must be compatible in types. If the swapping operation overarches two clauses (1b), the connecting relation must be a conjunction, and constituents filling the same case role are subject to the swapping operation. For the *argument* swap (2), type compatible arguments of distinct cases are picked. For the *mixed* swap (3), a case role filler is picked, as in (1a) and a type-compatible argument of another case, as in pattern (2). Finally, the *proposition* swap (4) merely inverts the order of the two clauses.

The pattern matching tests are associated with assigning marks to the results in some cases, which express additional information about plausibility or implausibility of the result. So far, we consider two criteria: (a) knowledge about symmetry or asymmetry of the relation (the *Pred* feature) whose cases are subject to the swapping operation: if such a relation is known as asymmetric, the result is considered implausible due to semantic reasons, if it is symmetric, due to pragmatic reasons, since the converse proposition conveys no new information; (b) a positive criterion to enhance the plausibility is mentioning of the implicitly referred candidate proposition in the previous discourse.

To extend the functionality of these straightforward patterns, we have defined a set of recasting rules (Fig. 6) invoked to reorganize the semantic representation prior to testing applicability of a suitable reconstruction rule. In contrast to the measures of inserting incomplete information contextually and expanding metonymic relations, as illustrated in the context of the linguistic analysis at the beginning of this section, the recasting operations are purely motivated for accommodating semantic representations for this purpose. We have defined three recasting rules (numbered accordingly in Fig 6):

1. *Lexical recasting*
   The semantics of some lexemes conflates the meaning of two related items. If one of them is potentially subject to a swapping operation, this item is not accessible for that operation without also affecting the other item so closely related to it. To deal with this problem, the representation of such lexemes is expanded, provided there is a sister case with a filler that is type compatible to the item newly created through the expansion.
2. *Case recasting*
   The dependency among items may sometimes not be reflected by the dependencies in the linguistic structure. Specifically, a dependent item may appear as a sister case in overarching case frame. The purpose of this operation is to build a uniform representation, by removing the dependent case role filler and inserting it as a modifier of the item it is dependent on.

---

1. *Lexical expansion*

$$Pred(x) \wedge Case_1(x,y) \wedge Lex-Expand(y,u,Case,v) \wedge Case_2(x,z) \wedge \neg(Case_1 = Case_2) \wedge Type-compatible(v,z) \rightarrow Swap(x,y,Case(u,v),x_p) \wedge Swap(x_p,z,v,x_p)$$

2. *Recast optional case as a head of an obligatory case*

$$Pred(x) \wedge Case_1(x,u) \wedge Case_2(x,v) \wedge Type(u,tu) \wedge Type(v,tv) \wedge Recastable(tv,Case_2,tu,Case_3) \wedge Case_3(x,w) \wedge Type-compatible(v,w) \wedge \neg(Case_1 = Case_2) \wedge \neg(Case_1 = Case_3) \wedge \neg(Case_2 = Case_3) \rightarrow Swap(x,u,v,x_p) \wedge Add(x_p,Case_3(v,u)) \wedge Remove(x_p,Case_2)$$

3. *Recast an optional case as a discourse relation*

$$Pred(x) \wedge Case(x,y) \wedge Member(Case,Subords) \rightarrow Build(Case(x_p,Case_2(x_p,y) \wedge Case_1(x_p,Remove(x,y))$$

---

**Fig. 6.** Recasting rules.

*Build-Parallel-Structure (x)*

1. Determine possible scopes for the potential application of swapping operations

$Embedding \leftarrow \epsilon$, $Structures \leftarrow \epsilon$
**if** $Pred(x)$ **then** $Scopes \leftarrow \{x\}$ **else**
  **if** $Subord(x) \wedge Case_1(x, y) \wedge Case_2(x, z) \wedge (y \in Prop - att)$
    **then** $Scopes \leftarrow \{z\}$, $Embedding \leftarrow y$
      **else** $Scopes \leftarrow \{z, x\}$
**endif endif**

2. Match patterns and build swapped structures

**forall** $Scope_1$ in $Scopes$ **do**
  $Structures \leftarrow Structures \cup < Case - swap(Scope_1) > \cup$
    $< Argument - swap(Scope_1) > \cup < Argument - swap(Case - recast(Scope_1)) > \cup$
    $< Mixed - swap(Scope_1) > \cup < Mixed - swap(Lex - recast(Scope_1)) > \cup$
    $< Prop - swap(Scope_1) > \cup < Prop - swap(Prop - recast(Scope_1)) >$
**end forall**
**if** $Embedding$ **then**
  $Structures \leftarrow Structures \cup < Propagate - swap(Structures, Embedding) >$ **endif**
**return** Structures

**Fig. 7.** The algorithm for building interpretations for implicitly specified propositions.

3. *Proposition recasting*
   Apart from expressing a discourse relation by a connective, a proposition filling a subordinate relation may also be expressed as a case role. Again, the purpose of this operation obtaining uniformity, through lifting the case role filler and expressing the discourse relation as a multiple clause construct.

To implement the recasting operations, additional predicates are used. $Lex - Expand(y, u, Case, v)$ re-expresses the semantics of $y$ by $u$, accompanied by a $Case$ role filled by $v$. $Type(x, y)$ associates the type $y$ with item $x$. The type information is used to access the table $Recastable(t_1, C_1, t_2, C_2)$ to verify whether case $C_1$ with a filler of type $t_1$ can also be expressed as case $C_2$ with a filler of type $t_2$. $Build(x)$ creates a new structure $x$. $Remove(x, y)$ is realized as a function, deleting occurrences of $y$ in $x$, and $Add(x, y)$ expands $x$ by an argument $y$.

### 3.3 The Parallel Structure Building Algorithm

In this section, we describe how we build implicitly conveyed parallel structures, based on the definitions of swapping operations, with optional incorporation of recasting operations if needed. The procedure consists of two main parts (see Fig. 7). In the first part, the scope for applying the swapping rules defined in Fig. 5 is determined, and in the second part, the results obtained when executing the rules are collected. Due to practical reasons, we introduce simplifications concerning the scope of "vice-versa" in the current formulation of the procedure. While the effect of this operator may range over entire paragraphs in some involved texts, we only consider single sentences with at most two coordinated clauses or one subordinated clause. We feel that this restriction is not severe for

uses in application-oriented systems, since it excludes only a few examples from the corpora we have examined, but none of the examples illustrated in Fig. 1.

Focusing on reconstruction in more detail, the procedure *Build-Parallel-Structure* takes the last input sentence $x$, examines its clause structure, and binds potential scopes to variable *Scopes*. For composed sentences, the entire sentence ($x$) as well as the second clause ($Case_2(x, z)$) is a potential scope for building parallel structures. If the first clause ($Case_1(x, y)$) expresses a propositional attitude, it is saved for a later propagation of the swapping operation.

In the second part of the procedure, each swapping pattern is subsequently tested for the two potential scopings, and results are accumulated in *Structures*. Thereby, the call $< X - swap(Scope_1) >$, with $X$ being one of $Case$, $Argument$, $Mixed$, and $Prop$ expresses the building of a set of all possible instantiations of the pattern specified when applied to $Scope_1$. Some of these operations are additionally invoked with alternative parameters, which are accommodated by a recasting operation fitting to the pattern used, that call being $< X - swap(Y - recast(Scope_1)) >$ with $Y$ being one of of $Case$, $Lex$, and $Prop$. Variants with swapping operations propagated to the first clause expressing a propositional attitude are carried out, in case *Embedding* is specified.

Linguistic analysis, structure reconstruction patterns, recasting rules, and the algorithms operating on top of these structures are formulated in a domain-independent way, also taking care that the tasks involved are clearly separated. Hence, it is up to a concrete application to elaborate lexical semantic definitions required (e.g. for a saxophonist to capture example (10) in Fig. 1) to populate the tables *Exchangeable* and *Recastable*, and to enhance preference criteria. For example, a domain-specific criterion is the unlikelihood of changes of fairly distant body parts (hip versus lower arm) in an observation, which makes an argument swap implausible in example (4) in Fig. 1, so that the alternative case role swap becomes the preferred interpretation.

## 4 Project Setting and Examples

Our research is part of the DIALOG project[2] [1]. Its goal is (i) to empirically investigate the use of flexible natural language dialog in tutoring mathematics, and (ii) to develop an experimental prototype system gradually embodying the empirical findings. The experimental system will engage in a dialog in written natural language to help a student understand and construct mathematical proofs. We envision a modular architecture, making use of the powerful proof system ΩMEGA [8]. Modular design enables detailed reasoning about the student's action and bears the potential of elaborate system responses.

In a corpus of dialogs collected in a Wizard-Of-Oz study [10], we have identified several instances of wording that involve implicit parallel structures. Utterances (7), (8), (11), (12) in Fig. 1 are examples of such formulations taken

[2] The DIALOG project is part of the Collaborative Research Center on *Resource-Adaptive Cognitive Processes* (SFB 378) at University of the Saarland [6].

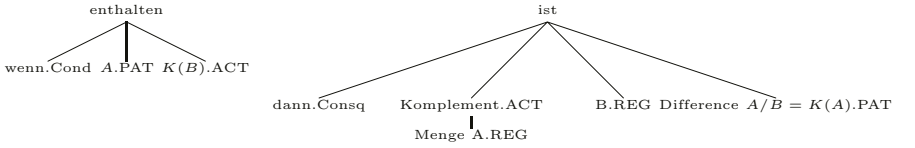



**Fig. 8.** Tectogrammatical representations of utterances (7), left, and (8), right.

directly from our corpus. Below, we briefly present the analyzes of the examples according to our interpretation procedure.

Utterance (7), whose tectogrammatical analysis is presented in Fig. 8, is a case of a *Mixed* swap. The relations Actor and Patient are found in the table *Exchangeable* (cf. Fig. 3) and are used to instantiate the pattern. In this example, the lower dependent is accessed based on the structure of the $K(B)$ constituent. We make use of information from a mathematical formula parser, employed at utterance pre-processing stage, which can identify the type of a mathematical expression based on the top-node operator. Here, $B$, of type set, is compatible with $A$, while $K$, an operator, is not. Another pattern that could apply in this case is a simple *Case role* swap, however, *Containment* being an asymmetric relation, makes this interpretation dispreferred. The decision as to whether the result is meaningful in the domain is deferred to the next stage of processing where a domain reasoner is used to evaluate the candidate interpretation in the context of the proof being developed.

A similar *Mixed* case interpretation is instantiated in utterance (8), where the Regard modifier of the Actor dependent is exchanged with the Regard dependent of the head predicate. Additionally, the exchange is propagated to all occurrences of the fillers, here: into the formula. An alternative instantiation is possible with the simple *Case role* swap, but is dispreferred on the grounds that it is not a likely candidate according to the table *Exchangeable*.

The last two examples, (11) and (12), by far domain-specific, are similar in that the swapping operation concerns arguments of a mathematical formula. In (11), the law of distributivity must be first linked to its formal axiomatic representation. Then, the arguments of the equality relation in the first utterance and of the subset relation in the latter can be instantiated as role fillers in a *Proposition* swap. Instantiation of the resulting variant of the distributivity axiom in the context of the given proof is left to the domain reasoner.

As this discussion illustrates, our method is able to adequately deal with the example sentences from our corpus and with the other examples exposed in Sec. 2, as far as building a restricted set of representation candidates, including one with the intended modification of item descriptions, is concerned. The correct selection of the most plausible candidate depends on elaborations within other system components, resp. on adequate representation fragments of everyday knowledge. In general, we can handle cases where the reconstruction of the explicit representation requires only *direct* structural recasting operations without prior inferencing, within the narrow scope of a few clauses.

## 5 Conclusions and Future Research

In this paper, we have presented techniques for formally reconstructing parallel structures implicitly specified by "vice-versa" or similar operators. We have addressed this problem by a domain-independent analysis method, that produces deep semantic and contextually enhanced representations, exploits recasting rules to accommodate linguistic variations into uniform expressions, and makes use of patterns matching major categories of parallel structures. Moreover, we have made domain-dependent interpretations of these techniques for a tutorial dialog context in a subdomain of mathematics.

Although we have devoted a lot of effort to build a principled method, the success is limited with respect to the generality of the problem – in some texts, the scope for building parallel structures overarches entire paragraphs, and deciding about the form of the reconstruction may require considerable inferencing. (see the collection in [11]). For our purposes, we are interested in expanding our method to other kinds of implicitly specified structures in the tutorial context, prominently interpretations of references to analogies, where structure accommodation and swapping of related items should also be prominent parts.

## References

1. C. Benzmüller, A. Fiedler, M. Gabsdil, H. Horacek, I. Kruijff-Korbayová, M. Pinkal, J. Siekmann, D. Tsovaltzi, B. Vo, and M. Wolska. Tutorial Dialogs on Mathematical Proofs. In *IJCAI Workshop on Knowledge Representation and Automated Reasoning for E-Learning Systems*, pp. 12–22, 2003.
2. C. Benzmüller, A. Fiedler, M. Gabsdil, H. Horacek, I. Kruijff-Korbayová, M. Pinkal, J. Siekmann, D. Tsovaltzi, B. Vo, and M. Wolska. A Wizard-of-Oz Experiment for Tutorial Dialogues in Mathematics. In *AIED2003 — Supplementary Proceedings of the 11th International Conference on Artificial Intelligence in Education*, pp. 471–481, Sydney, Australia, 2003.
3. http://www.sfb360.uni-bielefeld.de/
4. E. Hajičová, J. Panevová, and P. Sgall. A Manual for Tectogrammatical Tagging of the Prague Dependency Treebank. TR-2000-09, Charles University, Prague, 2000.
5. H. Horacek and M. Wolska. Interpreting Semi-Formal Utterances in Dialogs about Mathematical Proofs. In *Natural Language Processing and Information Systems (NLDB 2004)*, pp. 26–38, Springer, LNCS 3136, 2004.
6. SFB 378 web-site: http://www.coli.uni-sb.de/sfb378/.
7. P. Sgall, E. Hajičová, and J. Panevová. *The Meaning of the Sentence in its Semantic and Pragmatic Aspects*. Reidel Publishing Company, Dordrecht, 1986.
8. J. Siekmann et al. Proof Development with ΩMEGA. In *Proceedings of the 18th Conference on Automated Deduction*, pp. 144–149, Copenhagen, Denmark, 2002.
9. M. Wolska and I. Kruijff-Korbayová. Analysis of Mixed Natural and Symbolic Language Input in Mathematical Dialogs In *Proc. of the 42nd Annual Meeting of the Association for Computational Linguistics*, Barcelona, Spain, pp. 25–32. 2004.
10. M. Wolska, B. Q. Vo, D. Tsovaltzi, I. Kruijff-Korbayová, E. Karagjosova, H. Horacek, M. Gabsdil, A. Fiedler, and C. Benzmüller. An annotated corpus of tutorial dialogs on mathematical theorem proving. In *Proc. of 4th International Conference on Language Resources and Evaluation.*, Lisbon. pp. 1007–1010. 2004.
11. http://www.chiasmus.com/

# Text2Onto

## A Framework for Ontology Learning and Data-Driven Change Discovery

Philipp Cimiano and Johanna Völker

Institute AIFB, University of Karlsruhe
{pci,jvo}@aifb.uni-karlsruhe.de

**Abstract.** In this paper we present Text2Onto, a framework for ontology learning from textual resources. Three main features distinguish Text2Onto from our earlier framework TextToOnto as well as other state-of-the-art ontology learning frameworks. First, by representing the learned knowledge at a meta-level in the form of instantiated modeling primitives within a so called Probabilistic Ontology Model (POM), we remain independent of a concrete target language while being able to translate the instantiated primitives into any (reasonably expressive) knowledge representation formalism. Second, user interaction is a core aspect of Text2Onto and the fact that the system calculates a confidence for each learned object allows to design sophisticated visualizations of the POM. Third, by incorporating strategies for data-driven change discovery, we avoid processing the whole corpus from scratch each time it changes, only selectively updating the POM according to the corpus changes instead. Besides increasing efficiency in this way, it also allows a user to trace the evolution of the ontology with respect to the changes in the underlying corpus.

## 1 Introduction

Since ontologies provide a shared understanding of a domain of interest, they have become a key technology for semantics-driven modeling, especially for the ever-increasing need for knowledge interchange and integration. Semantic annotation of data with respect to a certain ontology makes it machine-processable and allows for exchanging this data between different applications. Therefore, ontologies are frequently used for the explicit representation of knowledge which is implicitly given by various kinds of data. Since building an ontology for a huge amount of data is a difficult and time consuming task a number of tools such as TextToOnto[1] [17], the ASIUM system [8], the Mo'k Workbench [2], OntoLearn [21] or OntoLT [3] have been developed in order to support the user in constructing ontologies from a given set of (textual) data. However, all these tools suffer from several shortcomings.

First of all, they all depend either on very specific or proprietary ontology models which can not always be translated to other formalisms in a straightforward way. This is certainly undesirable as ontology learning tools should be

[1] http://sourceforge.net/projects/texttoonto/

A. Montoyo et al. (Eds.): NLDB 2005, LNCS 3513, pp. 227–238, 2005.
© Springer-Verlag Berlin Heidelberg 2005

independent from a certain ontology model in order to be widely applicable and used. This is especially important in a context such as the Semantic Web in which different ontology models coexist next to each other. In Text2Onto[2] we overcome this problem by representing the learned ontological structures at a meta-level in form of so called modeling primitives rather than in a concrete knowledge representation language. As in [11], a collection of instantiated modeling primitives can then be translated into any target language. In this way we are able to handle the most prevalent representation languages currently used within the Semantic Web: RDFS, OWL and F-Logic.

Second, the interaction with end-users, in contrast to linguists or machine-learning specialists, has been largely neglected within such systems. As users are typically the ones who are most familiar with the domain, user interaction should be a central part of the system architecture. And third, most of these tools lack a certain robustness with respect to changes made to the data set. In fact, most state-of-the-art systems need to relearn the complete ontology once the underlying corpus has changed.

Text2Onto is a complete re-design and re-engineering of our system TextToOnto, a tool suite for ontology learning from textual data [17]. Text2Onto targets all these problems by introducing two new paradigms for ontology learning: (i) Probabilistic Ontology Models (POMs) which represent the results of the system by attaching a probability to them and (ii) data-driven change discovery which is responsible for detecting changes in the corpus, calculating POM deltas with respect to the changes and accordingly modifying the POM without recalculating it for the whole document collection. The benefits of these key design choices are various.

By assigning probabilities to the learned structures, the interaction with the user can be made more efficient by presenting him the learned structures ranked according to the certainty of the system or only presenting him the results above a certain confidence threshold. Moreover, in Text2Onto we store a pointer for each object in the POM to those parts of the document collection from which it was derived, allowing the user to understand why a certain concept, instance or relation was created and thus increasing the POM's traceability. And finally, the POM allows to maintain even inconsistent alternatives in parallel thus relegating the task of creating a consistent ontology to the user.

The benefits of data-driven change discovery are even more obvious. First, there is no need of processing the whole document collection when it changes thus leading to increased efficiency. Second, the user can explicitly track the changes to the ontology since the last change in the document collection thus being able to trace the evolution of the ontology with respect to changes in the underlying document collection.

This paper describes the framework and architecture of Text2Onto. It does not focus on the evaluation of the single ontology-learning algorithms, which will be presented elsewhere. Nevertheless, we also briefly describe the algorithms implemented in the framework so far.

[2] http://ontoware.org/projects/text2onto/

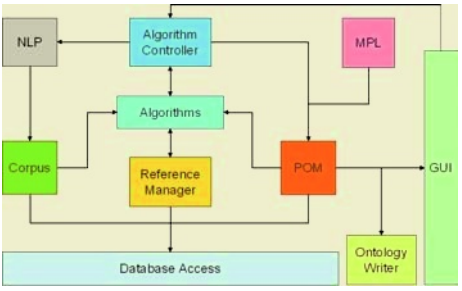



**Fig. 1.** Architecture of Text2Onto.

## 2 Architecture

The architecture of Text2Onto (cf. figure 1) is centered around the Probabilistic Ontology Model (see Section 2.1) which stores the results of the different ontology learning algorithms (cf. section 2.4). The algorithms are initialized by a controller, the purpose of which is (i) to trigger the linguistic preprocessing of the data, (ii) to execute the ontology learning algorithms in the appropriate order and (iii) to apply the algorithms' change requests to the POM. The fact that none of the algorithms has the permission of directly manipulating the POM guarantees maximum transparency and allows for the flexible composition of arbitrarily complex algorithms as described below.

The execution of each algorithm consists of three phases: First, in the *notification phase*, the algorithm learns about recent changes to the corpus. Second, in the *computation phase*, these changes are mapped to changes with respect to the reference repository, which stores all kinds of knowledge about the relationship between the ontology and the data (e.g. pointers to all occurrences of a concept). And finally, in the *result generation phase*, requests for POM changes are generated from the updated content of the reference repository.

The algorithms provided by the Text2Onto framework can be classified according to two different aspects: *task*, i.e. the kind of modeling primitives (see section 2.1) they produce, and *type*, that means the method which is employed in order to extract instances of the regarding primitives from the text. Each algorithm producing a certain kind of modeling primitive can be configured to apply several algorithms of different types and to combine their requests for POM changes in order to obtain a more reliable probability for each instantiated primitive (cf. [5]). Various types of pre-defined strategies allow for specifying the way the individual probabilities are combined.

### 2.1 The Probabilistic Ontology Model

A *Probabilistic Ontology Model* (POM) as used by Text2Onto is a collection of instantiated modeling primitives which are independent of a concrete ontology representation language. In fact, Text2Onto includes a *Modeling Primitive Library* (MPL) which defines these primitives in a declarative fashion. The obvious

benefits of defining primitives in such a declarative way are twofold. On the one hand, adding new primitives does not imply changing the underlying framework thus making it flexible and extensible. On the other hand, the instantiated primitives can be translated into any knowledge representation language given that the expressivity of the primitives does not exceed the expressivity of this target language. Thus, the POMs learned by Text2Onto can be translated into various ontology representation languages such as RDFS[3], OWL[4] and F-Logic [14]. In fact we follow a similar approach to knowledge representation as advocated in [11] and [19]. Gruber as well as Staab et al. adopt a translation approach to knowledge engineering in which knowledge is modeled at a meta-level rather than in a particular knowledge representation language and is then translated into different target languages. In Text2Onto we follow this translation-based approach to knowledge engineering and define the relevant modeling primitives in the MPL. So called *ontology writers* are then responsible for translating instantiated modeling primitives into a specific target knowledge representation language. The modeling primitives we use in Text2Onto are given below. The name of the corresponding primitive of Gruber's Frame Ontology is shown in parenthesis where applicable:

- concepts (CLASS)
- concept inheritance (SUBCLASS-OF)
- concept instantiation (INSTANCE-OF)
- properties/relations (RELATION)
- domain and range restrictions (DOMAIN/RANGE)
- mereological relations
- equivalence

It is important to mention that the above list is in no way exhaustive and could be extended whenever it is necessary. The motivation for considering exactly these relations is the fact that the algorithms integrated in the framework are currently only able to learn is-a, instance-of, part-whole as well as equivalence relations and restrictions on the domain and range of relations.

The POM is not probabilistic in a mathematical sense, but because every instantiated modeling primitive gets assigned a value indicating how certain the algorithm in question is about the existence of the corresponding instance. The purpose of these 'probabilities' is to facilitate the user interaction by allowing her to filter the POM and thereby select only a number of relevant instances of modeling primitives to be translated into a target language of her choice.

### 2.2 Data-Driven Change Discovery

In order to define the task of data-driven change discovery we first distinguish between *change capturing* and *change discovery*.

[3] http://www.w3.org/TR/rdf-schema/
[4] http://www.w3.org/TR/owl-features/

*Change capturing* can be defined as the generation of ontology changes from explicit and implicit requirements. Explicit requirements are generated, for example, by ontology engineers who want to adapt the ontology to new requirements or by the end-users who provide the explicit feedback about the usability of ontology entities. The changes resulting from this kind of requirements are called top-down changes. Implicit requirements leading to so-called bottom-up changes are reflected in the behavior of the system and can be induced by applying change discovery methods.

*Change discovery* aims at generating implicit requirements by inducing ontology changes from existing data. [20] defines three types of change discovery: (i) structure-driven, (ii) usage-driven and (iii) data-driven. Whereas structure-driven changes can be deduced from the ontology structure itself, usage-driven changes result from the usage patterns created over a period of time. Data-driven changes are generated by modifications to the underlying data, such as text documents or a database, representing the knowledge modeled by an ontology. Therefore, *data-driven change discovery* provides methods for automatic or semi-automatic adaption of an ontology according to modifications being applied to the underlying data set.

The benefits of data-driven change discovery are twofold. First, an elaborated change management system enables the user to explicitly track the changes to the ontology since the last change in the document collection thus being able to trace the evolution of the ontology with respect to changes in the underlying document collection. Second and even more important, there is no longer the need of processing the whole document collection when it changes thus leading to increased efficiency.

Independently from a particular application scenario some requirements have to be met by any application which is designed to support data-driven change discovery. The most important one is, of course, the need to keep track of all changes to the data. Each change must be represented in a way which allows for associating with it various kinds of information, such as its type, the source it has been created from and its target object (e.g. a text document). In order to make the whole system as transparent as possible not only changes to the data set, but also changes to the ontology should be logged. If ontological changes are caused by changes to the underlying data, the former should be associated with information about the corresponding modification to the data. Moreover, the system should allow for defining various *change strategies*, which specify the degree of influence changes to the data have with respect to the ontology or the POM respectively. This permits to take into account the confidence the user has in different data sources or the fact that documents might become out-dated after a while.

It is quite obvious that each algorithm in Text2Onto supporting automatic or semi-automatic data-driven change discovery requires a formal, explicit representation of two kinds of knowledge: First, knowledge about which concepts, instances and relations are affected by certain changes to the data and second, knowledge about how to react to these changes in an appropriate way, i.e. how

to update the POM in response to these changes. Consequently, the concrete knowledge to be stored by an ontology extraction system depends on the way these algorithms are implemented. A concept extraction algorithm, for example, might need to store the text references and term frequencies associated with each concept, whereas a pattern-based concept classification algorithm might have to remember the occurrences of all patterns matched in the text. Therefore, in Text2Onto each type of algorithm is provided with a suitable reference store (see section 2). It is among the algorithm controller's tasks to set up a suitable store each time a new algorithm is added.

### 2.3 Natural Language Processing

Many existing ontology learning environments focus either on pure machine learning techniques [2] or rely on linguistic analysis [3, 21] in order to extract ontologies from natural language text. Text2Onto combines machine learning approaches with basic linguistic processing such as tokenization or lemmatizing and shallow parsing. Since it is based on the GATE framework [7] it is very flexible with respect to the set of linguistic algorithms used, i.e. the underlying GATE application can be freely configured by replacing existing algorithms or adding new ones such as a deep parser if required. Another benefit of using GATE is the seamless integration of JAPE which provides finite state transduction over annotations based on regular expressions.

Linguistic preprocessing in Text2Onto starts by tokenization and sentence splitting. The resulting annotation set serves as an input for a POS tagger which in the following assigns appropriate syntactic categories to all tokens. Finally, lemmatizing or stemming (depending on the availability of the regarding processing components for the current language) is done by a morphological analyzer and a stemmer respectively.

After the basic linguistic preprocessing is done, a JAPE transducer is run over the annotated corpus in order to match a set of particular patterns required by the ontology learning algorithms. Whereas the left hand side of each JAPE pattern defines a regular expression over existing annotations, the right hand side describes the new annotations to be created (see listing 1.1). For Text2Onto we developed JAPE patterns for both shallow parsing and the identification of modeling primitives, i.e. concepts, instances and different types of relations (c.f. [13]).

**Listing 1.1.** JAPE pattern: Hearst

```
 (NounPhrase1):superconcept
 (
  {Token.kind == punctuation}
 )?
 {SpaceToken.kind == space}
 {Token.string == "such"}
 {SpaceToken.kind == space}
 {Token.string == "as"}
 {SpaceToken.kind == space}

 (NounPhrasesAlternatives):subconcept
):hearst1
-->
:hearst1.SubclassOfRelation = { rule = "Hearst1" },
:subconcept.Domain = { rule = "Hearst1" },
:superconcept.Range = { rule = "Hearst1" }
```

Since obviously, both types of patterns are language specific, different sets of patterns for shallow parsing and ontology extraction have to be defined for each language. Because of this and due to the fact that particular processing components for GATE have to be available for the regarding language, Text2Onto currently only supports ontology learning from English texts. Fortunately, thanks to recent research efforts made in the SEKT project[5] GATE components for the linguistic analysis of various languages such as German and Spanish have been made available recently. Since we want to provide full support for all of these languages in future releases of Text2Onto, we have already integrated some of these components, and we are currently working on the development of appropriate patterns for Spanish and German.

### 2.4 Algorithms

This section briefly describes for each modeling primitive the algorithms used to learn corresponding instances thereof. In particular we describe the way the probability for an instantiated modeling primitive is calculated.

**Concepts.** In Text2Onto we have implemented several measures to assess the relevance of a certain term with respect to the corpus in question. In particular, we implemented different algorithms calculating the following measures: Relative Term Frequency (RTF), TFIDF (Term Frequency Inverted Document Frequency), Entropy and the C-value/NC-value method in [10]. For each term, the values of these measures are normalized into the interval [0..1] and used as corresponding probability in the POM.

**Subclass-of Relations.** In order to learn subclass-of relations, in Text2Onto we have implemented various algorithms using different kinds of sources and techniques following the approach in [5]. In particular we implemented algorithms exploiting the hypernym structure of WordNet [9], matching Hearst patterns [13] in the corpus as well as in the World Wide Web and applying linguistic heuristics mentioned in [21]. The results of the different algorithms are then combined through combination strategies as described in [5]. This approach has been evaluated with respect to a collection of tourism-related texts by comparing the results with a reference taxonomy for this domain. The best result obtained was an F-Measure of 21.81%, a precision of 17.38% and a recall of 29.95%. As the algorithm already indicates the confidence in its prediction with a value between 0 and 1, the probability given in the POM can be set accordingly.

**Mereological Relations.** For the purpose of discovering mereological (part-of) relations in the corpus, we developed JAPE expressions matching the patterns described in [4] and implemented an algorithm counting the occurrences of patterns indicating a part-of relation between two terms $t_1$ and $t_2$, i.e. part-of($t_1$,$t_2$).

[5] www.sekt-project.com

The probability is then calculated by dividing by the sum of occurrences of patterns in which $t_1$ appears as a part. Further, as in the algorithm described above we also consult WordNet for mereological relations and combine the elementary probabilities with a certain combination strategy.

**General Relations.** In order to learn general relations, Text2Onto employs a shallow parsing strategy to extract subcategorization frames enriched with information about the frequency of the terms appearing as arguments. In particular, it extracts the following syntactic frames:

- transitive, e.g. love(subj,obj)
- intransitive + PP-complement, e.g. walk(subj,pp(to))
- transitive + PP-complement, e.g. hit(subj,obj,pp(with))

and maps this subcategorization frames to ontological relations. For example, given the following enriched subcategorization frame

hit(subj:*person*,obj:*thing*,with:*object*)

the system would update the POM with these relations:

hit(domain:*person*,range:*thing*)
hit_with(domain:*person*,range:*object*)

The probability of the relation is then estimated on the basis of the frequency of the subcategorization frame as well as of the frequency with which a certain term appears at the argument position in question.

**Instance of Relations.** In order to assign instances or named entities appearing in the corpus to their correct concept in the ontology, Text2Onto relies on a similarity-based approach extracting context vectors for instances and concepts from the text collection and assigning instances to the concept corresponding to the vector with the highest similarity with respect to their own vector as in [1]. As similarity measure we use the Skewed divergence presented in [15] as it was found to perform best in our experiments. Using this similarity measure as well as further heuristics, we achieved an F-Measure of 32.6% when classifying instances with respect to an ontology comprising 682 concepts [6]. Alternatively, we also implemented a pattern-matching algorithm similar to the one used for discovering part-of relations (see above).

**Equivalence.** Following the assumption that terms or concepts are equivalent to the extent to which they share similar syntactic contexts, we implemented algorithms calculating the similarity between terms on the basis of contextual features extracted from the corpus, whereby the context of a terms varies from simple word windows to linguistic features extracted with a shallow parser. This corpus-based similarity is then taken as the probability for the equivalence of the concepts in question.

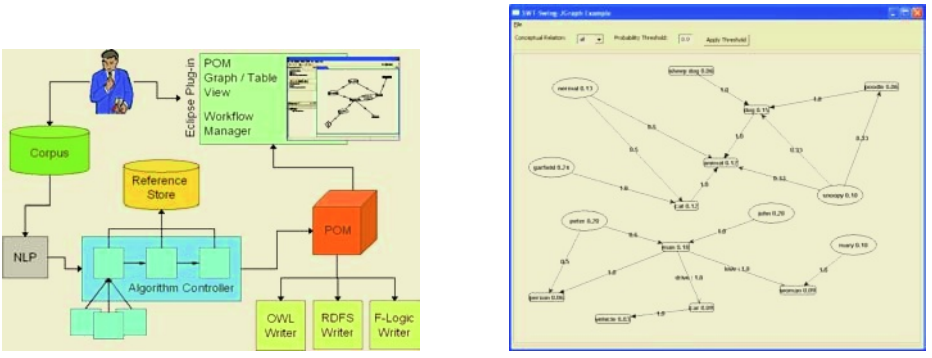



**Fig. 2.** Usage Scenario (left) and POM visualization (right).

## 3 Graphical User Interface

In addition to the core functionality of Text2Onto described above we developed a graphical user interface featuring a corpus management component, a workflow editor, configuration dialogues for the algorithms as well as tabular and graph-based POM visualizations. It will be available as an Eclipse[6] plug-in which could facilitate a smooth integration into ontology editors at a later development stage.

A typical usage scenario for Text2Onto is depicted by figure 2 (left). The user specifies a corpus, i.e. a collection of text, HTML or PDF documents, and starts the graphical workflow editor. The editor provides her with a list of algorithms which are available for the different ontology learning tasks, and assists her in setting up an appropriate workflow for the kind of ontology she wants to learn as well as to customize the individual ontology learning algorithms to be applied. Once the ontology learning process is started, the corpus gets preprocessed by the natural language processing component described in section 2.3, before it is passed to the algorithm controller. In the following, depending on the configuration of the previously specified workflow, a sequence of ontology learning algorithms is applied to the corpus. Each algorithm starts by detecting changes in the corpus and updating the reference store accordingly. Finally, it returns a set of requests for POM changes to its caller, which could be the algorithm controller, but also a more complex algorithm (cf. section 2). After the process of ontology extraction is finished, the POM is presented to the user.

Since the POM unlike any concrete ontology is able to maintain thousands of conflicting modeling alternatives in parallel, an appropriate and concise visualization of the POM is of crucial importance for not overwhelming the user with too much information. Although several pre-defined filters such as a probability threshold will be available for *pruning* the POM, some user interaction might still be needed for transforming the POM into a high-quality ontology. Currently, two different visualization types are available: a tabular view showing a number of sorted lists for all kinds of modeling primitives and a graph-based

[6] http://www.eclipse.org

representation which is depicted by figure 2 (right). After having finished her interaction with the POM, i.e. after adding or removing concepts, instances or relations, the user can select among various ontology writers, which are provided for translating the POM into different ontology representation languages.

## 4 Related Work

Several ontology learning frameworks have been designed and implemented in the last decade. The Mo'K workbench [2], for instance, basically relies on unsupervised machine learning methods to induce concept hierarchies from text collections. In particular, the framework focuses on agglomerative clustering techniques and allows ontology engineers to easily experiment with different parameters. OntoLT [3] is an ontology learning plug-in for the Protégé ontology editor. It is targeted more at end users and heavily relies on linguistic analysis. It basically makes use of the internal structure of noun phrases to derive ontological knowledge from texts.

The framework by Velardi et al., OntoLearn [21], mainly focuses on the problem of word sense disambiguation, i.e. of finding the correct sense of a word with respect to a general ontology or lexical database. In particular, they present a novel algorithm called SSI relying on the structure of the general ontology for this purpose. Furthermore, they include an explanation component for users consisting in a gloss generation component which generates definitions for concepts which were found relevant in a certain domain.

TextToOnto [17] is a framework implementing a variety of algorithms for diverse ontology learning subtasks. In particular, it implements diverse relevance measures for term extraction, different algorithms for taxonomy construction as well as techniques for learning relations between concepts [16]. The focus of TextToOnto has been so far on the algorithmic backbone with the result that the combination of different algorithms as well as the interaction with the user had been neglected so far. The successor Text2Onto targets exactly these issues by introducing the POM as a container for the results of different algorithms as well as adding probabilities to the learned structures to facilitate the interaction with the user.

Common to all the above mentioned frameworks is some sort of natural language processing to derive features on the basis of which to learn ontological structures. However, all these tools neglect the fact that the document collection can change and that it is unfeasible to start the whole learning process from scratch. Text2Onto overcomes this shortening by storing current results in stores and calculating POM deltas caused by the addition or deletion of documents. Very related is also the approach of [12] in which a qualitative calculus is presented which is able to reason on the results of different algorithms, resolving inconsistencies and exploiting synergies. Interesting is the dynamic aspect of the approach, in which the addition of more textual material leads to a reduction in the number of hypothesis maintained in parallel by the system.

Furthermore, as argued in the introduction, previously developed ontology learning frameworks all lack an explanation component helping the user to un-

derstand why something has changed in the underlying POM. In addition, most tools do not indicate how certain a learned object actually is, thus making it more difficult for the user to select only the most reliable findings of the system.

## 5 Conclusion and Outlook

We have presented our framework Text2Onto with the aim of learning ontologies from textual data. Its novel aspects as compared to similar frameworks are: (i) the independence of the actual ontology model or knowledge representation language, (ii) the introduction of probabilistic ontology models allowing more sophisticated models of user interaction and (iii) the integration of data-driven change discovery strategies increasing the efficiency of the system as well as the traceability of the learned ontology with respect to changes in the corpus, thus making the whole process more transparent.

In future versions of Text2Onto a graphical workflow engine will provide support for the automatic or semi-automatic composition of complex ontology learning workflows. For transforming the POM into a consistent OWL or RDF ontology we aim at a tight integration with the KAON evolution framework [20] which will allow to detect and resolve inconsistencies in the generated POMs. The development of the explanation component will be carried on with particular regard to the DILIGENT methodology [18]. By generating machine readable explanations we will make a major step in the direction of making Text2Onto part of the DILIGENT process. We are currently preparing an evaluation setting for comparing the results of the newly developed ontology learning algorithms with previous implementations provided by TextToOnto and other ontology learning tools. Moreover, a detailed user evaluation will offer valuable clues to the usability of the graphical user interface and the benefits gained from the availability of an explanation component.

## Acknowledgments

Research reported in this paper has been partially financed by the EU in the IST-2003-506826 project SEKT (http://www.sekt-project.com) and the IST-2001-34038 project Dot.Kom (http://www.dot-kom.org). We would like to thank our students Simon Sparn, Stephan Oehlert, Günter Ladwig and Matthias Hartung for their assistance in implementing the system and all our colleagues for fruitful discussions.

## References

1. E. Alfonseca and S. Manandhar. Extending a lexical ontology by a combination of distributional semantics signatures. In *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW)*, 2002.
2. G. Bisson, C. Nedellec, and L. Canamero. Designing clustering methods for ontology building - The Mo'K workbench. In *Proceedings of the ECAI Ontology Learning Workshop*, pages 13–19, 2000.

3. P. Buitelaar, D. Olejnik, and M. Sintek. OntoLT: A protégé plug-in for ontology extraction from text. In *Proceedings of the International Semantic Web Conference (ISWC)*, 2003.
4. E. Charniak and M. Berland. Finding parts in very large corpora. In *Proceedings of the 37th Annual Meeting of the ACL*, pages 57–64, 1999.
5. P. Cimiano, A. Pivk, L. Schmidt-Thieme, and S. Staab. Learning taxonomic relations from heterogeneous sources. In *Proceedings of the ECAI 2004 Ontology Learning and Population Workshop*, 2004.
6. P. Cimiano and J. Völker. Towards large-scale, unsupervised and ontology-based named entity recognition. Technical Report. AIFB, University of Karlsruhe, 2004.
7. H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the 40th Annual Meeting of the ACL*, 2002.
8. D. Faure and C. Nedellec. A corpus-based conceptual clustering method for verb frames and ontology. In *Proceedings of the LREC Workshop on Adapting lexical and corpus resources to sublanguages and applications*, 1998.
9. C. Fellbaum. *WordNet, an electronic lexical database*. MIT Press, 1998.
10. K. Frantzi, S. Ananiadou, and J. Tsuji. The c-value/nc-value method of automatic recognition for multi -word terms. In *Proceedings of the ECDL*, pages 585–604, 1998.
11. T. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.
12. U. Hahn and K. Schnattinger. Towards text knowledge engineering. In *AAAI/IAAI*, pages 524–531, 1998.
13. M. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th International Conference on Computational Linguistics*, pages 539–545, 1992.
14. M. Kifer, G. Lausen, and J. Wu. Logical foundations of object-oriented and frame-based languages. *Journal of the ACM*, 42:741–843, 1995.
15. L. Lee. Measures of distributional similarity. In *37th Annual Meeting of the Association for Computational Linguistics*, pages 25–32, 1999.
16. A. Maedche and S. Staab. Discovering conceptual relations from text. In W. Horn, editor, *Proceedings of the 14th European Conference on Artificial Intellignece (ECAI'2000)*, 2000.
17. A. Maedche and S. Staab. Ontology learning. In S. Staab and R. Studer, editors, *Handbook on Ontologies*, pages 173–189. Springer, 2004.
18. H. S. Pinto, C. Tempich, and S. Staab. Diligent: Towards a fine-grained methodology for distributed, loosely-controlled and evolving engingeering of ontologies. In *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI)*, 2004.
19. S. Staab, E. Erdmann, and A. Maedche. Engineering ontologies using semantic patterns. In *Proceedings of the IJCAI'01 Workshop on E-Business and Intelligent Web*, 2001.
20. L. Stojanovic. *Methods and Tools for Ontology Evolution*. PhD thesis, University of Karlsruhe, 2004.
21. P. Velardi, R. Navigli, A. Cuchiarelli, and F. Neri. Evaluation of ontolearn, a methodology for automatic population of domain ontologies. In P. Buitelaar, P. Cimiano, and B. Magnini, editors, *Ontology Learning from Text: Methods, Applications and Evaluation*. IOS Press, 2005. to appear.

# Interaction Transformation Patterns Based on Semantic Roles

Isabel Díaz[1,2], Lidia Moreno[2], Oscar Pastor[2], and Alfredo Matteo[1]

[1] Universidad Central de Venezuela – Laboratorio TOOLS – Escuela de Computacón
Ciudad Universitaria, Facultad de Ciencias, Caracas 1051, Venezuela
{idiaz,amatteo}@kuaimare.ciens.ucv.es
[2] Universidad Politécnica de Valencia – Dpto. de Sistemas Informáticos y Computación
Camino de Vera s/n, E-46022 Valencia, España
{idiaz,lmoreno,opastor}@dsic.upv.es

**Abstract.** This paper presents a strategy to deduce interactions from the text of use cases. This strategy is used by Metamorphosis: an automatic software production framework, conceived to facilitate the modelling of interactions of a system. Metamorphosis follows a linguistic engineering approach that is centred on the construction of models through the successive transformation of these models, in the definition of semantic roles and the application of design patterns. To obtain the Interaction Model of a system, three transformation levels are defined: the system, the use case, and the sentence. This paper focuses on how a transformation of a sentence is performed. Each transformation pattern specifies how to obtain information from the semantic context of a sentence, to deduce its corresponding interaction fragment. Some of the results obtained from the validation of these patterns are also presented.

## 1 Introduction

Metamorphosis is a framework to model interactions in an object-oriented automatic software production environment. An interaction is understood as the description of a behaviour unit which is described in terms of the information exchange between the system components. Metamorphosis assumes that the behaviour units are expressed differently, according to the Requirement Engineering activity being performed [1]. In the functional requirements specification, a behaviour unit is a use case that is described as a text written in natural language. This specification shows the actions that the system must perform when interacting with its users [2,3]. During the requirements analysis, the behaviour unit is expressed as an exchange of messages among instances that provides the system with the specified functionality. This system perspective is represented in an interaction model.

Many Software Engineering techniques have been proposed to deduce interactions from use cases [4,5,6]. These techniques are based on heuristics which application depends of the modeller experience. The analyst plays a decisive role in this task because he or she is the only owner of the knowledge used to construct the interaction model. The correspondence between a use case text and the obtained interaction is not explicitly registered. Thus, when the use case changes, it is not possible to determine what parts of the interaction must be modified or vice versa. On the other hand, there are some linguistic approaches that have proposed strategies to obtain modelled

A. Montoyo et al. (Eds.): NLDB 2005, LNCS 3513, pp. 239–250, 2005.
© Springer-Verlag Berlin Heidelberg 2005

primitives from texts [7,8,9]. These approaches are reasonably effective to establish persistent relationships between the text and the obtained model through predefined links between linguistic structures (that describe specific parts of the text) and model structures. However, the generated models are very limited because they produce elemental static models and very poor behaviour models.

The Metamorphosis framework can be situated in the midpoint between the linguistic approaches and the Software Engineering approaches. This framework allows to establish persistent relationship between the specification of the functional requirements of a system (Use Case Model) and its analysis (Interaction Model). Metamorphosis applies Natural Language Processing techniques to recognise in use case text the interaction elements. It is centred on the construction of models with a high level of abstraction and on the transformations of these models. The framework uses semantic roles and design patterns to achieve genericity and reusability.

The principal purpose of this paper is to describe the Metamorphosis transformation strategy. In particular, this work does emphasize on the linguistic aspects that should be considered to facilitate the interaction deduction. The paper is structured as follows. Section 2 describes the Metamorphosis models, the transformation defined between them and its linguistic perspective. Section 3 explains how a sentence transformation is carried out using language natural techniques. Section 4 presents an experiment performed to validate these patterns and the results. Section 5 describes the main researches that served as basis to formulate Metamorphosis. Finally, the last two sections are dedicated to the conclusions and references.

## 2 Metamorphosis: Transformation Strategy

The main objective of the transformation strategy is to describe how an interaction is deduced by Metamorphosis as a result of the text analysis of a use case. This strategy is based on a software development approach centred in which models are the primary artefacts. These models are Model Driven Architecture (MDA) compliant [10]. Following this approach, a target model (the Interaction Model) is obtained from the source model (the Use Case Linguistic Model) by transformation (Fig. 1). These are platform-independent models (PIMs) because they do not display implementation details. The model definition is abstract, at metamodel level, so that the transformation does not depend on neither the technological aspects nor on a specific domain. The transformation is automatic because it does not depend on the participation of the stakeholders to generate the basic Interaction Model.

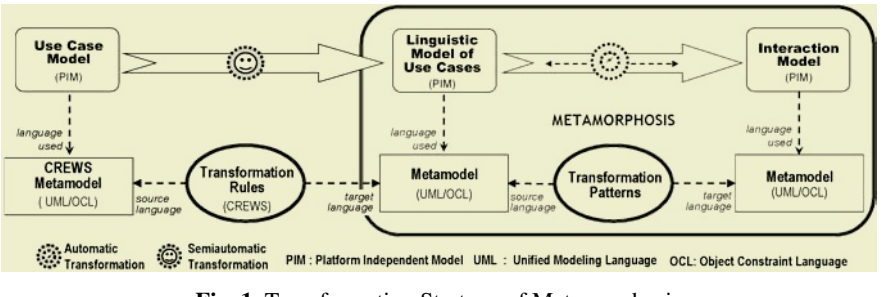



**Fig. 1.** Transformation Strategy of Metamorphosis

In Metamorphosis, the modelling elements are Unified Modelling Language (UML) compliant [11]. This language allows expressing the minimum set of concepts that is essential for specifying the system behaviour, in both the requirements specification and the analysis. Other concepts have been introduced using UML profiles by extending these elements. The following subsections describe each of the models that take part in this transformation. The transformation levels are presented too.

### 2.1 The Use Case Linguistic Model

The fundamental inputs of the Metamorphosis transformation are the use cases. This specification takes the form of a text that is written in natural language. The recognition of the text's linguistic features is of primary importance for the transformation process. To express such abstractions, the UML UseCases Package was extended with the concepts that allow to express such properties [11]. This tailoring of the UML concepts is described in the Metamorphosis Use Case Linguistic Profile. Two considerations were taken into account for this model:

- Many of the profile modelling elements are based on the concepts set forth by CREWS (Cooperative Requirements Engineering With Scenarios) in [3,12]. In these works, the study of the use cases is approached from the linguistic perspective for the purpose of improving their quality.
- It has been assumed that the use cases, which will be subject to transformation, have been previously normalised. The main objective of the normalisation process is generates a standard documentation of the use case. The normalisation wants that the use case text will be specified following the CREWS style and content guidelines [12]. The style guidelines indicate how to write a use case, while the content guidelines establish what information can be supplied by it. The purpose of these guidelines is not restricting the natural language used in the use cases with the intention of to facilitate the interaction identification. The application of the CREWS guidelines searches improving the use cases quality and, therefore, to improve the obtained interactions of the use cases. Different experiments have proven that these guidelines promote the completeness and correctness of the use cases [13]. Lastly, the normalization has to provide the text of use case with information that enables the construction of the Interaction Model. For this purpose, the constituents of each sentence and yours grouping in phrases are identified.

The specification of the Use Case Linguistic Model focuses on three orthogonal, yet complementary, aspects [2,3,14]: the *conceptual*, the *syntactic*, and the *semantic*. Each of these is briefly described below.

- The *conceptual aspect* determines what the meaning of *use case* in Metamorphosis is. This conceptualisation is independent of the way in which it is represented. The specification of a use case shows the complete and organised sequence of actions that the system must execute, including all their possible variations, when interacting with the actors. An *action* may express: a *communication* established between the actor and the system for the exchange of information; or a *behaviour* of the system, in response to the communication established with the actor which is expressed as it is perceived from the outside. The *special* actions enable the condition, the addition, or the repetition of action groups.

- In the *syntactic aspect* a use case is viewed as a text composed by a set of *sentences* that are grouped in parts or sections. Each section can be composed by other sections. The sentences have grammatical information about its syntactic structure and its constituents. Each syntactic structure has a head that is its principal constituent (i.e., the preposition is the head of a prepositional phrase). The sentences can be simple or special. A *simple sentence* describes a communication or behaviour action. It has a single subject and single main verb. It must be written in a declarative, affirmative and active form, in accordance with the CREWS style guidelines [12,14]. The grammatical function of the subject may only be performed by the noun phrase that designates the actor or the system under development. The predicate describes what is said about them in the sentence. It is an expression (that may or may not subordinate) which shows the action and consequences of the verb. The main verb of the sentence must be transitive, guaranteeing the presence of a direct object in the sentence. In Metamorphosis, the CREWS content guidelines have been redefined in order to give to these guidelines more flexibility. The sentence structures proposed by CREWS were combined to form complex structures. The simplified format of these complex structures is:

  `<subject><main-verb>{<object>{<preposi-phrase>[connect]}*}+|<subordi-clause>`

  The *Special sentences* are distinguished by having a predefined structure that makes use of key words (i.e., `INCLUDE`, `EXTEND` and `REPEAT`). Internally, each sentence is described as a set of words to which a category can be associated. These words can also form groups, according to the grammatical function they fulfil in the sentence and the phrase structure they present.
- In the *semantic aspect*, an action is expressed as a relationship established between the verb and one or more semantic or thematic roles. A semantic role denotes an abstract function performed by an element participating in an action. Some semantic roles used by Metamorphosis are: agent, object, destination, owner, owned, cooperator, state, location, causer and time [3,15,16]. Although the CREWS style and content guidelines reduce the syntactic possibilities, a single semantic role can fulfil more than one grammatical function. Metamorphosis uses semantic roles to make the elements of an action independent from the syntactical forms they may take. The semantic roles allow specifying the transformations at a high level of abstraction, without depending on the grammatical diversity of a particular language.

### 2.2 Interaction Model

Metamorphosis assumes the definition of interaction given by the UML [11]. An interaction describes the exchange of messages between two or more instances. Sequence diagrams are used to represent these interactions. However, they can also be expressed by means of other types of UML interaction diagrams (communication diagrams, or high level interaction diagrams). The modelling concepts of the UML Interaction Package were extended to define particular interaction structures. The Interaction Pattern Metamodel is a UML profile that allows the specification of interaction fragments by roles to identify its elements.

In UML terms, a fragment is an interaction that is part of another interaction [11]. Metamorphosis distinguishes both modelling elements as follows. The fragment is

deduced from a use case sentence. It may consist of one or more messages and of one or more instances that send and receive these messages. A complete interaction is the result of combining all the fragments obtained from the sentences of a use case. Graphically, the complete interaction corresponds to a sequence diagram, while a piece of this diagram represents a fragment of the interaction.

The result of the transformation of a sentence is a fragment as part of the complete interaction that describes the use case to which the sentence belongs. This fragment responds to an interaction pattern. An interaction pattern is a generic structure that describes a particular form of interaction fragment. To specify an interaction pattern in Metamorphosis, a technique based on the specialisation of UML metamodels has been applied [17]. This technique uses roles to describe the participants of an interaction pattern. A semantic role specifies the structural and semantic properties that a modelled element of Interaction Profile must have to form part of an interaction pattern. A role is a subtype of a base class of the Interaction Profile.

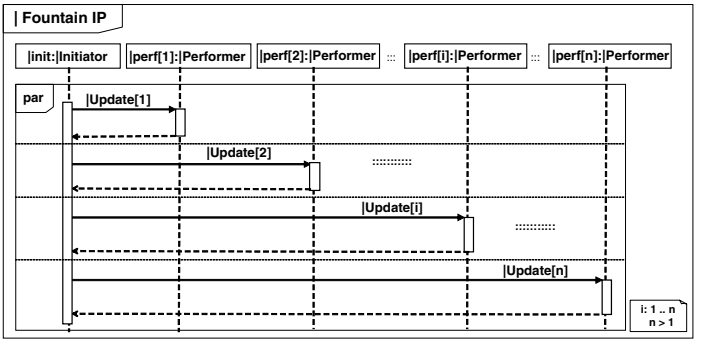



**Fig. 2.** Graphic Representation of an Interaction Pattern

Figures 2 and 3 present a part of the specification of the `Fountain Interaction Pattern (IP)`. Its graphic representation is presented in Figure 2. The expression `|init:|Initiator` indicates that the lifeline role `init` is played by an instance of an `Initiator` class. Similarly, the lifeline role `perf[i]` is played by the `ith Performer` instance in the set of performer classes. In the interaction fragment, each message is sent by an `Initiator` instance to a `Performer` instance. This message activates the `Update` operation that changes each `Performer` instance state that receives the message. The UML interaction operator `Par` indicates that, for each defined coregion [11]: (i) the messages can be sent by `Initiator` instance at the same time; and (ii) the received messages by each `Performer` instance can be executed at the same time.

A partial view of a specialized Interaction Pattern Metamodel is displayed in Figure 3. The `perf` is a pattern lifeline that represents an entity class in the system domain[1]. `init` is a pattern lifeline that represents a border class or a control class. `Update` is the only type of message in the `Fountain Interaction Pattern`. It appears in the pattern at least twice and can have one ore more parameters.

[1] The definitions of "border class ", "entity class" and "control class" correspond to the ones given in [5].

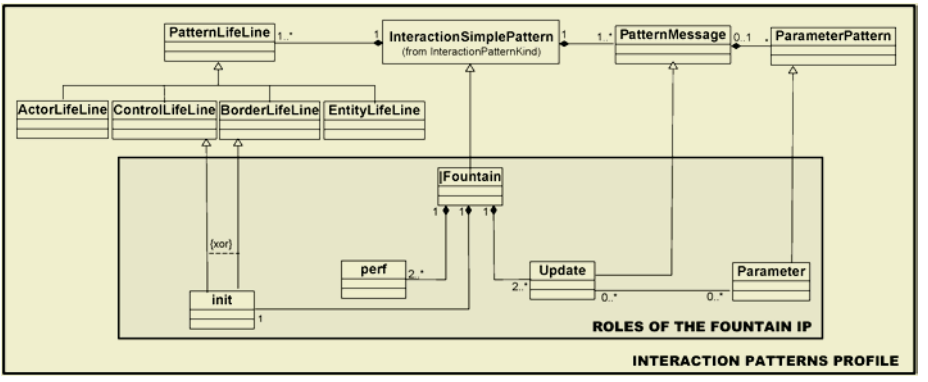



**Fig. 3.** Specialized Interaction Pattern with Semantic Roles (Partial View)

### 2.3 Transformation Levels

In Metamorphosis, to obtain the corresponding Interaction Model the transformation of the Use Case Linguistic Model is performed in three different levels: the one defined by the system, the one delimited by one of its use cases, and the one established by each text sentence of a use case. In the *system level*, the main purpose of the transformation is to integrate the interaction models that are obtained for each use case, until the Interaction Model of the system is completed, thereby guaranteeing its consistency and completeness. This means that the transformation must resolve the possible conflicts generated by this integration so that all partial representations are contained or expressed in the Interaction Model.

In the *use case level*, the transformation fulfils a similar purpose but this time, it endeavours to integrate and complement the information deduced from the use case sentences. The result of applying a transformation to a use case is the corresponding interaction. The transformation is responsible for integrating the information obtained from each sentence of the use case, in a consistent and complete way. Furthermore, it must incorporate into the interaction the information that can only be deduced through a joint analysis of the use case or of groups of sentences, for example: to construct all the sequence diagrams related to the use case (a sequence diagram for the basic path and one for each of its alternative paths); and to combine the interaction fragments that are deduced from each sentence of a use case until the corresponding sequence diagram is formed.

Finally, the transformation in a use case *sentence level* is responsible for identifying the basic elements of an interaction. The main objective of the sentence transformation is to obtain the interaction fragment that describes a sentence. It is based on the application of patterns. This is explained in next section.

## 3 Sentence Transformation Patterns

In Metamorphosis, the transformation patterns are used to describe how a sentence of a use case can be transformed into an interaction fragment. At the moment, it has been defined 21 transformation patterns of sentences for Metamorphosis. Each pattern is

specified using a basic schema of six elements [18]: (i) the *name* identifies the transformation pattern and it distinguishes from others; (ii) in the *description* gives a concise explanation about the pattern; (iii) the *interaction pattern* describes the interaction fragment that corresponding to the sentence (see Section 2.2); (iv) the *observations* shows particular cases of the pattern, application examples and/or several annotations. The following subsections describe the remaining elements of a sentence transformation pattern: (v) *context semantic* and (vi) *transformation rules*.

### 3.1 Semantic Context

A semantic context is a frame that describes the features of a sentence by means of semantic roles [19]. This frame only describes sentences allowed in a use case (see Section 2.1). A semantic context can be seen as a pattern too because is a general specification of a sentence type that can be reused when a use case sentence meets the characteristics of a frame. The main objective of a semantic context is to identify the semantic roles that are presents in a sentence. It is independent of the syntactic structure of the sentence. A semantic context is an application domain element of a sentence transformation pattern.

In Metamorphosis, a semantic context `SC` is specified using a first order logic formula $SC=<\alpha, \mu, \psi>$, where $\alpha$ and $\mu$ are, respectively, sets of variables and constants, and $\psi$ is a set of functions applicable to `SC` terms. These functions determine what semantic roles participate in the sentence, its properties and the relationship types with the principal verb. If a formula is fulfilled, then the transformation rules can be applied to the corresponding semantic context. A simplified version of the `Distributed Action by the Left SC` is specified in Figure 4. This context describes, in CREWS terms, a sentence composed by two or more internal atomic actions [3]. These clauses are separated using the copulative conjunction '`and`' or by commas. The clauses number of a sentence is calculated by the function `NumberClause`. The `agent` and the `main-verb` are distributed to each clause. Thus, the sentence can be decomposed on '`n`' sentences with the same `agent` (active entity that initiates or controls the action) and `main-verb`.

**Distributed Action by the Left SC**

$\forall V,A,O,St,L,D,Or,Od:$
$\exists n>1$ / NumberClause(Sentence)=n $\wedge$
$\forall i=1..n$ ( $Action_i$(Verb:V,Agent:A,$Object_i$:O) $\wedge$
( ($State_i$(Object$_i$:O,State$_i$:St) $\vee$ $State_i$(Object$_i$:O,State$_i$:?)) ) $\wedge$
$Ownership_i$(Owned$_i$:Object$_i$,Owner$_i$:Od) ) ;

**Fig. 4.** Semantic Context: An Example (Simplified Version)

In each clause can be identified a passive entity on which such action falls (the `object`). The status change undergone by the each `object`, as a consequence of the action performed, may or may not have been explicitly expressed in the sentence (`State` function). In addition, each `object` must have an `owner` (entity that possess to the `object`). Finally, if a simple sentence satisfies the conditions described above, it can be inferred that its semantic context is the one described by the `Distributed`

Action by the Left SC (Fig. 5). It is advisable to point out that this sentence type is not included in the CREWS content guidelines. The context semantic describes a sentence type as a composition of CREWS sentences. The added complexity can be controlled at pattern level in order to avoid analysis errors.

| S:'The System registers the buyer personal details, the salesman commission and the purchase deadline' |
|---|
| NumberClause(S)=3 ∧ |
| $Action_1$(Verb:'registers',Agent:'The System'; $Object_1$:'the personal details') ∧ |
| $State_1$($Object_1$:'the personal details';$State_1$:?) ∧ |
| $Ownership_1$ ($Owned_1$:'the personal details';$Owner_1$:'buyer') ∧ |
| $Action_2$(Verb:'registers',Agent:'The System'; $Object_2$:'the commission') ∧ |
| $State_2$($Object_2$:'the commission';$State_2$:?) ∧ |
| $Ownership_2$($Owned_2$:'the commission';$Owner_2$:'salesman') ∧ |
| $Action_3$(Verb:'registers',Agent:'The System'; $Object_3$:'the deadline') ∧ |
| $State_3$($Object_3$:'the deadline';$State_3$:?) ∧ |
| $Ownership_3$ ($Owned_3$:'the deadline';$Owner_3$:'purchase'); |

**Fig. 5.** Evaluation of the Semantic Context of a Sentence

### 3.2 Transformation Rules

The participants of semantic context can be turned into interaction fragment elements by means of transformation rules. These rules are described with a formula: its left side corresponds to interaction pattern elements and, its right side, indicates how to identify these elements. The specification of a transformation rule contains semantic or thematic roles and functions applied on these roles.

Figure 6 presents the transformation rules applicable to sentences whose semantic context is described by a Distributed Action by Left SC (Fig. 4). When these rules are applied, an interaction with the generic form of the Fountain IP is obtained (Fig. 2). The first rule establishes that the Initiator instance is identified as the Agent of the sentence. Each Performer instance is deduced of the Owner contained in each clause of the use case sentence. The head function extracts the most important constituent of the Owner. The normalization function allows obtaining the canonical form of the role heads. The signature of the Update operation is deduced by a Sequence function. This function constructs a label with the principal Verb, the Object and the Object State of each sentence clause.

| FountainIP ← Distributed Action by the Left SC | | |
|---|---|---|
| \|init:\|Initiator | ← | Agent |
| \|perf[i]:\|performer | ← | <Head($Owner_1$)>NORM ∀i=1..n |
| \|Update[i] | ← | Sequence(Verb,<$Object_1$> NORM,State) ∨<br>Sequence(Verb,<$Object_1$> NORM) ∀i=1..n |

**Fig. 6.** Transformation Rule: An Example

To transform a sentence its semantic context has to be recognized. The sentence semantic context is compared with the semantic context of each transformation pattern. The next step is to identify the rules defined by the transformation pattern. Fi-

nally, these rules are applied to obtain the corresponding interaction fragment. To do this, the syntactic information must be used to concrete the application of a transformation pattern. For example, we consider the sentence in Figure 5 which satisfies the `Distributed Action by Left SC`. The pattern that corresponds to this semantic context has the transformation rules given in Figure 4. Its interaction pattern is explained in Section 2.2. In addition, we assume the syntactic structure that is linked to the sentence roles.

According to the first transformation rule (Fig. 6), the name given to the border instance (`Initiator`) is extracted from the `Agent (system)`. This instance can be fulfilled by a control instance too. By default, the name of the system that is being developed is a border instance (Fig. 7). Three domain instances can be recognized from the `noun-phrase (head)` content in each `of-prepositional-phrase (Owner)`. The canonical form of these noun phrases must be considered. These instances are: `buyer`, `salesman` and `purchase`. Three synchronous and parallel sent messages by `system` instance are received by these instances. Each message is responsible for activating an updating operation named `registers` in each domain instance. The registered information in each receipted instance is deduced of the `object (noun-phrase)` of each clause. The name of an instance is normalised to obtain its canonical form.

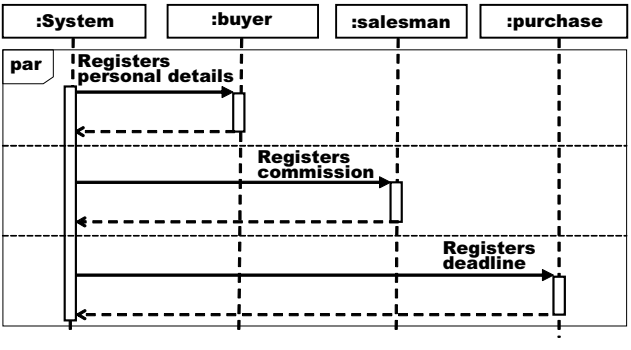



**Fig. 7.** Interaction Fragment obtained by Metamorphosis

## 4 Pattern Validation

In principle, the transformation patterns defined by Metamorphosis were designed through the direct observation of a sample of sequence diagrams obtained from the use cases of some academic and commercial information systems. To validate these patterns was applied a strategy in order to establish the limitations of the transformation patterns designed initially and then to improve and enrich them [20]. A transformer was developed to automatically validate the patterns. This tool was integrated into OO-Method, an automatic software production environment which supports requirements specification and analysis [6,21].

---

[1] Hereafter, we will use the terms "instance" and "object" interchangeably. The definitions of "border class object" and "entity class object" correspond to the ones given in [8].

To make the manual validation of the transformation patterns proposed, the Use Case Model of the Car Rental System (CRS) was developed. After normalization of the use cases, the Sequence Diagram Model was constructed. Both models were exhaustively revised by stakeholders in order to reach a consensus on the results obtained manually. The sequence diagrams were then compared with the sequence diagrams generated using the transformer in order to determine differences and similarities. Forty-one use cases were analyzed. This included a total of 574 steps, of which only 14% were special (conditionals, iterations, etc.).

The interactions manually obtained were compared with the interactions generated automatically for each step of the CRS use cases. The comparison had to establish whether automatically generated interactions were the ones expected by stakeholders. To do this, it was necessary to determinate when both interactions were equal, equivalent or different. We considered them to be equal when they were compounded by the same instances and the messages that these instances exchanged. We considered two interactions to be equivalents if both represented the same interaction goal even though the instances and messages were not the same. If the interactions were neither equal nor equivalent, we considered them to be different. Using these criteria, 66% of the transformation patterns were equals, 23% were equivalents and only 11% were categorized as different. This experience allowed us to establish which of the transformation patterns had to be improved or rejected. It was also possible to identify new transformation patterns of the semantic contexts.

## 5 Related Works

In the last three decades, many approaches based on natural language processing to develop information systems have been reported [7,22]. Of particular value to this paper are those that make use of the linguistic properties of a text to obtain information to construct models automatically [8,23]. The most useful are those proposals that establish relationships between a functional requirements specification and the elements of a interaction [9,24]. In this context, the works of Feijs and Li [25,26] can be considered representative because these works were specially conceived to transform use cases into interactions. The study presented by Feijs establishes correspondences between certain types of sentences, written in natural language, and MSCs (Message Sequence Charts). On the other hand, Li proposes a semiautomatic process to derive sequence diagrams from a limited subset of sentences. The Li and Feijs contributions are important to our approach because have similar objectives. However, the transformation processes has not a vision of Software Engineering and its specification does not promote the genericity, reusability and traceability.

## 6 Conclusions

This paper describes how Metamorphosis transforms use case sentences into interaction fragments. This transformation process requires that each use case has been previously normalised. This means that the use case fulfils the CREWS style and content guidelines. Also, the normalization adds syntactic information to each use case sen-

tence. In Metamorphosis, the content guidelines were expanded allowing to specify use cases using complex sentences. These complex sentences were obtained by composition of the CREWS basic syntactic structures. This was possible without ambiguity risk due to the application of transformation patterns. These patterns define an only possible way of interpreting sentences.

The sentence transformation establishes correspondences between a semantic context (that is described by semantic roles) and a fragment (that is specified by an interaction pattern). When the transformation is carried out an interaction fragment is obtained by each sentence. The convenient combination of these fragments, deduced from all the sentences of a use case, results in a complete sequence diagram. The sentence transformation strategy uses roles which facilitate the identification of the elements to be transformed without required to know its grammatical structure. Thus, they guarantee that the transformation rules do not depending on the language used to write the use cases. As example, a transformation pattern was described and applied to a sentence. Lastly, this paper briefly describes an experiment designed to validate the transformation patterns. Although the results can be considered to be positive, more replicas of this experiment should be performed.

## Acknowledgments

This work has been developed with the support of: (i) the project E-Services Development (MEC, Spain TIN2004-03534); (ii) the project CICYT TIC2003-07158-C04-03; and (iii) the Council of Humanistic and Scientific Development of the Central University of Venezuela (CDCH/UCV).

## References

1. Van Lamsweerde A.: Requirements Engineering in the Year 2000: A Research Perspective. In Proceedings of the 22nd Conference on Software Engineering (ICSE 2000). Pp. 5-19. ACM Press.
2. Cockburn A. Writing Effective Use Cases. Addison-Wesley. 2001.
3. Rolland C., Ben-Achour C.: Guiding the Construction of Textual Use Case Specifications. Data & Knowledge Engineering 25(1998), 125-160. Elsevier Science.
4. Evans G.: Getting from Use Cases to Code. Part 1: Use-Case Analysis. The Rational Edge. July 2004. http://www-106.ibm.com/developerworks/rational/library/5383.html
5. Jacobson I., Christerson M., Jonsson P., Övergaard G.: Object-Oriented Software Engineering. A Use Case Driven Approach. Addison-Wesley, 1992.
6. Insfrán E., Pastor O., Wieringa R.: Requirements Engineering-Based Conceptual Modeling. Requirements Engineering, 7(2), 61-72. Springer-Verlag. March 2002
7. Boyd N.: Using Natural Language in Software Development. Journal of Object Oriented Programming-Report on Object Analysis and Design. February 1999.
8. Juristo N., Moreno A., López M. How to Use Linguistic Instruments for Object-Oriented Analysis. IEEE Software Vol. 17 Issue 3. May/June 2000. Pp. 80-89.
9. Burg J.F.M., Van de Riet R.P.: Analyzing Informal Requirements Specifications: A First Step towards Conceptual Modeling. In Proceedings of the 2nd International Workshop on Applications of Natural Language to Information Systems. The Netherlands, 1996.
10. Object Management Group: MDA Guide. Version 1.01. Jun 03. http://www.omg.org/uml

11. Object Management Group: Unified Modeling Language: Superstructure Specification. Version 2.0. August 2003. http://www.omg.org/uml.
12. Ben Achour C.: Guiding Use Case Authoring. In Proceedings of the 8th European-Japanese Conference on Information Modelling and Knowledge Bases. Ed. P. Chen and R.P. van de Riet. Pp. 181-200. Finland. May 1998.
13. Ben Achour C., Rolland C., Maiden N.A.M., Souveyet C. Guiding Use Case Authoring: Results of an Empirical Study. In Proceedings of the Fourth IEEE International Symposium on Requirements Engineering (RE'99). Pp. 36-43. Ireland, June 1999.
14. Diaz I., Losavio F., Matteo A., Pastor O.: A Specification Pattern for Use Cases. Information & Management, Vol. 41/8 (2004). Pp. 961-975. Elsevier Science B.V.
15. Fillmore Ch.: The Case for Case. In Universals in Linguistic Theory, ed. By Bach & Harms. New York: Holt, Rinehart & Winston. 1968.
16. Moreda P., Palomar M., Suárez A.: Assignment of Semantic Roles based on Word Sense Disambiguation. In Proceedings of the IX Ibero-American Conference on Artificial Intelligence (IBERAMIA), Puebla, México. 2004.
17. France R., Kim D., Ghosh S., Song E.: A UML-Based Pattern Specification Technique. IEEE Transactions on Software Engineering, Vol. 30, N÷ 3, March 2004. Pp. 193- 206.
18. Gamma E., Helm R., Johnson R., Vlissides J.: Design Patterns. Elements of Reusable Object-Oriented Software. In Professional Computing Series, Addison-Wesley, 1992.
19. Guildea D., Jurafsky, D.: Automatic Labeling of Semantic Roles. Computational Linguistics 28(3): 245-280, 2002.
20. Díaz I., Moreno L., Fuentes I., Pastor O.: Integrating Natural Language Techniques in OO-Method. In Proceedings of the Sixth International Conference on Intelligent Text Processing and Computational Linguistics (CICLing 2005). LNCS, Springer-Verlag. méxico. February, 2005.
21. Pastor O., Gómez J., Insfrán E., Pelechano V.: The OO-Method Approach for Information Systems Modeling: from Object-Oriented Conceptual Modeling to Automated Programming. Information Systems 26 (2001): 507-534.
22. Métais E.: Enhancing IS Management with Natural Language Processing Techniques. Data & Knowledge Engineering. 41(2002), 247-272. Elsevier Science B.V.
23. Burg J.F.M., Van de Riet R.P.: Color-X: Linguistically-based Event Modelling: A General Approach to Dynamic Modeling. Proceedings of the 7th International Conference on Advanced Information Systems Engineering. LNCS. Springer-Verlag, 1995.
24. Flield G., Kop C., Mayerthaler W., Mayr H., Winkler C.: Linguistic Aspects of Dynamics in Requirements Specifications. In IEEE Proceedings of the 11th International Workshop on Databases and Expert Systems Applications (DEXA'00). 2000. Pp. 83-90.
25. Feijs L.M.G. Natural Language and Message Sequence Chart Representation of Use Cases. Information and Software Technology 42 (2000). Pp. 633-647.
26. Li L. Translating Use Cases to Sequence Diagrams. In Proceedings of the Fifteenth IEEE International Conference on Automated Software Engineering (ASE'00). Pp. 293-296.

# Query Refinement Through Lexical Clustering of Scientific Textual Databases

Eric SanJuan

LITA Université Paul Verlaine & URI-INIST/CNRS
Metz - France
eric.sanjuan@iut.univ-metz.fr

**Abstract.** TermWatch system automatically extracts multi word terms from scientific texts based on morphological analysis and relates them through linguistic variations. The resulting terminological network is clustered based on a 3-level hierarchical graph algorithm and mapped onto a 2D space. Clusters are automatically labeled based on variation activity. After a precise review of the methodology, this paper evaluates in the context of querying a scientific textual database, the overlap of terms and cluster labels with the keywords selected by human indexers as well as the set of possible queries based on the clustering output. The results show that linguistic variation paradigm is a robust way of automatically extracting and structuring a user comprehensive terminological resource for query refinement.

## 1 Introduction

The need to automatically identify research topics has become acute with the availability of huge stocks of specialized texts on the Internet. Be it for web-based information retrieval [37], topic and event detection [33], domain knowledge mapping [30] or text mining [12], the necessity to dispose of a condensed view of the important topics and their layout remains an important goal for research in these areas. However, to the best of our knowledge, existing clustering methods have relied on a basic principle to cluster similar units : the assumption that co-occurrence of two units (words, keywords, terms, phrases, authors, references) in one document is proof that the two units are similar and thus should end up in the same cluster. Statistical data analysis methods like Latent semantic analysis (LSA), hierarchical clustering and K-means start from a co-occurrence matrix of the index terms in the documents, from which they generate similar clusters. Usually, a co-occurrence threshold is set which eliminates in some cases up to 80% of the initial dataset from further analysis. The relevance of the co-occurrence assumption has rarely been questioned. In [18], Ibekwe - SanJuan argued that, in clustering texts units - words, phrases and terms, other relations hold between them (linguistic in nature) which should be sounded out as an alternative clustering principle. This is the hypothesis upon which the CPCL (Classification by Preferential Clustered Links) is based and implemented in the TermWatch system [19]. The idea is that if in a corpus, we find terms[1] related via

[1] To be understood here in their terminological sense, i.e., denominations of concepts and objects in a specialized field. So terms here cannot be reduced to words or word sequences.

A. Montoyo et al. (Eds.): NLDB 2005, LNCS 3513, pp. 251–262, 2005.
© Springer-Verlag Berlin Heidelberg 2005

some linguistic operations, it could be relevant to use these relations instead of co-occurrence as the clustering criteria. For example, the CPCL algorithm formed the following small cluster from a corpus of abstracts dealing with Information Retrieval (IR) used in this experiment: "*object software, object-oriented software, object-oriented software testing, object oriented testing, software testing*". No assumption is made on the co-occurrence of these terms in the same texts and no frequency information is used to obtain the cluster. The cluster is obtained through lexical operations. Basically, we consider lexical similarity as an indicator of semantic proximity, i.e., if $term_1$ is a substring of $term_2$, then there is a probability that the two terms share some semantic relation, usually hypernymy/hyponymy. In the above example, we observe how the insertion of a new modifier word specializes the concept of "*object software*" to "*object-oriented software*". In the same vein, the addition of a new head word shifts the topical emphasis from "*object-oriented software"* to "*object-oriented software testing" w*here the focus is now on *"testing"*. The two base terms *"object software"* and *"software testing"* are the primary connectors between the longer variants of the terms. These lexical operations involving insertions, additions and substitutions of new modifier or head words are called "variations" and have been extensively studied by computational terminologists [20, 10, 18]. Clustering text units based on lexical similarity and associations (we consider also other operations apart from subsumption) ensures that rare occurring units as well as highly occurring ones are given the same importance. This could be important for applications like science and technology watch where the focus is on new or unknown information characterized often by low frequency.

Other studies in the computational linguistics and terminology fields have employed lexical similarity as criteria for automatically identifying semantic relations between terms for applications like automatic thesaurus construction or ontologies [26, 27, 14]. However, these studies were interested in binary relations between pairs of terms which is what their applications required. No clustering is performed.

Clustering has been extensively used in IR relies to reduce the number of returned documents which the user has to sift through. The basic idea is that clustering similar documents into small groups will enhance information retrieval in general and query expansion in particular (see [2, 16] for comprehensive reviews). [37] applied a clustering method to snippets of ranked pages returned by a search engine. Since no linguistic definition of the text unit clustered is given in this study, texts are treated as word sequences[2]. This can lead either to isolating component words of a domain concept, as domain concepts are often formed by complex noun phrases (NPs), or to extracting invalid syntactic units on the border of two syntactic structures, NPs and verb phrases (VPs) for instance. Although phrases have been explored for automatic indexing [5], they are yet to be fully explored for text clustering.

Our clustering approach forms clusters depicting the research topics contained in the text collection and maps these clusters onto a 2D space where their layout can be

[2] The authors consider a phrase as "*an ordered sequence of one or more words*". This is the typical "bag-of-word" approach. In addition, if only "one word" can be a phrase, then the ordering aspect becomes null.

interpreted. This graphic aspect will not be developed in this paper as it has been presented elsewhere [19]. In the context of query refinement, we will focus here on the comparison of extracted terms and clusters labels with a reference list of keywords used by human specialists to index the abstracts in the corpus, and on the evaluation of the document accessibility through the clustering results.

After an overview of our methodology illustrated by its application on a collection of IR texts (§2), we compare the keywords selected by human indexers with the multi word terms extracted automatically in (§3) and with the cluster labels formed by TermWatch (§4). This is done by using lexical inclusion. In (§4) we also evaluate the usefulness of our cluster and component labels for query refinement process.

## 2 Methodology Overview

The lexical similarity approach underlying the TermWatch system symbolizes the meet between computational linguistics, especially computational terminology, and data analysis techniques. It has two major components, an NLP component which performs morphological analysis in order to extract terms and relates them through variations, and a clustering component. The results are mapped onto a 2D space via an information visualization package (Aisee[3]). Fig 1. summarizes the system architecture. External resources are underlined in this figure.

The corpus used in this experiment is an IR corpus of 3 355 scientific texts (titles and author abstracts of documents) published between 1997- 2003 in 16 prominent journals. It is made up of 454 412 words of which 20 928 unique words[4].

### 2.1 NLP Component

The term extraction and variant identification component builds on major research in computational terminology, notably on studies done by [20] and [10]. Our approach to term extraction is based on shallow parsing and selective NLP. The only linguistic resource needed is a dictionary like [31] and a tagger which enables each word to receive a morphological tag. In this experiment, we used the LTChunker together with the LTPOS part-of-speech tagger developed by Mikheev (1996)[5] and WordNet [13] to normalize the terms. After tagging by LTPOS, our pattern matching rules are then run on these tags in order to extract likely terminological units. The rules utilize the morpho-syntactic properties of terms known to comprise mainly noun phrases with possible prepositional attachments, as in NNN_prep_N ("*information seeking behaviour of engineers*") or simply a compound structure A/N_NN ("*new improvement algorithm*"). Because we do not filter the candidate terms with statistical indices, our term extraction module tends to produce large sizes of candidate terms. We

[3] www.aisee.com

[4] The IR corpus was extracted from the PASCAL database, a scientific bibliographic database with over 14 million records from hundreds of journals. The corpus was obtained curtesy of the INIST/CNRS, the French research center on scientific and technical information.

[5] http://www.ltg.ed.ac.uk/software/chunk/index.html

then filter them using the variation operations as only terms with variants will be considered for further processing.

The variation identification module identifies various types of lexico-syntactic operations formally described in [18] amongst the terms which enables us to output the graph of term variants. The lexico-syntactic variations considered involve term length modification as in "*information theory --> general information theory*" and/or structural modification as in "*information system --> management of information system*" whereby a new head word "*management"* accompanies a structural transformation (passage from a compound structure to a prepositional one). Such operations are called expansions because they relate terms in which one is a subpart of another. Some variations simply involve word change in a given position between terms of equal length as in "*information seeking*" and "*information retrieval*". These are called substitutions.

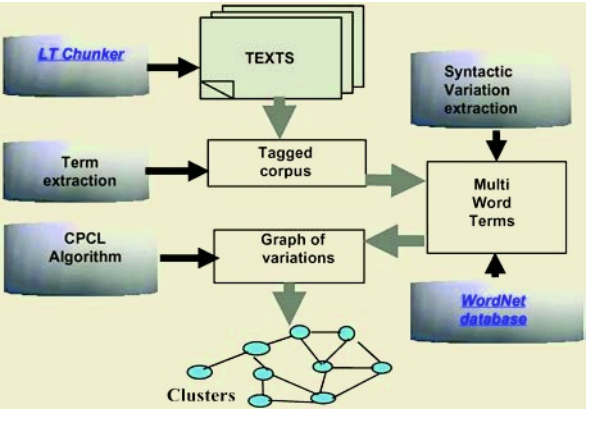



**Fig. 1.** TermWatch architecture: modules used in the present experiment.

In this experiment 50 737 candidate terms were extracted. 41 058 terms were involved in variation relations. Substitutions were filtered using two criteria. First those for which the substituted words are not in the same morpho-syntactic category were rejected. From the remaining substitutions, using the external semantic resource WordNet, we applied the 'jcn' similarity measure [28] implemented in the WordNet::similarity perl module[6] to further restrict substitutions to those involving similar nouns (jcn value >1). We shall denote by 'strong-Sub' the set of these substitutions. It contains 1 220 pairs involving 1 992 distinct terms. Among the remaining substitutions, only those involving terms with at least three words were retained, thus forming a set of 7 636 pairs involving 5 441 distinct terms. We shall call this set 'weak-Sub'.

Likewise, expansions were divided into two categories depending on if they related terms with the same head (left-Exp) or not (right-Exp). We found 4 333 pairs in

[6] Copyright (C) 2003-2004 Siddharth Patwardhan, Ted Pedersen, Satanjeev Banerjee, and Jason Michelizzi.

left-Exp involving 6 241 terms and 2 903 pairs in right-Exp involving 4 110 terms. Finally, we found 1 974 pairs of insertions (where modifiers are inserted within a term) involving 3 276 terms.

### 2.2 CPCL Clustering Algorithm

There exists in the literature different ways of classifying clustering algorithms. One such way is to distinguish between model-based versus heuristic-based algorithms [34]. We restrict ourselves to model-based algorithms. Indeed tracking terminology evolution implies a comparative analysis of different clustering outputs that have to be perfectly defined, this excludes heuristic-based algorithms.

The main mathematical models are probabilistic for expectation/maximization-like (EM) algorithms that generalize the k-means algorithm, and the residuation model for hierarchical algorithms. Both models enable the definition of clustering algorithms that do not require specification of the number of clusters in their input and whose output is unique (i.e., independent from any random choice of the starting point). Most of the probabilistic models are based on Gaussian and Bernouilli mixtures as explained in [7] for continuous and binary data respectively. It has been also shown in [17] that Latent Semantic Analysis (LSA) can be advantageously redefined in mixture models (Probabilistic Latent Semantic Analysis, PLSA). However, the main difficulty in applying these algorithms on large and sparse data is that their convergence speed depends on the choice of the starting points. The farther this starting point, the slower the convergence speed.

Graph theoretic algorithms are better suited to the nature of our data. Recent graph algorithms have been introduced in bio-informatics. In [15], an algorithm extracting high density subsets of vertices is proposed. However, density functions cannot be defined on our sparse graphs. In [1], a Cluster Affinity Search Technique (CAST) is proposed as a heuristic-based method that efficiently converges on gene expression data but not in the general case.

The graph theoretical algorithm that has the strongest mathematical properties is clearly the Single Link Clustering (SLC) but it is also well known that SLC has a major drawback called the chain effect. This generates very long clusters and thus makes SLC only adapted for special datasets where the desired clusters have this property. However, residuation theory [4] allows to show that the set of hierarchical clustering outputs on a data set is a lattice known as the lattice of ultrametrics as recalled in [23]. The mathematical properties of SLC comes from the fact that the output of this algorithm can be characterized in this lattice. This gives an alternative approach to model-based clustering algorithms.

CPCL is a hierarchical two-step extractor of clusters from a graph of term variants. One notable attribute of this algorithm is that the clustering begins not at the atomic level (term level), but at the component level. Components are obtained by grouping terms that share a subset of variation relation (called COMP relations) that mainly affect the modifiers in terms. The clustering stage then consists in merging iteratively components that share several variations of the head type (called CLAS relations). A normalized coefficient is used to indicate the proximity between two components as a

function of the number of CLAS relations between them and the proportion of the particular CLAS relation in the graph.

This way of reducing the graph $L_{COMP}$ step by step leads to a hierarchical clustering algorithm of $L_{COMP}$. We start from the graph $L_{COMP}$ and the index $d$, and we compute a sequence of reduced graphs and corresponding indices.

Based on this theoretical framework and on experiments, it has been shown in [3] that CPCL output is unique like in SLC while avoiding the chain effect. Indeed, on a graph of 300 components, to cluster 192 elements, SLC algorithm needed 12 iterations and merged 125 units in a sole cluster while CPCL algorithm needed only 2 iterations and its clusters had at most 12 elements. Moreover it has been shown that each iteration can be computed in linear time on sparse data set. Indeed, let $E$ be the number of edges of the graph and $d$ the maximal degree of a vertex, then the time complexity of one iteration is bounded by: $O(d.E)$.

The last problem to solve in a clustering approach is how to select the labels of the clusters. Our solution is to choose the term with the highest number of variations in CLAS, since this is the term that should better reflect the position of the cluster *vis-à-vis* the other clusters in the final network.

In the application to the IR corpus, we selected 'strong-Sub' and 'left-Exp' as COMP variations, the remaining ones : weak-Sub, right-Exp and insertions being considered as CLAS variations. We stopped at the second phase of the clustering algorithm, thus at the first iteration. The system computed 289 classes comprising 628 components and 2 614 terms. The biggest cluster labeled by "*Information retrieval*" had 142 terms, among which 84 were in the same component labelled : "*Information retrieval system*". Overall, only 9 clusters had more than 30 terms.

The image of the whole network of classes is too huge to be visualized without an interactive graph displayer. However, different subnetworks can be highlighted. A central network formed around "*information retrieval"* appears to be strongly related to *"natural language"* and *"public library"*. We emphasize that these relations only rely on the existence of multiple paths between these terms in the graph of variations and not on co-occurrence information. Among the peripheral networks, one appears to be formed around the term *Rough Set*. When unfolding the two related clusters labeled *"rough set"* and *"transaction analysis tool"*, to see the components involved, variation relations between labels became more explicit. Here we found out that the relation between *"rough set"* and *"transaction analysis tool"* comes from the term "*rough set datum analysis*" which appears in an abstract dealing with application of rough set theory to web log files. Naturally, the cluster *Rough set* involves components formed around expansions of "*rough set*" that suggest possible refinement of queries dealing with rough sets.

## 3 Terminology Similarity Test

Since the keywords used to index the abstracts come from a gold standard: the PASCAL controlled lexicon, we seek here to determine the terminology overlap between the whole set of corpus terms extracted by TermWatch (50 737) and the

whole set of thesaurus terms used to index the texts (5 168). Exact terminological overlap was however extremely low. Only 40 exact matches were found (0,5% of the total terminology). All the terms in the intersection were binary terms (made of two words) like "*artificial intelligence, probabilistic model, peer review, language processing, information policy*". Only very generic terms from CPCL clusters matched the equally very generic PASCAL keywords. Allowing for the impact of our term extraction approach on this low performance (we can extract long noun phrase sequences like "*valued propositional truth value logic lvpl")*, we relaxed the matching rule to allow for mutual lexical inclusion, thus for partial match. We then calculated the proportion of corpus terms that are subsumed by keywords or vice versa.

We first look at the ratio of lexically-associated corpus terms to keywords. 17 262 corpus terms (43%) were found to be lexically associated to a keyword. The following table gives the rank of the most frequent corpus terms ('Term_id') that are lexically associated to the keywords. 'KW_id' is the rank of the keyword by frequency.

From this table, we can see that frequency is not correlated with lexical association as only three cases (*information retrieval, information system, wide web*) of very frequent corpus terms and keywords are lexically matched. Most frequent corpus terms are lexically associated to lower frequency keywords like "*digital library*" since this keyword only arrives at the 473rd place. Conversely, the most frequent keyword in this corpus, here "*theory*" (956 occurrences) is not associated with frequent corpus terms. Thus, the two methods will not rank in the topmost position the same terms and disagreements will occur very quickly. We also observe that some frequent terms like *experimental result* ranked at the 9th position are not lexically associated with any keyword.

**Table 1.** The topmost frequent corpus terms lexically-associated to keywords.

| Term_id | KW_id | Term | KW |
|---|---|---|---|
| 1 | 2 | information retrieval | Information retrieval |
| 2 | 4 | information system | Information system |
| 3 | 12 | information science | Information science |
| 4 | 348 | retrieval system | Information retrieval systems |
| 5 | 5 | wide web | World Wide Web |
| 7 | 45 | neural network | Neural networks |
| 8 | 7 | information technology | Information technology |
| 10 | 2 | information retrieval system | Information retrieval |
| 11 | 28 | web site | Web site |
| 13 | 79 | object oriented | Object oriented programming |
| 14 | 473 | digital library | Digital library |
| 15 | 80 | information need | Information need |
| 16 | 60 | information source | Information source |
| 17 | 73 | genetic algorithm | Genetic algorithms |

However, if we look for the distribution of corpus terms which are lexically-associated with 12% (599 keywords) of the most frequent keywords (occurring at least 3 times), we obtain the plot in figure 2. The plot gives the probability for a corpus term to be related to one of the most frequent keywords, knowing the frequency

rank of the term. This plot completes the information given in table 1 and shows that as we move down the frequency rank of corpus terms, the proportion of corpus terms lexically-associated to the most frequent keywords drops significantly. For instance, at the 73rd rank, less than 50% of corpus terms share lexical associations with the most frequent keywords.

Now we look at lexical associations from the point of view of keywords. We found that only 460 keywords are lexically associated to the corpus terms. Figure 3 shows the 13 most frequent keywords and the number of lexically-associated corpus terms (The *x* axis represents the number of corpus terms to which each keyword, *y* axis, is lexically-associated). We observe a more random movement showing that the frequency of a keyword cannot be used to predict lexical association with corpus terms. Thus a very frequent keyword from an external semantic resource (here a controlled vocabulary lexicon) may simply not be lexically-included in the corpus terms. This is the case for "*fuzzy sets*, *comparative study"* and "*mathematical models*". *Fuzzy set* ranked 9th by frequency has been used as a generic keyword to manually index a large set of documents related to fuzziness and containing corpus terms like "*fuzzy number*, *fuzzy rule*, *fuzzy system"*. Whereas, figure 2 showed that when considering highly frequent corpus terms, the proportion that shared lexical association with keywords was high (among the 35 most frequent corpus terms, 80% had lexical associations with keywords), this information is not correlated from the keywords perspective.

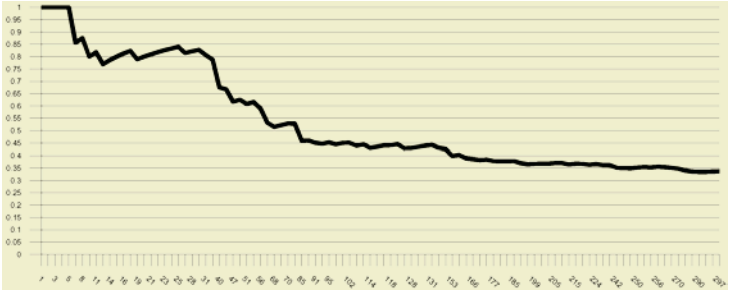


**Fig. 2.** Distribution of the most frequent corpus terms sharing lexical association with 12% of the most frequent keywords.

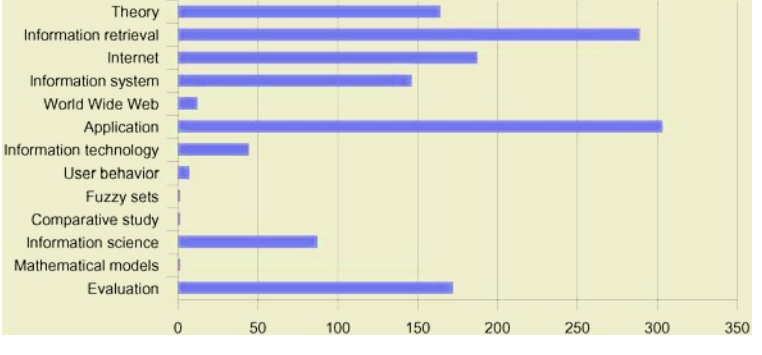



**Fig. 3.** Distribution of the most frequent keywords lexically-associated with corpus terms.

Thus 31% of extracted corpus terms are lexical expansions of only 460 frequent keywords. These 460 keywords are not sufficient to index the whole set of documents in the IR corpus (89% have no keyword among the 460 selected). However, each document has at least one term that is an expansion of one of these keywords. Thus, starting from the 460 keywords it is possible to access the whole set of documents by lexical expansion. This can be useful for assisted query refinement. A similar result could have been obtained using FASTR, the term extractor of Jacquemin C. [20]. This software directly looks for variation of keywords from a thesaurus among texts. The main feature of completely unsupervised systems like TermWatch is to provide support for query refinement not dependent on lexical resources. This is the focus of next section.

## 4 Cluster Evaluation for Query Refinement

To evaluate if the clustering output can help IR users in a query refinement task, we first map the cluster units (terms in clusters), the component labels and the cluster labels onto the set of keywords using left-expansion (a more restrictive criterion than lexical inclusion). Thus, if a term is a left-expansion of some keyword, then it is associated with the longest one.

At the level of terms, we obtain that 1 633 out of 2 614 terms in clusters are left-expansion of some keyword. The number of related keywords being 262. At the component level, 374 labels out of 628 are related to 208 frequent keywords and in the case of cluster labels, 172 out of 289 are related to 132 keywords. Thus the proportion of labels that are related to keywords by left expansion is unchanged (60%) as we move up in the clustering hierarchy. Consequently, 60% of labels proposed by TermWatch can be considered as possible refinements of keyword queries based on Pascal lexicon. This proportion is twice than the one for the general list of extracted terms and shows that terms with a high variation activity are significant expansions of a reference list of keywords.

Now we examine how documents can be accessed only from corpus terms in clusters by query refinement. Table 2 summarizes the results of this evaluation. Only 5% (2 614) of the total number of extracted terms appear in the clustering output. Notwithstanding, this 5% is able to select 60% (1 896) of the documents which have at least one term among them. However, the rate number of index terms per number of documents is too high (1.38). An important result of our experimentation is that this rates significantly drops to 0.35 for clusters labels. Since our clustering process selects terms only based on their variation activity, this tends to show that variation is a relevant criteria for selecting index terms and that our clustering output offers to the user a network of 289 terms that allows him to directly access 816 documents. Then, by exploiting the clusters contents as possible query refinements, the user can progressively access in a three steps approach to the information contained in 60% of the documents. To access the remaining documents, it is necessary to re-iterate the clustering process only on terms in these missing documents. This can be done interactively since the overall clustering process requires less then 39 seconds on a pentium IV PC running on DEBIAN LINUX. Moreover, the fact that the variation relations

we selected are sufficient to involve at least one term in each document, guarantees access to any document in the corpus.

However, taking into consideration only occurrences of terms in documents is not enough to evaluate the effectiveness of such process if too large sets of documents are indexed by the same terms in the clustering output. This is not the case. To prove it, we computed all the logical views of documents using terms in clusters and labels of clusters. These logical views are simply maximal sequences of terms or labels that appear in documents. Table 2 gives their number at each level of the clustering process as well as their number for accessing unique documents, pairs or triplets of non discernible documents. We see that 80% of documents containing terms in the cluster output can be accessed individually or in small sets (less than four documents). Naturally, as we move up in the hierarchy (towards clusters), the labels become more generic and this figure drops to 37%, meanwhile the rate between the number of terms or labels over the number of logical views increases up to 94% for cluster labels.

**Table 2.** Statistics on accessible documents via the clustering output.

| | **Terms in clusters** | **Component labels** | **Cluster labels** |
|---|---|---|---|
| **Number of terms** | *2614* | *628* | *289* |
| **Total number of accessible documents** | *1896* | *1119* | *816* |
| **Rate terms / accessible documents** | *1,38* | *0,56* | *0,35* |
| **Number of logical views** | *1402* | *520* | *272* |
| **Accessible documents individually** | *1232* | *398* | *195* |
| **Number of accessible documents in pairs** | *186* | *96* | *52* |
| **Number of accessible documents per three** | *93* | *84* | *54* |
| **Rate accessible documents in small sets** | *80%* | *52%* | *37%* |
| **Rate logical views /number of terms** | *54%* | *83%* | *94%* |

## 5 Conclusion

Variations are a mean of capturing terminological evolution “in the making”. It is a well known terminological theory that new terms are coined by using existing words of existing terms. While in the current experiment, we used only minimal linguistic resources (morphological tag of each word) to extract terms and relate them with lexical operations, we are aware that this is limited in that semantically close terms who do not share lexical association will be missed. Nevertheless, lexical similarity constitutes a very robust method for clustering domain terms. Our experiment tends to show that TermWatch, while not clustering from a reference terminology nor from information on co-occurrence of extracted terms in the documents, was able to automatically select a structured terminological resource for query refinement. In the future, we envisage to carry out experiments involving end users in a query refinement process.

## References


1. Ben-Dor A., Yakhini Z.. Clustering gene expression patterns. In *Proceedings of the Third Annual International Conference on Research in Computational Molecular Biology*, April 11-14, 1999, Lyon, France. ACM, 1999: pp. 33-42
2. Baeza-Yates, Ribeiro–Neto B. Query operations. In *Modern Information retrieval*. ACM Press, 1999, pp. 117-139.
3. Berry A., Kaba B., Nadif M., SanJuan E., Sigayret A. Classification et désarticulation de graphes de termes. In *7th International conference on Textual Data Statistical Analysis (JADT 2004)*, Leuven, Belgium, 10-12 march, 2004, pp. 160-170.
4. Blyth, T.S., Janowitz., M.F. Residuation Theory. Pergamon Press, 1972.
5. Buckley, C., Salton, G., Allen J., Singhal. A. Automatic query expansion using SMART: TREC-3. In D. K. Harman (ed.), *The Third Text Retrieval Conference (TREC-3)*. U.S. Department of Commerce, 1995.
6. Callon M., Courtial J-P., Turner W., Bauin S. From translation to network : The co-word analysis. In *Scientometrics* 5(1) 1983.
7. Celeux, G, Govaert G. Comparison of the mixture and the classification maximum likehood. In *clusters analysis. Journal of Statistical Computation and simulation* 47 1993, pp.127-146.
8. Courtial J-P. Introduction à la scientométrie. Anthropos – Economica, Paris, 1990, 135p.
9. Cutting, D.,R., Karger, D. R., Pedersen, J., O., Tukey, J. W. Scatter/Gather: a Cluster-based Approach to Browsing Large Document Collections. In *Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1992, pp. 318-329.
10. Daille, B. Study and Implementation of Combined Techniques for Automatic Extraction of Terminology. In P. Resnik and J. Klavans (eds) *The Balancing Act: Combining Symbolic and Statistical Approaches to Language*, MIT Press, 1996, pp. 49-66.
11. Dobrynin, V., Patterson D., Rooney N. Contextual Document Clustering. In *Proceedings of the European Conference on Information Retrieval*, Sunderland, UK, April 5-7 2004, pp. 167-180.
12. Feldman, R., Fresko, M., Kinar, Y. Text mining at the term level. In Zytkow, J. M., Quafafou, M. (Eds.), *Principles of Datamining and knowledge discovery*. *Proceedings of the 2nd European symposium PKDD.* Berlin-Springer, Nantes - France, 1998, pp. 65-73.
13. Fellbaum, C. (Ed.). WordNet, An Electronic Lexical Database. MIT Press, 1998.
14. Grabar, N., Zweigenbaum, P. Lexically-based terminology structuring: Some inherent limitations. In *Recent Trends in Computational Terminology: Special Issue of Terminology* 10 (1), 2004, pp. 23-53.
15. Matsuda, H., Ishihara, T., Hashimoto, A. Classifying Molecular Sequences Using a Linkage Graph With Their Pairwise Similarities. In *Theoretical Computer Science*, 1999, 210(2): pp. 305-325.
16. Hearst M. A. The use of categories and clusters in information access interfaces. In T. Strzalkowski (ed.), *Natural Language Information Retrieval*, Kluwer Academic Publishers, 1999, pp. 333-374.
17. Hofmann, T. Unsupervised learning by Probabilistic Latent Semantic Analysis. In *Machine Learning*, 42, 2001, pp. 177 - 196.
18. Ibekwe-SanJuan, F. A linguistic and mathematical method for mapping thematic trends from texts. In *Proceedings of the 13th European Conference on Artificial Intelligence*, Brighton UK, 23-28 August, 1998, pp. 170-174.

19. Ibekwe-SanJuan, F., SanJuan, E., Mining textual data through term variant clustering: the termwatch system. In: *RIAO Proceedings*, 2004, pp. 487-503.
20. Jacquemin C. Spotting and discovering terms through Natural Language Processing, MIT Press, 2001, 378p.
21. Jain, A. K., Murty, M. N. and Flynn, P. J. Data Clustering: A Review. In *ACM Computing Surveys*, 31(3), 1999.
22. Jain, A.K., Dubes, R.C. Algorithms for Clustering Data. Prentice Hall, Englewood Cliffs, NJ, 1988.
23. Leclerc B. The residuation model for the ordinal construction of dissimilarities and other valued objects. In Van Cutsem B. (eds.) Classification and dissimilarity analysis, *Lecture Notes in Statistics*, n° 93, Springer-Verlag, 1994, pp. 149-171.
24. Leydesdorf, L. Words and Co-Words as Indicators of Intellectual Organization. In *Research Policy* 18, 1989, pp. 209-223.
25. Milligan, G.W., Cooper M.C. A study of the comparability of external criteria for hierarchical cluster analysis. *In Multivariate Behavioural Research*, 21, 1986, pp. 441-458.
26. Morin E, Jacquemin C. Automatic acquisition and expansion of hypernym links. In *Computer and the humanities* 38 (4) 2004, pp. 363-396.
27. Nenadic, G., Spassic, I., Ananiadou, S. Mining term similarities from corpora. In *Recent Trends in Computational Terminology: Special Issue of Terminology* 10(1), 2004, p.34.
28. Pedersen, T., Patwardhan, Michelizzi. WordNet::Similarity - Measuring the Relatedness of Concepts. *In Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI)*, July 25-29, 2004, San Jose, CA.
29. Polanco X., Grivel L., Royauté J. How to do things with terms in informetrics : terminological variation and stabilization as science watch indicators. *Proceedings of the 5th International Conference of the International Society for Scientometrics and Informetrics*, Illinois USA, 7-10 June 1995, pp. 435-444.
30. Schiffrin, R., Börner, K. Mapping knowledge domains. *In Publication of the National Academy of Science (PNAS)*, 101(1) 2004, pp. 5183-5185.
31. Silberztein M. Dictionnaire électronique et analyse automatique des textes. Le système INTEX. Masson, Paris, 1993.
32. Small H.. Visualizing science by citation mapping. Journal of the American society for Information Science 50(9) 1999, pp. 799-813.
33. Yang, Y., Pierce, T., Carbonell, J. G. A Study on Retrospective and On-line Event Detection. In *ACM SIGIR Conference on Research and Development in Information Retrieval* 1998, pp. 28-36.
34. Yee Yeung, K. Clustering or automatic class discovery: non-hierarchical, non-SOM. In *A practical approach to microarray data analysis*, Kluwer Academic Publisher, 2003.
35. Yeung, K, Y., Haynor H., Ruzzo W, L. Validating Clustering for Gene Expression Data. In HYPERLINK "http://www.cs.washington.edu/homes/kayee/cluster" *Bioinformatics* HYPERLINK "http://www.cs.washington.edu/homes/kayee/cluster" , 17, 2001, pp. 309-318.
36. Yeung, K, Y., Ruzzo W, L. Details of the Adjusted Rand Index and clustering algorithms. Supplement to the paper "An experimental study on Principal Component Analysis for clustering gene expression data". In HYPERLINK "http://www.cs.washington.edu/homes/kayee/cluster" *Bioinformatics* HYPERLINK "http://www.cs.washington.edu/homes/kayee/cluster" 17, 2001, pp. 763-774.
37. Zamir, O. and Etzioni, O. Web document Clustering, A feasibility demonstration. In *ACM SIGIR Conference on Research and Development in Information Retrieval,* 1998, pp. 46 - 54.

# Automatic Filtering of Bilingual Corpora for Statistical Machine Translation

Shahram Khadivi and Hermann Ney

Lehrstuhl für Informatik VI - Computer Science Department
RWTH Aachen University
Ahornstrasse 55
52056 Aachen, Germany
{khadivi,ney}@cs.RWTH-Aachen.de

**Abstract.** For many applications such as machine translation and bilingual information retrieval, the bilingual corpora play an important role in training the system. Because they are obtained through automatic or semi automatic methods, they usually include noise, sentence pairs which are worthless or even harmful for training the system. We study the effect of different levels of corpus noise on an end-to-end statistical machine translation system. We also propose an efficient method for corpus filtering. This method filters out the noisy part of a corpus based on the state-of-the-art word alignment models. We show the efficiency of this method on the basis of the sentence misalignment rate of the filtered corpus and its positive effect on the translation quality.

## 1 Introduction

Bilingual corpora play an important role in developing statistical machine translation systems. But, since the manual compilation of bilingual corpora is a very expensive process, most of available bilingual corpora are generated in an automatic way. The parallel documents[1] are becoming more and more available, mainly on the Web, so that there is a need for automatic methods for bilingual corpus compilation. The automatically generated corpora usually include noise, sentence pairs which are worthless or even harmful for training the system. The noise might be due to any difference between the contents of source and target documents, non-literal translation, or errors in aligning documents, paragraphs, and sentences.

### Related Work

Automatically generated bilingual corpora usually contain a considerable number of noisy sentence pairs. These noisy sentence pairs may have a negative impact on the training of the statistical machine translation or bilingual information retrieval systems. Due to this problem, various researchers have investigated different methods for corpus filtering. Here, we give a brief overview of the important works done in this field.

In [1], the authors remove parallel documents for which their respective file size differ largely or for which, after applying sentence alignment, a relatively large number of

[1] Documents available in more than one language but with the same content.

A. Montoyo et al. (Eds.): NLDB 2005, LNCS 3513, pp. 263–274, 2005.
© Springer-Verlag Berlin Heidelberg 2005

empty alignments appear. They also make use of the length similarity between sentence pairs as well as the existence of bilingual dictionary entries in a sentence pair.

In [2] and [3], the authors make use of a literalness criterion for each sentence pair to filter a noisy Japanese-English corpus. They measure the literalness between source and target sentences by referring to a translation dictionary and counting the number of times that the translation dictionary entries occurred only in the source sentence, only in the target sentence, or in both source and target sentences.

In [4], the author studies the use of a noisy corpus in addition to a large clean training corpus in order to improve the translation quality. He identifies the noisy sentence pairs by accumulating five alignment scores for each sentence pair based on the following features: three different sentence length features and two lexical features based on IBM model 1 score. The sentence pairs which have a score less than a threshold are considered as noise.

In [5], the effect of parallel sentence extraction from in-domain comparable corpora on the machine translation performance has been studied. They align each sentence in a source document to all possible target sentences in several associated target documents. The associated target documents are the most similar documents to the source document among a relatively large number of documents. Then, they filter out noisy sentences by using two classifiers: a simple rule-based classifier and a maximum entropy based classifier. They show the significant improvement of the end-to-end translation quality, when the extracted corpus is added to the baseline out-of-domain corpus.

All above authors have investigated different methods for corpus filtering and showed the positive effect of corpus filtering on their statistical machine translation systems. But, the impact of the level of corpus noise on training the statistical machine translation models has not been specifically investigated. In this paper, we study the effect of different levels of corpus noise on an end-to-end statistical machine translation system. We also introduce an efficient approach for corpus filtering, and we show that which specific model among different statistical machine translation models must be trained on the filtered corpus.

The remaining part of this paper is organized as follows. Section 2 deals with our statistical translation engine. In section 3, we describe the corpus compilation procedure. In section 4, we introduce the length-based and translation likelihood-based filtering. In section 5, we deal with the experiment results and the related discussion.

## 2 Statistical Machine Translation

In statistical machine translation , we are given a source language ('French') sentence $f_1^J = f_1 \dots f_j \dots f_J$, which is to be translated into a target language ('English') sentence $e_1^I = e_1 \dots e_i \dots e_I$. Among all possible target language sentences, we will choose the sentence with the highest probability:

$$\hat{e}_1^I = \underset{e_1^I}{\operatorname{argmax}} \left\{ Pr(e_1^I | f_1^J) \right\} \tag{1}$$

$$= \underset{e_1^I}{\operatorname{argmax}} \left\{ Pr(e_1^I) \cdot Pr(f_1^J | e_1^I) \right\} \tag{2}$$

The decomposition into two knowledge sources in Equation 2 is known as the source-channel approach to statistical machine translation [6]. It allows an independent modeling of target language model $Pr(e_1^I)$ and translation model $Pr(f_1^J|e_1^I)$ [2]. The target language model describes the well-formedness of the target language sentence. The translation model links the source language sentence to the target language sentence. It can be further decomposed into alignment and lexicon model. The argmax operation denotes the search problem, i.e. the generation of the output sentence in the target language. We have to maximize over all possible target language sentences.

An extension to the classical source-channel approach is the direct modeling of the posterior probability $Pr(e_1^I|f_1^J)$. Using a log-linear model [7], we obtain:

$$Pr(e_1^I|f_1^J) = \exp\left(\sum_{m=1}^{M} \lambda_m h_m(e_1^I, f_1^J)\right) \cdot Z(f_1^J)$$

Here, $Z(f_1^J)$ denotes the appropriate normalization constant. The term $h_m(e_1^I, f_1^J)$ denotes various models which are involved in the translation process. And each model is weighted by its model scaling factor $\lambda_1^M$. As a decision rule, we obtain:

$$\hat{e}_1^I = \underset{e_1^I}{\operatorname{argmax}} \left\{ \sum_{m=1}^{M} \lambda_m h_m(e_1^I, f_1^J) \right\}$$

This approach is a generalization of the source-channel approach. It has the advantage that additional models or feature functions can be easily integrated into the overall system. The model scaling factors $\lambda_1^M$ are trained according to the maximum entropy principle, e.g. using the GIS algorithm. Alternatively, one can train them with respect to the final translation quality measured by some error criterion [8].

## 3 Corpus Compilation

Manual collection of bilingual text corpora might be very expensive, e.g. in the EuTrans project [9]. An efficient and cheap approach for collecting parallel text is mining the Internet for parallel documents [10] and [11]. For the purpose of bilingual corpus generation, we download the parallel documents in HTML format from *the European Union Website*[3] which exists in all official languages of the European Union.

The documents are already aligned at the document level. After extracting the plain text from HTML documents, we apply a hierarchical rule-based method for text tokenization. Then, by using an automatic sentence aligner program, we form a bilingual corpus. The last step is the corpus filtering to obtain a high quality bilingual corpus for training the translation models. The paragraph/sentence alignment, and the corpus filtering will be described in the next sub-section and Section 4, respectively.

[2] The notational convention will be as follows: we use the symbol $Pr(\cdot)$ to denote general probability distributions with (nearly) no specific assumptions. In contrast, for model-based probability distributions, we use the generic symbol $p(\cdot)$.

[3] `http://europa.eu.int`

**Sentence Alignment**

We employ an improved version of the Gale and Church algorithm [12] to align the paragraphs and sentences in the parallel documents. The Gale and Church algorithm is a dynamic programming method for aligning corresponding sentences in two parallel documents. This method is based on the length of the two sentences to be aligned. The length of the sentence is measured in term of its characters.

We extend the algorithm in the following ways. In order to improve the sentence alignment quality, we make use of a small dictionary. Each entry in the dictionary is used as an anchor point. In other words, each entry in the dictionary tells the sentence alignment algorithm if the source word (phrase) of a given entry exists in source sentence then the target word (phrase) might be in the target sentence. To each entry of the dictionary, two values have been assigned: presence bonus and absence (mismatch) penalty. By introducing these two values, the dictionary entries can be taken into account in alignment algorithm more easily and efficiently.

We also integrate several heuristics to the Gale and Church algorithm. Some of them are as follows:

- If the source sentence starts with a digit or an item symbol, it is very likely that the target sentence starts with the same digit or symbol.
- The number of numerical sequences in two aligned sentences should be very close to each other.
- The number of parenthesis pairs in two aligned sentences should be very close to each other.

There are a few other sentence alignment algorithms available like [13], [14], [15], [16], and *Champollion*, the sentence aligner of the Linguistic Data Consortium[17]. In [18], there is an evaluation on different sentence alignment methods on the task of Portuguese and English parallel texts. They mentioned that due to the very similar performance of the methods, choosing the best sentence aligner is not an easy task. In addition, they found that the performance of the sentence aligners vary for different tasks. Due to this result and the aim of our research which is studying the effect of noise on statistical machine translation system, finding and employing the best sentence aligner is not a crucial matter in this task.

To obtain a bilingual corpus from the documents available in *the European Union Website*, we apply the developed sentence aligner in two steps: aligning the paragraphs of documents, and aligning the sentences within two aligned paragraphs. The following alignment mappings between source and target sentences are permitted: 1:1, 1:0, 0:1, 2:2, 2:1, 1:2. In this paper, we refer to the bilingual corpus obtained from *the European Union Website* as the EU corpus.

## 4 Corpus Filtering

The automatic methods for corpus compilation are noise-prone. The main reasons are sentence alignment errors and the existence of the noise in the original source and target documents. The latter is very obvious in comparable corpora which make them more

complicated for obtaining bilingual sentence pairs. But also in parallel documents, two corresponding documents may not be fully suitable for a bilingual corpus extraction due to free translation or summarization of the original text, selective translation of important parts of the original document, or existence of useless data like tables in the parallel documents. An additional type of noise is caused by the use of a third language in the original documents. For example, in the EU Spanish corpus, we may have a sequence of words in French.

Thus, there is high probability that an automatically aligned corpus contains many noisy sentence pairs which are worthless or even harmful for statistical machine translation . Therefore, we need a filtering scheme to remove the noisy sentence pairs from the corpus. We will describe two different methods for corpus filtering:

- a *length-based filtering* which makes use of length constraints of sentence pairs,
- a *translation likelihood-based* which makes use of the translation likelihood measure for the given sentence pairs.

### 4.1 Length-Based Filtering

We develop a length-based filtering algorithm to remove presumably harmful or worthless sentence pairs. The rules are based upon the length of the source and target sentences and work as follows:

1. The lengths of source sentence and target sentence must not differ largely. When $I$ and $J$ denote the lengths of target and source sentences, respectively, the above rule can be expressed by the following detailed rules:
   - $(6 \cdot I > J \wedge I < 6 \cdot J)$
   - $(I < 3 \vee J < 3 \vee (I < 2.2 \cdot J \wedge J < 2.2 \cdot I))$
   - $(I < 10 \vee J < 10 \vee (I < 2 \cdot J \wedge J < 2 \cdot I))$
2. At least one alphabetical character must occur in each sentence of a sentence pair.
3. The sentence end symbols in source sentence and target sentence must be similar.
4. The source sentence and the target sentence must not be empty.

In addition, we also identify the language of the source and target sentences to be in the language which is supposed. The developed language identification system is a maximum entropy based language classifier for identifying the language of each text line using the YASMET toolkit [19]. The maximum entropy features are the most frequent trigrams of each language.

The main problem of this filtering scheme is that it also removes many correct or useful sentence pairs from the corpus. In other words, the length-based method can clean the corpus with a high precision but with a low recall.

### 4.2 Translation Likelihood-Based Filtering

In order to filter out worthless or harmful sentence pairs from the compiled bilingual corpus in a more systematic scheme, we make use of the translation probability of each sentence pair which is produced by word alignment models [20] and [21]. For this purpose, we train IBM model 1, Hidden Markov model, and IBM model 4 in a successive

manner using the maximum likelihood algorithm on the whole corpus (unclean corpus). The final parameter values of a simpler model serve as starting point for a more complex model. We train these models in both directions, from source to target and from target to source. Hence, for each sentence pair we have two probabilities: $\sum_{a_1^J} p(f_1^J, a_1^J|e_1^I)$ and $\sum_{b_1^I} p(e_1^I, b_1^I|f_1^J)$ where $a_1^J$ / $b_1^I$ is an alignment which describes a mapping from the source / target position $j$ / $i$ to the target/source position $a_j$ / $b_i$ . By scaling these probabilities with the source and target sentence lengths, we arrive at the following score for each sentence pair $(f_1^J, e_1^I)$ in the corpus:

$$\begin{aligned}\text{Score}(f_1^J, e_1^I) = &\frac{1}{J} \log \sum_{a_1^J} p(f_1^J, a_1^J|e_1^I) + \\ &\frac{1}{I} \log \sum_{b_1^I} p(e_1^I, b_1^I|f_1^J)\end{aligned} \tag{3}$$

A very good approximation and computationally efficient variation of Equation 3 is achieved by calculating the Viterbi alignment instead of the summation over all alignments, i.e. by replacing the the $\sum$ operator with $\max$ operator in the equation:

$$\begin{aligned}\text{Score}(f_1^J, e_1^I) = &\frac{1}{J} \log \max_{a_1^J} p(f_1^J, a_1^J|e_1^I) + \\ &\frac{1}{I} \log \max_{b_1^I} p(e_1^I, b_1^I|f_1^J)\end{aligned} \tag{4}$$

Now, we have a score for each sentence pair. We empirically determine the threshold value for discriminating correct sentence pairs from incorrect sentence pairs.

The translation likelihood scores can also be utilized for corpus weighting instead of corpus filtering. It means that the sentence pairs with a better score will get a higher weight. It reduces the impact of noisy sentence pairs on training the statistical machine translation models.

## 5 Results

Here we will present the results of the corpus filtering for two corpora, Xerox and EU. The language pair for both corpora is Spanish-English. The Xerox corpus (Table 1) is a noise free corpus which has been manually aligned, it is composed of technical manuals describing various aspects of Xerox hardware and software installation, administration, usage, etc. The EU corpus (Table 2) has been automatically aligned and is a noisy corpus (details in Section 3).

To study the effects of noise on an end-to-end statistical machine translation , we use the Xerox corpus. Then, by introducing artificial noise on this corpus, we study the effect of noise on statistical machine translation . We make use of the EU corpus as a real case study for analyzing the effect of translation likelihood-based filtering.

We evaluate the proposed filtering scheme by two criteria, sentence alignment evaluation and end-to-end translation quality.

**Table 1.** Statistics of the Xerox corpus.

| | English | Spanish |
|---|---|---|
| Train: Sentences | 56K | |
| Words | 665K | 753K |
| Vocabulary Size | 8K | 11K |
| Vocabulary Singletons | 2K | 3K |
| Test: Sentences | 1125 | |
| Words | 8K | 10K |

**Table 2.** Statistics of the EU corpus.

| | English | Spanish |
|---|---|---|
| Train: Sentences | 975K | |
| Tokens | 19M | 22M |
| Vocabulary Size | 73K | 94K |
| Vocabulary Singletons | 25K | 32K |
| Test: Sentences | 2000 | |
| Words | 48K | 54K |

### 5.1 Sentence Alignment Evaluation

To evaluate the sentence alignment quality, we select the EU corpus as a noisy corpus. We generate two clean corpora by applying each of the two corpus filtering methods to the noisy corpus. In each corpus, we keep just one instance per sentence pair, then the sentence alignment evaluation will be more accurate. We randomly selected 400 sentence pairs from each corpus. Then, we asked an expert to judge sentence alignment accuracy in all sentence pairs by assigning correct or incorrect to each pair. The details of sentence alignment evaluation for the length-based filtered corpus and translation likelihood-based filtered corpus are shown in in Table 3.

**Table 3.** Sentence Alignment Evaluation.

| Filtering method | Misalignment error rate [%] |
|---|---|
| Length-based | 5.0 |
| Translation likelihood-based | 3.2 |

The results show that the translation likelihood-based filtering is better than the length-based filtering in removing incorrectly aligned sentence pairs. At the same time, we observed that the number of filtered sentence pairs in the translation likelihood-based filtering is less than the length-based filtering. This observation along with the significance test confirm the superiority of translation likelihood-based filtering over length-based filtering.

In order to measure the efficiency of translation likelihood-based filtering , we perform another experiment on the Xerox corpus. We introduce different levels of noise to

the clean Xerox corpus, by randomly scrambling a given amount of the sentence pairs, i.e. we randomly select two sentence pairs with about the same length and then we make them noisy by exchanging their target parts. After making the corpus noisy, we apply the translation likelihood-based filtering to the corpus and measure its accuracy in identifying the noisy sentence pairs. This type of noise can not be identified by any length-based filtering approaches. Table 4 shows the results of this experiment, the first column is the level of introduced artificial noise to the clean corpus and the second column is the accuracy of identifying the noisy sentence pairs. The results show even with 80% noise in the corpus the translation likelihood-based filtering is able to identify the noisy sentence pairs with the accuracy about 90%.

**Table 4.** Sentence Alignment Evaluation of the Xerox Artificial Noisy Corpus.

| Fraction of incorrect sentence pairs [%] | Filtering error rate [%] |
|---|---|
| 20 | 10.4 |
| 40 | 11.9 |
| 60 | 13.0 |
| 80 | 11.6 |

### 5.2 Translation Results

In this section, we study the effect of corpus noise on the translation quality of an end-to-end statistical machine translation . We make use of a phrase-based translation engine [7]. In all translation experiments, we will report the baseline translation results in BLEU score [22].

In the first experiment, we study the effect of different levels of corpus noise on the translation quality. Again, we use the Xerox corpus which is a clean corpus, and introduce different levels of artificial noise to the corpus with the same method as described in the last section. Then, we train the statistical machine translation models on each of the artificially noisy corpora. The translation results are shown in Table 5. The first column shows the percentage of noisy sentence pairs in the corpus. The second column shows the translation scores in BLEU when the full corpus is utilized for training the system. The last column shows the translation results when only the clean part of the corpus is used for training.

As we expected, the translation quality decreases with a growing level of noisy sentence pairs. Another important observation of this table is the difference between the translation results if we use the whole corpus or only the clean part. Even with about 40% of noise, the difference in BLEU score is about 0.4%. In summary, this table states that the corpus noise does not deteriorate the translation results.

There exist three important models in training our statistical machine translation system. They are word alignment model(s) (WA), bilingual-phrase model (BP), and language model (LM). The bilingual-phrase model must be trained on the clean part of a corpus, as there is no useful bilingual information assumed to be in the noisy sentence pairs. But, the effect of noise in training the word alignment model (WA) and its impact

**Table 5.** Translation Results on the Xerox Artificially Noisy Corpus.

| Artificial Noise [%] | the whole corpus BLEU [%] | only clean part of the corpus BLEU [%] |
|---|---|---|
| 0 | 61.2 | 61.2 |
| 20 | 60.4 | 59.7 |
| 40 | 58.1 | 58.5 |
| 60 | 52.4 | 54.2 |
| 80 | 44.0 | 49.1 |

on the translation quality is unclear. We also study the effect of noise in training the language model (LM), as one type of corpus noise is the existence of the sentences from another language in the corpus.

In the second experiment with the Xerox corpus, we study the effect of noise in training the word alignment model (WA). We make again about 20% of the Xerox corpus noisy, then we reorder the sentence pairs in the corpus according to the translation likelihood scores in ascending order. It means that the noisy sentence pairs supposed to be at the end of the corpus, i.e. from 80% to 100%. Table 6 shows the experimental results for this noisy corpus.

**Table 6.** Translation Results on a 20% Artificial Noisy Xerox Corpus.

| Fraction of Sentence Pairs[%] | filtering on BP BLEU[%] | filtering on WA+BP BLEU[%] |
|---|---|---|
| 10.0 | 38.7 | 30.0 |
| 40.0 | 57.3 | 47.9 |
| 70.0 | 59.8 | 59.6 |
| 75.0 | 59.7 | 59.6 |
| 77.5 | 59.9 | 59.9 |
| 80.0 | 60.0 | 59.7 |
| 82.5 | 60.4 | 60.2 |
| 85.0 | 60.1 | 59.6 |
| 90.0 | 59.8 | 60.1 |
| 95.0 | 60.2 | 60.1 |
| 100.0 | 60.4 | 60.4 |

The first column shows the percentage of the sentence pairs extracted from the first of the corpus. The second column depicts the translation results when only the clean part of the corpus is used for training the BP model (WA is trained with the whole corpus). The third column contains the translation results when the WA and BP models are trained both with the clean part of the corpus. In this experiment, the language model is always trained with the whole corpus, as there is no noise in the target sentences of the Xerox corpus. The contents of this table state that training the word alignment with the whole corpus (including noise) causes slightly better translation quality.

**Table 7.** Translation Results on the EU corpus.

| Fraction of Sentence Pairs[%] | filtering on BP BLEU[%] | WA+BP BLEU[%] | WA+BP+LM BLEU[%] |
|---|---|---|---|
| 92.5 | 46.9 | 46.6 | 46.6 |
| 95.0 | 47.1 | 46.8 | 46.7 |
| 97.5 | 47.2 | 46.8 | 46.8 |
| 98.3 | 47.0 | 46.9 | 46.9 |
| 100.0 | 46.8 | 46.8 | 46.8 |

We continue the experiments with a real noisy corpus, the EU corpus (Table 2). We reorder the sentence pairs in the corpus based on the translation likelihood-based filtering score. A human expert estimates about 1.7% to 2.0% sentence misalignment rate in the corpus, depending on the accuracy of judgment. We performed a set of experiments to study the effect of noise on the translation results and also its effect on different models, as shown in Table 7.

This table also shows the best translation results are achieved when the word alignment and language models are trained on the whole corpus. As it can be expected the filtering on language modeling training has no effect, as the level of noise in the target (monolingual) corpus is not considerable. The difference in BLEU score between the clean corpus (97.5% of the corpus) and the whole corpus is about 0.4%. However, a significance analysis [23] on the sentence level between these two systems show that the improvement is statistically significant.

**More Experiments**

We continue our experiments by including new scores to the Equation 4. We consider the following scores: normalized source language model ($\frac{1}{J}\log p(f_1^J)$), normalized target language model ($\frac{1}{I}\log p(e_1^I)$), and sentence length-difference penalty model ($\log(1/(1.0+|I-J|)$). The language models seem to be useful for filtering those sentences which are from another language than the language of the corpus. The sentence-length difference penalty model explicitly penalizes the dissimilarity between source and target lengths. The translation experiments did not show any translation quality improvement when we made use of these extended models over the word alignment models. It seems that the word alignment models are robust enough against the noise of the EU corpus. We have also studied the idea of corpus weighting instead of corpus filtering by using translation likelihood filtering. The translation result when we utilized the weighted corpus for training the system was 46.9% BLEU. It means using the weighted corpus had no superiority over using the simple corpus.

## 6 Conclusions

In this paper, we presented an efficient approach for corpus filtering. The experiments showed that translation likelihood-based filtering is a robust method for removing noise

from the bilingual corpora. It improves the sentence alignment quality of the corpus and at the same time keeps the corpus size as large as possible. It has also been shown that the translation likelihood-based filtering enhances the training corpus for the translation task. The translation quality of the filtered training corpus has statistically significant improvement over the noisy corpus. One surprising result of the experiments is that even a large percentage of incorrect sentence pairs does not seem to deteriorate the performance of a statistical machine translation system. It means that the statistical machine translation models are robust enough against the corpus noise.

## Acknowledgments

This work has been funded by the European Union under the RTD project TransType2 (IST 2001 32091) and the integrated project TC-STAR - Technology and Corpora for Speech to Speech Translation -(IST-2002-FP6-506738, http://www.tc-star.org).

## References

1. Nie, J., Cai, J.: Filtering noisy parallel corpora of web pages. In: IEEE Symposium on NLP and Knowledge Engineering, Tucson (2001) 453–458
2. Imamura, K., Sumita, E.: Automatic construction of machine translation knowledge using translation literalness. In: 10th Conference of the European Chapter of the Association for Computational Linguistics, Budapest, Hungary (2003) 155–162
3. Imamura, K., Sumita, E.: Bilingual corpus cleaning focusing on translation literality. In: 7th International Conference on Spoken Language Processing (ICSLP-2002), Denver, Colorado (2002) 1713–1716
4. Vogel, S.: Using noisy bilingual data for statistical machine translation. In: 10th Conference of the European Chapter of the Association for Computational Linguistics, Budapest, Hungary (2003) 175–178
5. Munteanu, D.S., Fraser, A., Marcu, D.: Improved machine translation performance via parallel sentence extraction from comparable corpora. In Susan Dumais, D.M., Roukos, S., eds.: HLT-NAACL 2004: Main Proceedings, Boston, Massachusetts, USA, Association for Computational Linguistics (2004) 265–272
6. Brown, P.F., Cocke, J., Della Pietra, S.A., Della Pietra, V.J., Jelinek, F., Lafferty, J.D., Mercer, R.L., Roossin, P.S.: A statistical approach to machine translation. Computational Linguistics **16** (1990) 79–85
7. Och, F.J., Ney, H.: Discriminative training and maximum entropy models for statistical machine translation. In: Proc. of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, PA (2002) 295–302
8. Och, F.J.: Minimum error rate training in statistical machine translation. In: Proc. of the 41th Annual Meeting of the Association for Computational Linguistics (ACL), Sapporo, Japan (2003) 160–167
9. Vidal, E., et al.: Final report of esprit research project 30268 (EuTrans): Example-based language translation systems. Technical report (2000)
10. Resnik, P.: Mining the web for bilingual text. In: Proc. of the 37th Annual Meeting of the Association for Computational Linguistics (ACL), University of Maryland, College Park, MD (1999) 527–534

11. Chen, J., Nie, J.Y.: Automatic construction of parallel english-chinese corpus for cross-language information retrieval. In: Proceedings of the sixth conference on Applied natural language processing, Seattle, Washington, Morgan Kaufmann Publishers Inc. (2000) 21–28
12. Gale, W.A., Church, K.W.: A program for aligning sentences in bilingual corpora. Computational Linguistics **19** (1993) 75–102
13. Zhao, B., et al.: Efficient optimization for bilingual sentence alignment based on linear regression. In: HLT-NAACL 2003 Workshop: Building and Using Parallel Texts: Data Driven Machine Translation and Beyond, Edmonton, Canada (2003) 81–87
14. Melamed, I.D.: Pattern recognition for mapping bitext correspondence. In Véronis, J., ed.: Parallel Text Processing: Alignment and Use of Translation Corpora. Kluwer Academic Publishers (2000) 25–47
15. Melamed, I.D.: A geometric approach to mapping bitext correspondence. In Brill, E., Church, K., eds.: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Somerset, New Jersey, Association for Computational Linguistics (1996) 1–12
16. Simard, M., Foster, G., Isabelle, P.: Using cognates to align sentences in bilingual corpora. In: Fourth Int. Conf. on Theoretical and Methodological Issues in Machine Translation (TMI-92), Montreal, Canada (1992) 67–81
17. LDC: Champollion tool kit (2004) `http://champollion.sourceforge.net/` .
18. Caseli, H.M., Nunes, M.G.V.: Evaluation of sentence alignment methods on portuguese-english parallel texts. Scientia **14** (2003) 1–14
19. Och, F.J.: YASMET: Toolkit for conditional maximum entropy models (2001) http://www-i6.informatik.rwth-aachen.de/~och/software/YASMET.html.
20. Vogel, S., Ney, H., Tillmann, C.: HMM-based word alignment in statistical translation. In: COLING '96: The 16th Int. Conf. on Computational Linguistics, Copenhagen, Denmark (1996) 836–841
21. Brown, P.F., Della Pietra, S.A., Della Pietra, V.J., Mercer, R.L.: The mathematics of statistical machine translation: Parameter estimation. Computational Linguistics **19** (1993) 263–311
22. Papineni, K.A., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. Technical Report RC22176 (W0109-022), IBM Research Division, Thomas J. Watson Research Center (2001)
23. Bisani, M., Ney, H.: Bootstrap estimates for confidence intervals in asr performance evaluationx. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, Montreal, Canada (2004) 409–412

# An Approach to Clustering Abstracts*

Mikhail Alexandrov[1,2], Alexander Gelbukh[1], and Paolo Rosso[2]

[1] Center for Computing Research, National Polytechnic Institute, Mexico
dyner1950@mail.ru, gelbukh@gelbukh.com
www.gelbukh.com
[2] Polytechnic University of Valencia, Spain
prosso@dsic.upv.es

**Abstract.** Free access to full-text scientific papers in major digital libraries and other web repositories is limited to only their abstracts consisting of no more than several dozens of words. Current keyword-based techniques allow for clustering such type of short texts only when the data set is multi-category, e.g., some documents are devoted to sport, others to medicine, others to politics, etc. However, they fail on narrow domain-oriented libraries, e.g., those containing all documents only on physics, or all on geology, or all on computational linguistics, etc. Nevertheless, just such data sets are the most frequent and most interesting ones. We propose simple procedure to cluster abstracts, which consists in grouping keywords and using more adequate document similarity measure. We use Stein's MajorClust method for clustering both keywords and documents. We illustrate our approach on the texts from the Proceedings of a narrow-topic conference. Limitations of our approach are also discussed. Our preliminary experiments show that abstracts cannot be clustered with the same quality as full texts, though the achieved quality is adequate for many applications; accordingly, we suggest Makagonov's proposal that digital libraries should provide document images of full texts of the papers (and not only abstracts) for open access via Internet, in order to help in search, classification, clustering, selection, and proper referencing of the papers.

## 1 Introduction

### 1.1 Difficulties in Clustering Short Documents

In Information Retrieval, clustering algorithms are used to analyze large collections of documents by means of subdividing them into groups of similar documents. A typical approach to document clustering in a given domain is to transform the textual documents into vector form basing on a list of index keywords and then use well-known numerical procedures of cluster analysis [13]. The list of keywords is constructed from a training document set belonging to the same domain. If the domain is unknown then this list is constructed directly from the document collection itself. The keyword list used for document representation is weighted using, for example, the well-known *tf-idf* technique [15].

* Work done under partial support of the Government of Valencia, Mexican Government (CONACyT, SNI, CGPI, COFAA-IPN), R2D2 CICYT (TIC2003-07158-C04-03), and ICT EU-India (ALA/95/23/2003/077-054).

A. Montoyo et al. (Eds.): NLDB 2005, LNCS 3513, pp. 275–285, 2005.
© Springer-Verlag Berlin Heidelberg 2005

Currently such an approach is used for clustering not only full-text documents, but also for clustering short documents containing 50–100 words – as, for example, news, brief historical or advertising information, etc. However, the fact that good results have been obtained in these particular cases is not a reason for optimism: this approach gives very *unstable* or *imprecise* results when clustering abstracts of scientific papers, technical reports, patents, etc. Nevertheless, just these cases are the most interesting ones: most digital libraries and other web-based repositories of scientific and technical information nowadays provide free access only to abstracts and not to the full texts of the documents. Therefore, the results prove to greatly depend on the type of short documents being clustered. Let us consider the following two different cases:

1. Document collection containing the documents that belong to essentially different domains, such as sport, culture, politics, etc.
2. Document collection containing the documents from one narrow domain, such as physics, linguistics, urbanistics, etc.

This partition of document collections may seem very subjective. For example, in the domain of *physics* we can distinguish *nuclear physics*, *optics*, *chemical physics*, *experimental physics*, etc. In fact by different domains we mean the domains whose keyword vocabularies have no or very few words in common. In this case the size of the documents is not important for clustering in the keyword space: any clustering procedure will divide such documents into clusters - which are just the domains - well enough, since the documents are mapped to completely different keyword subspaces of the whole keyword space of the document collection.

A weak intersection of domain vocabularies can slightly disfigure the results: some documents can contain keywords only from this intersection, which is much more probable for short documents than for large ones. To avoid such an effect, one can remove common words from the list of index keywords and work only with the "pure" domains without keywords in common, as described above. Some short documents may in this case prove not to have any index keywords; they can be collected together and classified manually. The words in common can be found by a simple two-step procedure: (1) construct the list of keywords for the whole document collection and use it for clustering all documents; (2) construct the keyword lists for every document cluster and select the words in common.

When we deal with documents from one given domain, the situation is cardinally different. All clusters to be revealed have strong intersections of their vocabularies and the difference between them consists not in the set of index keywords but in their proportion. This causes very unstable and thus very imprecise results when one works with short documents, because of very low absolute frequency of occurrence of the keywords in the texts. Usually only 10%–20% of the keywords from the complete keyword list occur in every document and their absolute frequency usually is 1 or 2, sometimes 3 or 4. In this situation, changing a keyword's frequency by 1 can significantly change the clustering results. Thus the first difficulty consists in:

– very low absolute frequencies of occurrences of the keywords from the general keyword list in the text, which leads to unstable results.

Thus our first problem is to assure stability and thus correctness of the results of clustering.

Consider now another classification of short documents:

1. Document collection containing news or other self-contained information;
2. Document collection containing abstracts of full-text scientific or technical documents not included in the collection.

When clustering a new data collection, we must assure first of all validity of clustering results, i.e., good quality of grouping according to the given internal criteria. When we work with abstracts, we always have an additional specific criterion of closeness between abstracts being clustered and the corresponding full-text documents. Consider manual clustering of full texts as the gold standard, we can assess usability of our clustering results. Some researchers evaluate the usability comparing automatic versus manual clustering of only abstracts; however, we use a more rigorous gold standard consisting in dealing with full texts.

Our experiments with clustering abstracts and full-text papers show that it is practically impossible to obtain coincident or at least very close clusters. In fact, abstracts and full-text documents have different contents: Indeed, the abstracts explain the goals of the research reported in the paper (the problem), while the paper itself explains the methods used to achieve these goals (e.g., the algorithms). In consequence, a collection of abstracts and a collection of full-texts documents have significantly different keyword lists; at least they use the lexicon in different ways. Thus, the second difficulty consists in:

– significant difference between the use of keywords in abstracts and their full-text counterparts, which leads to imprecise results.

Consequently, our second goal is to provide more exact results with respect to the closeness between clustering abstracts and full papers. The problem of clustering abstracts arises when one works with documents from one narrow domain. Abstracts belonging to significantly different domains are clustered well, but this case is not interesting; any search engine can easy classify such abstracts.

### 1.2 Related Work

Though there exists extensive literature on information retrieval [3, 21], the problem of clustering narrow-domain short documents is not well-studied. One of the reasons for this consists in that when clustering algorithms are applied to multi-domain document collections no problems arise.

The only works concerning categorization of short documents we are aware of use supervised methods, i.e., are based on prior training [8; 22]. These works obtained excellent results, but for a different situation, because we deal in an unsupervised manner with clusters that are unknown beforehand, rather than with predefined categories. Makagonov *et al.* [10] considered the problem of clustering abstracts. However, the document collection contained the texts from easily distinguishable domains, and the number of domains was known beforehand.

Makagonov *et al.* [12] used stronger criteria of keyword selection and a combined measure of closeness between documents (cosine and polynomial ones). These criteria can give more confidence to the low absolute frequencies of keyword occurrences in abstracts; the combined measure can make the results closer to expert opinion. However, both techniques received some critics because they were not justified well enough and were not tested in more complex situation (the number of clusters was

known in advance). Alexandrov *et al*. [2] considered two procedures for clustering abstracts from a given narrow domain using clustering keywords. Those results were very preliminary, and no discussion of limitations of the method was provided. Besides, both works used well-known clustering methods - *k*-means and the nearest neighbor method - while there exists very promised method MajorClust [17], which have been demonstrated to give excellent results on clustering textual documents in comparison with the two mentioned methods [18].

In this paper we suggest the following modifications of the traditional approach, which significantly improve clustering results:

- Grouping words from the word frequency list, to make the results more stable and correct;
- Transformation of usual cosine measure of document similarity, to take into account the difference between abstracts and full-text documents.

The first suggestion is close to that proposed in [9], where clusters of keywords were used for constructing semantic space for information retrieval problems. Our procedure is simpler: we do not use any linguistic information on the relation betweens words; we rely only on links extracted with statistical methods from a document corpus. Our second suggestion continues that proposed in [12], where combined measure of document similarity was used. Again, our procedure is simpler: we use logarithmic transformation of term frequencies, inspired by the information-theoretic aspect of the problem.

For clustering keywords, we used MajorClust method. In the paper we pay attention to the limitations caused by grouping keywords and their stronger selection.

## 2 Main Algorithm

### 2.1 Indexing

**Preliminary keyword selection.** In the framework of the traditional approach, random character of document presentation (as considered in the vector space model) is not taken into account: all word occurrences are considered fixed values; this is justified for long documents. The corresponding procedure finds all words in the whole document collection, filters out stopwords, and joins the words having the same base meaning using, for example, Porter stemming algorithm [14].

In case of abstracts, we have a different situation. Namely, one or two occurrences of any low-frequency word in a text double its frequency count. Because of the random nature of such occurrences, the error of the frequency estimation becomes comparable to the frequency itself. To increase the confidence for low-frequency words, we have to perform word selection. For this, we use the following criterion introduced in [11]: only those words $W$ are included in the index keyword list that satisfy the following inequality:

$$F_{Dom}(W) >> F_{Gen}(W); \text{ namely, } F_{Dom}(W) / F_{Gen}(W) > k, \quad (1)$$

where $F_{Dom}(W)$ and $F_{Gen}(W)$ are the frequencies of the word $W$ in the given document collection and in a general balanced corpus of the given language (a corpus of general use), respectively.

The parameter $k$ is determined empirically. Its value is related to the statistical estimation of the mean error in the measuring of the frequencies due to the limited size

of the sample texts. A reasonable value for $k$ must be greater than 3 or 4 for low-frequency words in short texts; in our practical work we used $k = 7$ in order to obtain stable enough results.

On the other hand, a stable result does not mean a good result: taking only one keyword we would obtain the most stable but absolutely useless result. Indeed, eliminating words we lose certain information; extremely strong filtering leads to absurd results. Thus in our experiments we used $k = 2$ while stability of results was achieved by grouping keywords as described below (which allows for a lower threshold $k$).

To conflate the words having the same base meaning, we used empirical formulas for testing word similarity [1].

**Grouping keywords and weighting.** Grouping keywords is an efficient way to compensate for the effect of low frequencies of these keywords. In this case, every group of keywords can be considered a new coordinate in the index space, equal to the sum of the occurrences of all keywords in a given group. We will call this sum *cluster frequency*. One can suggest several variants of how to group the words having close semantics. For example:

- use synsets of an appropriate ontology (e.g., WordNet or a synonym dictionary);
- use a thesaurus related to a given domain;
- cluster the words in the space of documents themselves.

Two former variants use external information. The latter variant is the simplest; we discuss just this variant in this paper. However, here we have to answer on two important questions:

- How many keyword clusters should one take for clustering documents?
- How to evaluate semantic significance of each keyword cluster?

Obviously, the result of clustering documents crucially depends on the number of keyword clusters. To solve such a problem, we suggest using one of density-based methods. These methods automatically determine the number of clusters; one of them, MajorClust, constructs very natural clusters (in the sense that the revealed clusters are the closest ones to those selected by human experts). This method was suggested in [17] and investigated in [4, 18, 19, 20].

When grouping keywords, we suppose a certain semantic closeness between them. Indeed, we suppose that the absence of some words from a cluster may be compensated for by the presence of the others. To evaluate the semantic significance of a cluster, we use the average distance between all words included in the cluster:

$$S_k = \sum_{i,j} d_{i,j} / N_k, \tag{2}$$

where $k$ is the number of the cluster, $i$ and $j$ are the elements of this clusters ($i \neq j$), $N_k$ is the number of links in the cluster $k$ .

## 2.2 Clustering

**Document similarity measure.** We consider the vector model of document representation. To evaluate the closeness between two documents, we use the well-known cosine measure:

$$C_{1,2} = \frac{\sum_k (x_{k1}, x_{k2})}{\|x_1\| \|x_2\|}, \tag{3}$$

where $x_{k1}$ and $x_{k2}$ are the vectors corresponding to the documents 1 and 2. In our case, $x_{k1}$ and $x_{k2}$ are relative cluster frequencies. The difference from the traditional cosine measure consists in the following:

- the coordinates in (3) correspond to the clusters of keywords as described above,
- these coordinates are weighted using the semantic coefficients from (4) below.

We have already mentioned that in case of clustering abstracts one should take into account the difference between contents of abstracts and their full-text papers. Namely, the direct ratio of the coordinates should be changed taking into account the information-theoretic aspect of the problem. Indeed, the abstracts usually introduce the reader to the possibilities of a suggested approach or method, while the full papers give its more or less detailed explanation. This leads to the necessity to change the document representation in the document similarity measure to use logarithm of the vector coordinates:

$$x_k = \log(1 + f_k), \tag{4}$$

where $f_k$ are the relative cluster frequencies. It is these coordinates that are included in (3). The experiments described in the next section support this hypothesis.

**Clustering methods.** In the suggested approach clustering is applied twice: for grouping keywords and for grouping abstracts. It is well-known that the number of methods and their modifications used in cluster analysis are more than authors working in this area. [7]. Extensive literature is devoted to such methods and their applications in text processing [13, 19].

For clustering abstracts, we used the *K*-medoid method, the nearest neighbor method, and MajorClust method. These are the simplest implementations of the exemplar-based, hierarchy-based, and density-based approaches, respectively. Our goal was not selecting the best clustering method for abstracts; from the previous discussion it is clear that this is separate and difficult task. We used the former two methods, since they are in a sense the most "contradictory" ones: they give the least coincident results on various data sets as compared with other pairs of clustering methods. This was noted by Solomon [16], where instead of *K*-medoid method the usual *K*-means was tested. This was also investigated in detail by Stein *et al*. [4, 18]. Thus closeness of the results of these methods would be a strong indication of stability of obtained clusters.

MajorCluster is a new method firstly desribed in [17]. We slightly modified it in order to avoid circling related with weak connections between the documents due to their small size.

The idea of the method is very simple: it distributes objects to clusters in such a way that the similarity of an object to the assigned cluster exceeds its similarity to any other cluster. Neither *K*-means (*K*-medoid) nor the nearest neighbor (NN) methods possess such optimization property: the former one provides the maximum closeness of objects to the centers of clusters to be constructed, while the latter one provides maximum connectivity of objects inside clusters, independently of their similarity to

other clusters. MajorCluster method works as follows: first, every object is considered a separate cluster. Then the objects are joined to the nearest cluster. In the process of cluster construction, the objects can change their cluster – in contrast to the NN-method. This algorithm is an implementation of the algorithm for graph clustering based on the notion of weighted edge connectivity [17].

For clustering words we used only MajorClust method, because we have no a priori information concerning the number of keyword clusters. Numerous experiments show that this method outperforms both other methods, independently of index selection for the given document collection, such as RCV1-Reuters Corpus, vol. 1 [18–20].

## 3 Experiments with Web-Retrieved Information

### 3.1 Data Source and Clustering Quality

**CICLing-2002 conference collection.** In our experiments, we used a document collection consisting of the abstracts of the CICLing-2002 Conference (Conference on Computational Linguistics and Intelligent Text Processing; www.CICLing.org), which is a narrow domain-oriented conference [5]. The document collection consisted of 48 abstracts (40 KB of text). After indexing, the domain dictionary contained approximately 390 words. Though this is a small collection, our research is preliminary, our goal being to attract attention to the problem and to the possible ways of its solution. For this, one does not need a very large collection. On the other hand, this allowed for careful manual classification and detailed evaluation of the results.

A human expert manually classified the papers of the Conference into 4 and 11 classes, which were, according to the expert's judgment, natural. This can be explained as follows. Classification into 2 classes was not interesting: the binary situation may be effectively analyzed by the corresponding methods. Classification into 3, 5, ..., 10 classes was not considered by the expert to be balanced. This means that in these cases the classes proved to belong to different levels of hierarchy if the hierarchy-based clustering would be applied. And classification into more than 11 classes gave very small groups.

Here are the four categories of papers selected by the expert (the titles are given only for the reader's convenience):

- Linguistic (semantics, syntax, morphology, parsing),
- Ambiguity (word sense disambiguation, anaphora, tagging, spelling),
- Lexicon (lexicon and corpus, text generation),
- Text processing (information retrieval, summarization, text classification).

The selected categories (classes) are rather fuzzy. For example, the intersection of vocabulary for the documents from the most different second and forth groups was about 70%. This implies that the selected domain is rather narrow.

**Validity and usability of clustering.** Quality of clustering is evaluated using various objective criteria based in the relations within and between the clusters. The procedure of evaluating the clustering quality is called cluster validation. For testing cluster validity we used so-called index of expected density of clustering [20]:

$$\bar{\rho}(C) = \sum_{i=1}^{k} \frac{|V_i|}{|V|} \cdot \frac{w(G_i)}{|V_i|^{\theta}}, \quad \textit{where} \quad |V|^{\theta} = w(G), \tag{5}$$

where $G$ is a given graph of connections between the objects, $G_i$ is its subgraph, $V$ is the total number of the objects in the graph, $V_i$ is the number of the objects in $i$-th subgraph, $w(\cdot)$ is the total weight of the edges in the graphs or subgraphs, $C$ stands for a given clustering, and $G_i$ corresponds to the $i$-th cluster. Higher values of $\bar{\rho}$ mean better clustering.

The correspondence between the results of automatic and human clustering can be evaluated by various measures. The procedure of evaluating is called cluster usability estimation. We use the $F$-measure in the form presented in [4]:

$$F = \sum_{i=1,\dots l} \frac{|C_i^*|}{V} \max_{j=1,\dots,k} \{F_{i,j}\}; \quad F_{i,j} = \frac{2\, prec\,(i,j)\, rec\,(i,j)}{prec\,(i,j) + rec\,(i,j)}, \tag{6}$$

where $\{C^*_1, \dots, C^*_l\}$ stand for the human classification, $\{C_1, \dots, C_k\}$ for the clusters obtained by the algorithm, $prec(i,j)$ is the precision of the cluster $j$ with respect to the class $i$: $|C_j \cap C^*_i| \,/\, |C_j|$; $rec(i,j)$ is the recall of cluster $j$ with respect to the class $i$: $|C_j \cap C^*_i| \,/\, |C^*_i|$.

Stein *et al.* [20] showed that $\bar{\rho}$-index correlates well with $F$-measure. Thus to provide a good usability of results, $\bar{\rho}$-index can be calculated without participation of human experts.

### 3.2 Experiments

**Experimental setting.** Following the traditional approach, we indexed all the words according to well-known *tf* and *tf-idf* measures, where *tf* stands for term frequency and *idf* for inverse document frequency [15].

In the suggested approach, we used all words selected by the rule (1) discussed above, and then clustered them with MajorClust method. This gave 14 clusters of keywords, which were semantically weighted by the formula (2). Before this procedure, the weakest connections were eliminated from the connection graph.

In all our experiments, the standard cosine similarity measure was used to evaluate both the similarity between documents and the similarity between keywords.

**Testing the grouping of keywords.** The goal of the first series of experiments was to compare the traditional and suggested approaches to indexing. We also checked the sensibility of the results to the change of the document set. Since we had the reference classification provided by a human expert, we could evaluate the quality of clustering using $F$-measure.

The number of document clusters to be constructed was fixed at 4. The MajorClust method revealed confirmed that this is the natural number of documents clusters.

We conducted our experiments with two document sets: the original one, containing all 48 abstracts, and a reduced one, containing 75% of the whole set, i.e., 36 documents. Because of reducing the number of documents, the vocabulary used for clustering procedure changed. The results are presented in Table 1. Scaling stands for the correction of document coordinates according to (4).

These results show that the suggested approach gives better and more stable results with respect to changing the document set. Besides, one can see that using *idf* factor reduces the stability of the results. The possible explanation of this is a high level of randomness of word occurrences in texts.

**Table 1.** F-measure for comparison of ways of indexing.

| Indexing | Scaling | 48 abstracts | 36 abstracts |
|---|---|---|---|
| *Tf-idf* | No | 0.56 | 0.49 |
| *Tf* | No | 0.56 | 0.51 |
| Grouping | Yes | 0.64 | 0.58 |

**Testing logarithmic measure.** The goal of the second series of experiments was to demonstrate the advantage of using logarithmic scaling in cosine similarity measure. Here we used the whole document set.

**Table 2.** *F*- measure for evaluation of scaling.

| Indexing | Scaling | Without scaling |
|---|---|---|
| *tf-idf* | 0.64 | 0.56 |
| Grouping | 0.64 | 0.58 |

The results show that logarithmic scaling improves clustering both for traditional approach to indexing and for the suggested one.

**Testing methods and model complexity.** In our last series of experiments, we tested the clustering methods under different number of expected classes. We considered the *K*-medoid, NN- and MajorCluster methods. The number of classes was equal 4 and 11. Since the MajorCluster determines the number of clusters automatically, it was used only one time. The quality of clustering was evaluated by *F*-measure and $\bar{\rho}$ index. The results are presented in Table 3. Note that the values for *F* and $\bar{\rho}$ vary in the range between 0 and 1.

**Table 3.** *F*- measure and $\bar{\rho}$ -index for different cluster number.

| | 4 clusters | | 11 clusters | |
|---|---|---|---|---|
| Method | *F* | $\bar{\rho}$ | *F* | $\bar{\rho}$ |
| *NN* method | 0.64 | 0.71 | 0.46 | 0.42 |
| MajorClust | 0.64 | 0.71 | – | – |
| *K*-medoid | 0.56 | 0.67 | 0.49 | 0.39 |

The results demonstrate a quite good stability of clustering with 4 clusters: both contradictory methods gave the same results. Worse quality of clustering on 11 clusters can be explained as follows. By joining keywords, we improve stability of the results, but simultaneously lose some information related to the semantics of individual keywords. This means that by clustering keywords we limit the structure complexity (number of classes), which can be revealed from the data using given set of keyword clusters. Reduced value of the validity index supports such a hypothesis.

It is easy to see that changing of expected density index reflects the change of *F*-measure. However, to consider their correlation, one should conduct experiments on different data sets and with different number of clusters. For RCV1 document collection, this was elaborated by Stein *et al*. [20].

## 4 Conclusions and Future Work

**Conclusions of the experiments.** Nowadays the problem of clustering abstracts is important for handling documents in digital libraries of scientific information, where the most part of the data is presented in the form of abstracts. We have suggested a technique for clustering such information, which consists in grouping indexes and using logarithmic scaling in document similarity measure. Our experiments with abstracts show its advantage in comparison with traditional approaches.

**Proposal on open access to full text document images.** Clustering only abstracts one cannot achieve as good results as when clustering full text papers. To facilitate the work of search engines, both in search and in clustering the search results, especially in context of the Semantic Web effort, we propose that digital libraries and Internet repositories *provide open access to document images of the papers.* A document image is a vector of word frequencies, which can be restricted to a small list of keywords extracted from the whole document collection. This does not violate the copyright laws because it is impossible to recover the full text of the paper from such a document image. This proposal was originally suggested by Makagonov [12] and is now under consideration in the Library of the Mixteca University.

**Future work.** This is a preliminary paper. In the future, we plan to use WordNet and other ontology-related techniques for grouping semantically similar keywords and compare the results with those obtained in this paper. We plan to consider the hypervolume clustering criterion [6] to improve cluster validity. We also plan to apply our techniques for very large medical database of Czech Ministry of Healthcare, in cooperation with our Czech colleagues from Masaryk university. Finally, we plan to evaluate our methods on the 20newsgrous collection.

We will also test the methods of assessing the cluster quality taking into account the possibilities for the division of a given test set into classes. This depends on the intersection of the vocabulaty of the classes.

## References

1. Alexandrov, M., X. Blanco, P. Makagonov. Testing Word Similarity: Language Independent Approach with Examples from Romance. In: F. Meziane *et al*. (Eds.) *Natural Language Processing and Information Systems*, Springer, LNCS N 3136, 2004, pp. 223–234.
2. Alexandrov, M., A. Gelbukh, P. Rosso. Clustering Very Short Documents based on Grouping Keywords. *Abstracts of the 30-th Latin-American Conf. on Informatics*, Univ. Edition, Peru, 2004, p. 133.
3. Baeza-Yates, R., B. Ribero-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.
4. Eissen, S.,M., B. Stein. Analysis of Clustering Algorithms for Web-based Search. In: *Practical Aspects of Knowledge Management*, LNAI N 2569, Springer, 2002, pp.168–178.
5. Gelbukh, A., (ed.). *CICLing-2002*, *Comput.Linguistics and Intelligent Text Processing*. LNCS N 2276, Springer-Verlag, 2002; www.CICLing.org.
6. Hardy, A., P. Andre. An investigation of nine procedures for detecting the structure in a data set. In: *Advances in data science and classification*, Springer, "Studies in Classification, Data Analysis and Knowledge Organization," 1998, pp. 29–36.
7. Hartigan, J. *Clustering Algorithms.* Wiley, 1975.
8. Hynek, J, K. Jezek, O. Rohlikm. Short Document Categorization – Itemsets Method. In: *PKDD-2000*, Springer, LNCS N 1910, 2000, 6 pp.

9. Kang Bo-Y., H-J. Kim, S-j. Lee. Performance Analysis of Semantic Indexing in Text Retrieval. In: A. Gelbukh (Ed.), *Computational Linguistics and Intelligent Text Processing*, CICLing-2004, LNCS N 2945, Springer-Verlag, 2004, pp. 433–436.
10. Makagonov, P., M. Alexandrov, K. Sboychakov. Keyword-based technology for clustering short documents. In: *Selected Papers. Computing Research*, CIC-IPN, Mexico, 2000, pp. 105–114.
11. Makagonov, P., M. Alexandrov, K. Sboychakov. A toolkit for development of the domain-oriented dictionaries for structuring document flows. In: *Data Analysis, Classification, and Related Methods*, Studies in classification, data analysis, and knowledge organization, Springer, 2000, pp. 83–88.
12. Makagonov, P., M. Alexandrov, A. Gelbukh. Clustering Abstracts instead of Full Texts. In: *Text, Speech, Dialog*, LNAI N 3206, Springer, 2004, pp. 129–135.
13. Manning, D., C. and H. Schutze. *Foundations of statistical natural language processing*. MIT Press, 1999.
14. Porter, M. An algorithm for suffix stripping. *Program*, 14, 1980, pp. 130–137.
15. Salton, G., C. Buckley. *Term-weighted approaches in autometic retrieval*. Information Processing in Management, v.24, 1988, N 5, pp. 513–523.
16. Solomon, G. Data dependent methods of cluster analysis. In: *Classification and Clustering*, Academic Press, 1977, pp. 129–147 (Russian version).
17. Stein, B., O. Niggemann. On the Nature of Structure and its Identification. In: *Graph-Theoretic Concepts in Computer Science*. LNCS, N 1665, Springer, 1999, pp.122–134.
18. Stein, B., S. M. Eissen. Document Categorization with MajorClust. In: *Proc. 12th Workshop on Information Technology and Systems*, Tech. Univ. of Barcelona, Spain, 2002, 6 pp.
19. Stein, B., S. M. Eissen. Automatic Document Categorization: Interpreting the Performance of Clustering Algorithms. In: *Proc. 26th German Conf. on Artificial Intelligence*, LNCS N 2821, Springer, 2003, pp. 254–266.
20. Stein, B., S. M. Eissen, F. Wissbrock. On Cluster Validity and the Information Need of Users. In: *Proc. 3-rd IASTED Intern. Conf. on Artificial Intelligence and Applications* (AIA'03), Acta Press, 2003, pp. 216–221.
21. Strzalkowski, T. (Ed.). *Natural Language and Information Retrieval*. Kluwer Academic Publishers, 1999.
22. Zizka, J., A. Bourek. *Automated Selection of Interesting Medical Text Documents by the TEA Text Analyzer*. In: A. Gelbukh (Ed.), Computational Linguistics and Intelligent Text Processing, CICLing-2002, LNCS N 2276, Springer-Verlag, 2002, pp. 402–404.

# Named Entity Recognition for Web Content Filtering*

José María Gómez Hidalgo,
Francisco Carrero García, and Enrique Puertas Sanz

Universidad Europea de Madrid, Villaviciosa de Odón, 28670, Madrid, Spain
{jmgomez,francisco.carrero,epuertas}@uem.es
http://www.esi.uem.es/~jmgomez

**Abstract.** Effective Web content filtering is a necessity in educational and workplace environments, but current approaches are far from perfect. We discuss a model for text-based intelligent Web content filtering, in which shallow linguistic analysis plays a key role. In order to demonstrate how this model can be realized, we have developed a lexical Named Entity Recognition system, and used it to improve the effectiveness of statistical Automated Text Categorization methods. We have performed several experiments that confirm this fact, and encourage the integration of other shallow linguistic processing techniques in intelligent Web content filtering.

## 1 Introduction

In the recent years, we have witnessed an impressive growth of on-line information and resources. The self-regulating nature of Web publishing, along with the ease of making information available on the Web, has allowed that some publishers make offensive, harmful or even illegal contents present in Web sites across the world. This fact makes the use of filtering and monitoring systems a necessity in educational environments, and in the work place, to protect children and prevent Internet abuse.

There is a number of filtering solutions available in the market, including commercial products like CyberPatrol or NetNanny, and open source systems like SquidGuard or DansGuardian. According to in-depth evaluations of these products (e.g. the one performed in the European Project NetProtect [1]), their filtering effectiveness is limited by the use of simple techniques, like URL blocking, or keyword matching. There is the need of more sophisticate and intelligent approaches to increase effectiveness of filtering solutions, and thus, to improve children's protection and Internet abuse prevention.

Our goal is to investigate to what extent shallow linguistic processing can improve the effectiveness of current filtering software. This kind of analysis is

* The work described in this paper is partly funded by the European Commission under the Safer Internet Action Plan, contract POESIA - 2117 / 27572. POESIA stands for Public Open-source Environment for a Safer Internet Access, and more information about it can be fount at http://www.poesia-filter.org.

A. Montoyo et al. (Eds.): NLDB 2005, LNCS 3513, pp. 286–297, 2005.
© Springer-Verlag Berlin Heidelberg 2005

suitable for that Web pages that have been considered suspicious by simpler techniques, being the majority of Web pages tagged as harmful or harmless by faster and more simple methods. We focus on Spanish language pornographic Web pages, but the methods discussed here can easily extended to other European languages.

In this work, we are concerned with the possibility of using Named Entity Recognition to improve statistical classification of pornographic Web pages. According to recent evaluations ([2]), there are a number of well known and relatively effective methods for detecting Named Entities (NEs) in running text. Some NEs can be highly indicative of pornographic content (e.g. names of famous porn stars, or manufacturers of sex toys), while others suggest harmless content (e.g. names of politicians, corporations, or locations).

We test this intuition by conducting a set of experiments that compare a statistical pornographic detection approach based on state-of-the-art Text Categorization techniques, with an improvement of this one that considers also NEs as attributes of Web pages. The NEs are detected using an simple approach based on lexical evidence and learning. The results of our experiments show that the usage of NE Recognition (NER), and thus, of shallow linguistic analysis, can improve the accuracy of otherwise highly accurate methods.

## 2 Web Content Filtering

In this section, we review the main approaches to Web content filtering, focusing specially on those that use intelligent content analysis based on text classification. Also, we present our basic approach for pornographic Web content detection based on Automated Text Categorization state-of-the-art methods.

### 2.1 Review of Recent Work

The Web Content Filtering (WCF) approaches have been classified in four major groups [3]:

- Self or third party ratings, specially the usage of Platform for Internet Content Selection (PICS) or Internet Content Rating Association (ICRA) ratings. Authors or reviewers label Web pages according to several types of content and levels, which are used by the filtering system to allow or block the pages according to the settings defined by the user or administrator. Unfortunately, only a small fraction of Web pages are labeled, and authors can inadvertently (or intentionally) label their pages with incorrect tags.
- Uniform Resource Locator (URL) listing, that is, maintaining a list of blocked and/or allowed web sites. A Web page is blocked if its URL contains a blocked URL, or its outgoing links point to blocked URL addresses. These kinds of lists can be automatically or manually built, but they are difficult to keep updated, and do not account for domain aliasing.

- Keyword matching, in which a set of indicative keywords or key-phrases ("sex", "free pics") are manually or automatically derived, form a set of pornographic Web pages. A Web page is blocked if the number or frequency of keywords occurring in it, exceeds a predetermined threshold. This approach is prone to over-blocking, that is, blocking safe Web pages in which these keywords occur (e.g. sexual health, etc.).
- Intelligent content filtering, which involves a deeper understanding of the semantics of text and other media items (specially pictures), by using linguistic analysis, machine learning, and image processing techniques. The heavy cost of building linguistic analyzers and image processing components, their domain dependence (e.g. technique for detecting nudes are quite different to those for recognizing Nazi symbols), and the delay caused by in-depth analysis, limit the applicability of these techniques.

The first three approaches, widely used in current filtering solutions, have proved quite ineffective, and have serious limitations [1]. We argue that intelligent content analysis is feasible, as far as the system design deals with delay issues, and linguistic and image processing are kept as shallow as possible. In particular, the project POESIA [4, 5] is designed for having two levels of filtering: Light filtering, for those Web pages that are not suspicious, or clearly pornographic; and heavy filtering, for those Web pages in which light filters are not able to give a clear judgment. Linguistic and image processing techniques in the light filters are very limited and efficient, while heavy filter use more advanced (but shallow, anyway) methods, giving a more accurate but delayed answer.

We focus here on shallow linguistic processing of Web pages textual items, in order to take a sensible decision about the pornographic orientation of the pages, when simpler methods fail. Our intuition is that, while Automated Text Categorization can be very effective on the task, it might be improved with shallow text analysis involving e.g. POS-tagging, Noun Phrase chunking, and in particular, Named Entity Recognition. We discuss the basic Text Categorization approach in the next section.

### 2.2 Web Content Filtering as Text Categorization

Automated Text Categorization (ATC) is the assignment of text documents to predefined categories. Text documents are usually news items, scientific reports, e-mail messages, Web pages, and so. Categories are often thematic, and include library classifications (e.g. the Medical Subject Headings), keywords in Digital Libraries, personal e-mail folders[1], Web directory categories (like Yahoo!'s), etc. Automated Text Categorizers can be built by hand (e.g. by writing rules for e-mail messages filing in personal folders), or they may be constructed automatically, by learning a text classifier on a set of manually labeled documents. This latter, learning-based, approach has become dominant, and current techniques allow building ATC systems as accurate as human experts in a domain [6].

[1] Some e-mail folders of an average e-mail user may be not thematic at all, but based e.g. on the sender (for instance, my brother's messages).

Pornography detection has been approached as learning-based ATC in several recent works, including [3, 7–10], and ours [5]. From these works, we can model pornography detection as a 2-class learning problem: learn a classifier that decides if Web page is pornographic or not. In the learning phase, given two sets of pornographic (P) and safe (S) Web pages (the *training collection*), the following steps are given:

1. Each Web page in P or S is processed to extract the text it includes (pieces of text inside <TITLE>, <H1>, <P>, <META>[2]; or, just all tags are stripped out), defining its text content. The text is tokenized into words, which may be stemmed and/or stop listed (ignoring those occurring in a function word list), producing a list of text tokens.
2. Each page is represented as a term-weight (or attribute-value) vector, being the terms the previous text tokens. The weights can binary (a token occurs in the Web page, or not), Term Frequency (number of times the token occurs in the page), TF.IDF (the previous one times the Inverse Document Frequency), Relative Term Frequency, etc. [6].
3. Optionally, a number of tokens is selected according to a quality metric like Information Gain or $\chi^2$ [11]. This step allows to reduce the dimensionality of the problem, speeding up learning and even increasing accuracy. The resulting set of tokens is the final *lexicon*.
4. Finally, a *classifier* is induced using a training algorithm over de set of training vectors and its associated class labels. Algorithms used in this problem include the probabilistic Naive Bayes algorithm [7] and Bayesian Networks [8], variants of lazy learning [9, 10], semi-supervised Neural Networks [3], and linear Support Vector Machines [5].

The two first steps define the text representation model, which in this case is often called the *bag of words* model. It corresponds to the traditional Vector Space Model in Salton's work [12]. In other works focused on Web page categorization according to thematic classes, there has been propose to enrich the text in the page with the text of nearby Web pages in the link graph structure (from incoming or outgoing links) [13].

The classification phase involves, given a new Web page which class is not known, its representation as term-weight vector similar to those in the training collection, and its classification according the model generated in the learning phase. This phase must be extremely efficient, avoiding long delays in Web pages delivery when they are allowed (classified as safe).

### 2.3 Our Text Categorization Approach

The previous approach has been proved quite successful in the literature, but current experimental results are not perfect, and filters often miss-classify web pages in which there is nearly no text, or those concerned with sexual education.

[2] Usually, restricted to attributes NAME and CONTENT.

Also, experimental results may be different to those obtained in operational environments, which have been not test yet. We believe that this model can be improved in the uncertain cases, by using linguistic techniques that provide a more meaningful insight of the page topic.

As a pilot study, we have included a NER system in our baseline ATC approach. The details of our method are:

- Text is extracted from training Web pages with an HTML parser.
- The extracted text is tokenized, each token converted to lowercase, and stemmed using an Spanish stemmer.
- We select the text tokens with an Information Gain score (that is, those stems that provide any indicative information in the training collection). The resulting tokens are the vocabulary.
- Each training Web page is represented as a term-weight vector, in which terms are vocabulary tokens, and their weights are their Term Frequency in the page.
- We learn a linear classifier (a linear function of the weights of the stems in the vocabulary) using linear Support Vector Machines[3].

We provide details on the Web page training collection in the section 4.1. Provided enough and representative training data, this approach leads to fast and accurate text classifiers in most cases, letting anyway some space for effectiveness improvements for the case of difficult Web pages.

## 3 Spanish Named Entity Recognition

In this section, we review the recent work in Spanish and Language Independent NER, along with our approach to the problem. We also provide effectiveness results on standard data collections, and how we have integrated the NER system in the previous ATC method.

### 3.1 Recent Work in Spanish NER

NER has been considered as an important task in the wider Information Extraction field, and it is nowadays fully integrated in typical text analysis tasks for learning-based Information Extraction applications [14]. It has been also the focus of recent Computational Natural Language Learning (CoNLL) Shared Task competitions (2002, 2003) [2], giving the area an important impulse. Currently, and given enough training data, it is possible to build a NER system that reaches high levels of accuracy (e.g. an $F_1$ value over .80 for Spanish, and near .90 for the English language).

[3] These steps have been performed by using the following open-source packages: the HTMLParser (`http://htmlparser.sourceforge.net/`), the SnowBall Spanish stemmer (`http://snowball.tartarus.org/`), and the WEKA Machine Learning library (`http://www.cs.waikato.ac.nz/~ml/weka/`).

Most top performing NER systems in CoNLL Shared Task competitions[4] follow a learning approach, sometimes enriched with the use of external resources as e.g. gazetteers. The task is approached as a Term Categorization problem, in which each word must be tagged as the beginning (B) of a Named Entity, an inner word in a Named Entity (I), or as other (O). The types of entities addressed in CoNNL competitions are persons, organizations, locations and miscellanea. As an example, the expression "Robinson lived in a island near South America" should be tagged as "Robinson/B-PER lived/O in/O a/O island/O near/O South/B-LOC America/I-LOC".

We discuss the top performing system presented for the Spanish language NER competition [15], as it illustrates the dominant method in the field. In this work, the NER is a two level learning system, the first level detecting the limits of a NE, and the second finding out its type. Its main characteristics are:

- The set of features for each word is defined in terms of the context of the word, including the word itself, its Part-Of-Speech (POS), orthographic features of the word (involving capitalization, hyphenation and others), the form of the words in a fixed length contextual window (lexical features), left predictions, and others.
- The learning method for both levels is the meta-learner Adaboost applied to small fixed-depth decision trees. This meta-learner has the property of improving a base learner, by iteratively focusing on those examples (words) that are incorrectly labeled by the trees learned in previous iterations.
- Two external resources are also used, a gazetteer that includes a number of NEs for Spanish, not see in the training phase, and a set of hand-crafted trigger words. These resources lift accuracy in a 2% for the type of entity classifier.

This work both shows the main characteristics of current learning-based NERs, and it demonstrates that high levels of accuracy can be achieved in the Spanish NER task. It has also partly (along with [16]) guided the design of our lexical NER method, which has been designed as a *knowledge poor* approach for the pornographic Web content detection pilot study.

### 3.2 A Lexical Spanish NER

Our NER system is designed to use only the most reliable features in the surrounding context of the target word, given the unstructured nature of Web pages (quite different from news items, used in previous experiments). In fact, we call our NER *lexical* because we found by trial and error that only this kind of information can be robustly extracted from Web pages text. In particular, we consider a set of features that includes: binary orthographic features for the words in a fixed-length window surrounding the target word, a list of frequent words and

[4] These can be compared in the tasks web pages, for CoNNL 2002 (`http://cnts.uia.ac.be/conll2002/ner/`) and 2003 (`http://cnts.uia.ac.be/conll2003/ner/`).

punctuation symbols in the window, and the predicted class for the previous words in the window.

The number of words (lexical items) considered, the width of the window, the binary or numeric nature of attribute values, and the utilization or not of previous tags, are parameters of our system. We have tested a wide number of parameter settings, in a wrapper approach: given a configuration of parameters (window size = 2, etc.), we test its accuracy by 5-fold cross validation on the training set, by using a decision tree learner (C4.5), and assessing its classification accuracy.

The best results have been obtained using 44 attributes: in a ±2-size window, if the word has an initial capital letter (5 attributes, one per word in the window), if the word is all uppercased (5 attributes), if the target word is only a capital letter (1 feature), a capital letter or a period (1 feature), starts with one capital letter or it is a period (1 feature), and it uppercased or it is a period (1 feature). Last 30 features are the position of each of the 30 most frequent tokens (either words, lowercased, or punctuation marks) in the window, if they occur within the window.

We have after tested a representative range of learning algorithms on this feature configuration, including a Naive Bayes classifier, the C4.5 decision tree learner itself, Adaboost applied to C4.5 (ABC4.5), linear Support Vector Machines (SVM), and the lazy learner $k$-Nearest Neighbors with $k = 1, 3, 5$ values. The results of these experiments, along with the evaluation of the selected learners in the CoNLL experimental framework, are discussed in the next section.

### 3.3 Experimental Results on the CONLL Framework

Since in this pilot study, we are interested on detecting NEs, but not classifying them according to its type, we present results only for B, I and O tags, and for entire NEs.

First we present the three top performing learners results, obtained by 5-fold cross validation on the training set. In the Table 1, we show the $F_1$ measure for the three types of tags (B, I and O), and the overall accuracy. $F_1$ is a kind of average of recall (the proportion of items detected over the real number of items) and precision (the proportion of correctly retrieved items over the number of retrieved items). The accuracy (Acc) is the number of correct items over the total number of items.

**Table 1.** Results on training data for the top performing learners.

| Algorithm | $F_1$/B | $F_1$/I | $F_1$/O | Acc |
|---|---|---|---|---|
| C4.5 | 0.889 | 0.825 | 0.989 | 0.972 |
| ABC4.5 | 0.886 | 0.831 | 0.988 | 0.972 |
| SVM | 0.886 | 0.815 | 0.988 | 0.971 |

The results shown demonstrate that the three learners are roughly equivalent, in terms of effectiveness. We are concerned with the less frequent B and I tags, that mark NEs in the texts. For these tags, ABC4.5 is slightly better than C4.5, but the decrease in speed (C4.5 with boosting is two orders of magnitude slower than C4.5 alone) does not deserve using it.

This analysis is confirmed by the results obtained on the CoNLL test data, using the standard evaluation scripts for the task. In the Table 2, we show the accuracy (Acc), recall (Re), precision (Pr), and $F_1$ scores for the C4.5, ABC4.5 and SVM learners. On this dataset, the C4.5 learner performs a bit better than ABC4.5.

**Table 2.** Results on CoNLL test data for the top performing learners.

| Algorithm | Acc | Re | Pr | $F_1$ |
|---|---|---|---|---|
| C4.5 | 0.969 | 0.816 | 0.818 | 0.817 |
| ABC4.5 | 0.969 | 0.800 | 0.821 | 0.810 |
| SVM | 0.969 | 0.806 | 0.812 | 0.809 |

We must note that the results of our NER system are not comparable with those by participants in the CoNLL Shared Tasks competitions, because they are forced to predict the type of the NE detected. This fact makes their results better than ours, because a decrease of effectiveness is expected on this phase. Also, we have detected a subtle tendency of our NER to classify any starting token in a sentence as a NE, which should be corrected in an operational implementation of the overall filtering approach.

On the basis of these experiments, we use a NER system based on the 44 attribute vector representation described above, and the C4.5 decision tree learner, for our following work.

### 3.4 Integrating NER in ATC-Based Web Content Filtering

We have performed a straightforward integration of the NER method in the ATC-based Web page filtering approach. We have applied the NER system to the training collection Web pages text, and added to the lexicon the extracted NEs with an Information Gain score over 0. For these tokens, we also make use of Term Frequency weights.

The NER system has been trained on the training collection of the CoNLL Shared Task competition for Spanish (2002). However, it has been applied to the text extracted from Web pages, violating a basic assumption in Machine Learning: test (or working) data must resemble training data. We view this as a *domain transfer* of the NER system, and given this violation, we do not expect it to be as effective as it is on news items.

## 4 Experiments

In this section, we describe our data collection, and the experiments we have performed on it, in order to confirm that shallow linguistic processing (specifically, NER) can improve statistical ATC methods.

### 4.1 Data Collection

The data collection used in our experiments is the POESIA project Spanish Text Corpus. This corpus contains 6,463 pornographic Web pages and 29,133 non-pornographic Web pages, in HTML format, and they have been collected the following way:

1. An initial set of URL addresses have been obtained from the Spanish section of the Open Directory Project Web page[5], containing a list of around 1,000 pornographic URLs and 100,000 non-pornographic URLs. These latter list has been randomly sub-sampled to get around 5,000 URLs.
2. An initial corpus has been built by downloading those Web pages with less than 10 seconds answer. These Web pages have been filtered to delete frame based and Error 404 Web pages. Also, the files have been processed to get in-site links, providing a second set of URLs. The resulting lists have been sub-sampled, and the process has been repeated one more.

Essentially, current contents of the corpus include front to second level Web pages from a representative sample of pornographic and non-pornographic Web sites.

For our experiments, we have divided the corpus into a training set containing 2/3 of the corpus Web pages, and a test set with the remaining pages. Since the HTML parser fails to extract text on strongly unstructured Web pages, a portion of the pages containing less than 10 bytes of text have been removed from training and testing sets. This leads to 4,188 pornographic and 18,446 safe training Web pages, and 2,094 pornographic and 9,200 safe test Web pages.

After performing text extraction, tokenization, and stemming, we have collected a set of 17,859 word stems with an Information Gain score greater than 0 in the training collection. Also, we have applied the NER system to the text in training Web pages, deriving a set of 8,491 NEs with an Information Gain score over 0. The combined lexicon has 26,350 units, being NEs a 32.22% of them. However, there are only 17 NEs among the 200 top scoring units, and NEs are often section names with capitalized words (e.g. “Fotos” – the pictures section of the Web site).

### 4.2 Results and Analysis

The results of our experiments on Spanish pornographic Web content detection are summarized in the Table 3. In this table, we show the performance of linear

[5] Available at `http://dmoz.org`.

**Table 3.** Results of SVM on the three kinds of lexicon used.

| Lexicon | Re/P | Pr/P | $F_1$/P | Re/S | Pr/S | $F_1$/S | Acc |
|---|---|---|---|---|---|---|---|
| Stems | 0.867 | 0.983 | 0.921 | 0.997 | 0.971 | 0.983 | 0.972 |
| NEs | 0.812 | 0.960 | 0.880 | 0.992 | 0.959 | 0.975 | 0.958 |
| Both | 0.891 | 0.982 | 0.934 | 0.996 | 0.976 | 0.986 | 0.976 |

Support Vector Machines on three text representation models: one using only word stems, one using only NEs, and one using both. For each of the classes, pornographic Web pages (P) and safe Web pages (S), the recall (Re), precision (Pr) and $F_1$ scores are shown, along with the overall accuracy (Acc).

The results of these experiments are encouraging. It is clear that NEs are even good indexing units by themselves, although this may be due in part to the fact that many of them are ordinary words with some kind of capitalization (that is, they correspond to NER over-detection mistakes). The combination of both sources of evidence, stems and NEs, clearly outperforms NEs in isolation, and noticeably improves an stem based representation.

In a detailed analysis, the stem-based approach miss-classifies 31 safe Web pages, and 279 pornographic Web pages, while the combined approach miss-classifies 34 safe Web pages, and 229 pornographic Web pages. There is a relative improvement of the accuracy on pornographic pages, at the expense of a affordable decrease of performance on safe pages.

### 4.3 Additional Considerations

In an operational environment, a classifier like those proposed here must be able to determine the class (porn or safe) of a high number of Web pages per minute. The stem based representation is easy and quick to compute, and the linear SVM perform linearly on the number of attributes (terms). This configurations warrants fast response times, even in the Java language, according to our experience in the POESIA project.

An important concern is if shallow linguistic analysis in general, and NER in particular, may slow down the filtering operation, even making it unfeasible. This point has been addressed in POESIA by using two level filtering. A light filter, based on statistical methods (e.g. SVM) and simple text representations (e.g. word stems) is first called, allowing a quick answer for most of the requests. If this filter classifies the Web page as "unsure", it is passed to a heavy filter that performs a deeper (and slower) processing of the request. According to our experience, this filter can employ even seconds to process a request, implying a delay for the user. However, this delay is accepted by most educational institutions, where Internet resources are shared by hundreds to thousands of students, and there is always a delay due to bandwidth limitation. So in short, the framework for shallow linguistic analysis of Web pages proposed here, is acceptable in operational environments.

It is remarkable that commercial filters tend to block health Web pages, according to the study for the Kaiser Family Foundation [17]. Our model is specially suitable for this kind of contents, on which the cost of the delay is acceptable if classification accuracy is significantly improved.

## 5 Conclusions and Future Work

In this paper, we argue that shallow linguistic analysis in general, and Named Entity Recognition in particular, can be used to improve the effectiveness of text classification in the framework of intelligent Web content filtering. We have implemented a lexical NER system, and used it to demonstrate how NEs discovered on a training phase, can enrich a traditional, stem based vocabulary, leading to a more accurate filter.

The results of our experiments are encouraging, and motivate us to improve our NER system, and to include more shallow linguistic processing techniques (like e.g. text chunking) in our Web content filtering method. We believe that filtering methods can benefit from a deeper understanding of the meaning of text in Web pages.

## References

1. Brunessaux, S., Isidoro, O., Kahl, S., Ferlias, G., Rotta Soares, A.: NetProtect report on currently available COTS filtering tools. Technical report, NetProtect Deliverable NETPROTECT:WP2:D2.2 to the European Commission (2001) Available: `http://www.netprotect.org`.
2. Roth, D., van den Bosch, A., eds.: Proceedings of CoNLL-2002, Taipei, Taiwan, Association for Computational Linguistics, Special Interest Group on Natural Language Learning (2002)
3. Lee, P., Hui, S., Fong, A.: A structural and content-based analysis for web filtering. Internet Research **13** (2003) 27–37
4. Gómez, J., de Buenaga, M., Carrero, F., Puertas, E.: Text filtering at POESIA: A new internet content filtering tool for educational environments. Procesamiento del Lenguaje Natural **29** (2002) 291–292
5. Hepple, M., Ireson, N., Allegrini, P., Marchi, S., Montemagni, S., Gómez, J.: NLP-enhanced content filtering within the POESIA project. In: Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC 2004). (2004)
6. Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys **34** (2002) 1–47
7. Chandrinos, K.V., Androutsopoulos, I., Paliouras, G., Spyropoulos, C.D.: Automatic Web rating: Filtering obscene content on the Web. In Borbinha, J.L., Baker, T., eds.: Proceedings of ECDL-00, 4th European Conference on Research and Advanced Technology for Digital Libraries, Lisbon, PT, Springer Verlag, Heidelberg, DE (2000) 403–406 Published in the "Lecture Notes in Computer Science" series, number 1923.
8. Denoyer, L., Vittaut, J.N., Gallinari, P., Brunessaux, S., Brunessaux, S.: Structured multimedia document classification. In: DocEng '03: Proceedings of the 2003 ACM symposium on Document engineering, ACM Press (2003) 153–160

9. Du, R., Safavi-Naini, R., Susilo, W.: Web filtering using text classification. In: Proceedings of the 11th IEEE International Conference on Networks, Sydney, IEEE (2003) 325–330
10. Su, G.Y., Li, J.H., Ma, Y.H., Li, S.H.: Improving the precision of the keyword-matching pornographic text filtering method using a hybrid model. Journal of Zhejiang University SCIENCE **5** (2004) 1106–1113
11. Yang, Y., Pedersen, J.: A comparative study on feature selection in text categorization. In: Proceedings of ICML-97, 14th International Conference on Machine Learning. (1997)
12. Salton, G.: Automatic text processing: the transformation, analysis, and retrieval of information by computer. Addison Wesley (1989)
13. Ghani, R., Slattery, S., Yang, Y.: Hypertext categorization using hyperlink patterns and meta data. In Brodley, C., Danyluk, A., eds.: Proceedings of ICML-01, 18th International Conference on Machine Learning, Williams College, US, Morgan Kaufmann Publishers, San Francisco, US (2001) 178–185
14. Zhang, T., Damerau, F., Johnson, D.: Text chunking based on a generalization of winnow. J. Mach. Learn. Res. **2** (2002) 615–637
15. Carreras, X., Màrques, L., Padró, L.: Named entity extraction using adaboost. In: Proceedings of CoNLL-2002, Taipei, Taiwan (2002) 167–170
16. Chieu, H., Ng, H.: Named entity recognition: A maximum entropy approach using global information. In: Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002). (2002) 190–196
17. Richardson, C., Resnick, P., Hansen, D., Holly A. Derry, Rideout, V.: Does Pornography-Blocking Software Block Access to Health Information on the Internet? Journal of the American Medical Association **288** (2002) 2887–2894

# The Role of Word Sense Disambiguation in Automated Text Categorization

José María Gómez Hidalgo[1],
Manuel de Buenaga Rodríguez[1], and José Carlos Cortizo Pérez[2]

[1] Universidad Europea de Madrid
Villaviciosa de Odón, 28670, Madrid, Spain
{jmgomez,buenaga}@uem.es
http://www.esi.uem.es/~{jmgomez,buenaga}
[2] AINet Solutions
Fuenlabrada, 28943, Madrid, Spain
jccp@ainetsolutions.com
http://www.ainetsolutions.com/

**Abstract.** Automated Text Categorization has reached the levels of accuracy of human experts. Provided that enough training data is available, it is possible to learn accurate automatic classifiers by using Information Retrieval and Machine Learning Techniques. However, performance of this approach is damaged by the problems derived from language variation (specially polysemy and synonymy). We investigate how Word Sense Disambiguation can be used to alleviate these problems, by using two traditional methods for thesaurus usage in Information Retrieval, namely Query Expansion and Concept Indexing. These methods are evaluated on the problem of using the Lexical Database WordNet for text categorization, focusing on the Word Sense Disambiguation step involved. Our experiments demonstrate that rather simple dictionary methods, and baseline statistical approaches, can be used to disambiguate words and improve text representation and learning in both Query Expansion and Concept Indexing approaches.

## 1 Introduction

Automated Text Categorization (ATC) – the automatic assignment of documents to pre-defined categories – is one of the most important text classification tasks nowadays. Automated text classifiers can be employed to route e-mail messages to e-mail folders [1], to populate Web directories with Web pages [2], or to filter out inappropriate e-mail or Web content, like *spam* [3] or pornography [4].

While it is possible to build text classifiers by hand, the most popular approach is using Information Retrieval and Machine Learning techniques to automate the process. In short, given a set of manually classified documents, they are represented as feature vectors and feeded into a learning algorithm, which produces an automatic classifier. There is empirical evidence that ATC systems built this way can achieve human levels of performance, specially when categories are subject or topic oriented, like *arts*, *computers* or *entertainment* [5].

A. Montoyo et al. (Eds.): NLDB 2005, LNCS 3513, pp. 298–309, 2005.
© Springer-Verlag Berlin Heidelberg 2005

However, like Information Retrieval (IR), ATC faces the problem of language variability. The same way polysemy (one word with two or more senses) make ambiguous queries to retrieve irrelevant results, it also can make an ATC classifier to incorrectly label a new document. Also, failing to recognize synonyms make relevant documents not retrieved for a query; the effect in ATC is similar. These problems have been faced since the very early days in Information Retrieval [6], often through the utilization of lexical-semantic resources like thesauri (e.g. Roget's Thesaurus) or Lexical Data Bases (LDBs, e.g. WordNet [7]), and making use of Word Sense Disambiguation (WSD).

WSD consists on the identification of the actual sense of a word in a context. It plays a key role in the usage of lexical-semantic resources, because it is used to map actual word occurrences to their suitable semantic classes (or senses). The increased interest in WSD, represented by the SENSEVAL competitions[1], has led to important performance improvements, although its accuracy is much below other frequent Natural Language Processing tasks, like POS-Tagging.

Being ATC different to IR in many ways, we focus on integrating lexical-semantic resources in it, and studying the role of WSD. Our general goal is improving ATC effectiveness by addressing language variability. We have extended two traditional models of integration of thesauri in IR to ATC, namely Query Expansion and Concept Indexing, and tested relatively simple WSD methods for it. Our WSD methods are based on the two dominant models nowadays, dictionary and learning based. We use the LDB WordNet and the test collections Reuters-21578 and SemCor in our experiments.

Our hypothesis is that it is possible to improve ATC by using LDBs and simple WSD methods. This is due to the fact that in ATC, there is much information available *a priori* in the form of manually labeled documents, which are not available in IR. The results of our experiments confirm the hypothesis, and encourage future work on integrating other resources in ATC.

## 2 Word Sense Disambiguation in Text Classification

In this section, we review the usage of WSD in IR and in ATC, and the current state-of-the-art in WSD methods. The work in IR can be extended to ATC in the form of two models that are described in the next sections.

### 2.1 WSD in Information Retrieval

Since the early days in Information Retrieval (see e.g. [6]), automatic *thesauri* (synonym dictionaries, or lists of semantically related word classes) have been used with the aim of improving retrieval effectiveness. The underlying idea is to overcome language variation problems, specially polysemy and synonymy. If a query submitted to a retrieval engine contains a polysemous word – but intended by the user to be interpreted with only one of its senses (e.g. "bank" in the financial sense), there are many chances that the engine will return false

[1] See `http://senseval.org/`.

matches, corresponding to word occurrences with a different sense (e.g. "bank" as a place to sit on). Also, if a query contains a word that has several synonyms (e.g. "astronaut"), only documents in which the query word occur will be retrieved, probably missing other relevant documents which its synonyms occur in (e.g. documents with the word "cosmonaut"). The first situation affects precision, while the second mainly hurts recall.

Given a semantic classification of words (e.g. a thesaurus), in which related words are grouped in classes, it can be used to improve retrieval effectiveness in two basic ways:

- By replacing word occurrences in documents and queries by the semantic classes they refer to. This way, only the correct meaning of a word will be used for retrieval, avoiding false matches due to polysemy, and improving precision. Most likely, recall will also be improved, because synonyms will be in the same semantic classes, and documents in which they occur will be indexed with regard to the same semantic class.
- By replacing word occurrences by their synonyms or semantically related words (words in the same class). This way, all documents containing a word will also contain their synonyms, and they will be retrieved when any of the words in the class are used in the query. This has the effect of improving recall. We must note that is more efficient to replace only words in queries, and indexing the documents by using only the words occurring in them.

We call the former method *concept indexing*, and the latter *query expansion*. In any case, a WSD system is required. In concept indexing, both queries and documents must be disambiguated, in order to identify the right semantic classes used, and taking them as indexing units. In query expansion, ambiguous words (pertaining to two or more semantic classes) must be disambiguated, allowing their replacement by the right set of synonyms.

Both methods have been applied when using WordNet in Information Retrieval since WordNet's very beginning[2]. The first work was done by Voorhees, dating back to the beginning of the previous decade. In [8] she used a form of concept indexing focused in the WordNet nouns subset, showing a general decrease of performance. Query expansion with WordNet synsets and conceptual relations were after tested by her on a TREC collection, with no better results [9]. These experiences have led her to state that "linguistic techniques must be essentially perfect to help" [10], meaning in this context that Word Sense Disambiguation low effectiveness has a direct impact in the usage of WordNet for Information Retrieval.

However, other works have demonstrated that WordNet can improve text retrieval. In particular, indexing with WordNet synsets has been tested on a (rather artificial) collection in [11], showing that: (1) under perfect disambiguation, concept indexing with WordNet synset greatly improves retrieval effectiveness; and (2) up to a 60% disambiguation errors can be tolerated, while improving performance.

[2] See `http://engr.smu.edu/~rada/wnb/` for a comprehensive bibliography. In this paper, we review only those works we consider specially relevant to our research.

Summing up all experiences, there is an intuition but no clear results supporting that Word Sense Disambiguation can improve Information Retrieval. But in this work, we demonstrate that WSD does improve ATC.

## 3 Learning Based ATC

The most popular model for building ATC systems nowadays, is based on IR and Machine learning techniques, and involves the following steps [5]:

- First, documents in a manually classified collection (the *training collection*) are represented as attribute or feature vectors, in which features are usually stemmed words, after deleting the most frequent ones using a stop list. The value of an attribute in a document vector (its *weight*) can be binary (1 if the stem occur in the document, and 0 otherwise), Term Frequency (TF, the number of times the stem occurs in the document), or TF.IDF (being IDF the Inverse Document Frequency, a function of the times the stem occurs in the whole document collection). This document representation is regarded in the literature as the *bag of words* model, and it corresponds to the standard Vector Space Model in Information Retrieval [12].
- Secondly, attributes are filtered according to an feature quality metric, in order to reduce the vector space dimensionality, allowing learning and generally improving the overall accuracy of the obtained system. Effective quality metrics include Information Gain, $\chi^2$ and the Document Frequency [13].
- Finally, the document vectors are taken as examples by a Machine Learning algorithm, which builds a classification function or *classifier*, based on the previous attributes. The classifier can take the form of a set of rules, a decision tree, a linear discrimination function, etc., depending on the algorithm used. The most effective learners used for this problem include Support Vector Machines, $k$-Nearest Neighbors, and classifier committees like Boosting [5, 14].

Effectiveness of ATC systems built this way is comparable to humans. In words by Sebastiani: *"Automated TC (. . . ) has reached effectiveness levels comparable to those of trained professionals."* [5].

However, ATC faces the same language variability problems as IR. Polysemy makes automatic classifiers wrongly classify new documents. For instance, a document containing "bank" in the sense of a financial institution, may be classified in the *environment* category, if the system understands the word in the bank of a river sense. Also, and regarding synonymy, a classifier trained on documents on which the word "astronaut" occur (in the *space* category), may fail to recognize another document in which only the word "cosmonaut" occurs.

## 4 WSD in Automated Text Categorization

Automated Text Categorization offers new opportunities for improving effectiveness using Word Sense Disambiguation. The limited information and short life

of text retrieval queries contrast with populated, long living categories. Also, information quality metrics used in ATC for dimensionality reduction allow to select appropriate indexing units in text representation. This makes usage of lexical-semantic resources in ATC even more promising than in IR, and WSD less critical for it.

Most works in WSD for ATC (focused in the LDB WordNet), have adopted the concept indexing model from IR. The basic idea of concept indexing with WordNet synsets is recognizing the sysnets to which words in texts refer, and using them as terms for representation of documents in a Vector Space Model. Synset weights in documents can be computed using the same formulas for word stem terms in the bag of words representation.

Experiments focused on concept indexing with WordNet synsets for TC have mixed results. On one side, lack of disambiguation has led to loss of effectiveness in some works. On the other, it is not clear that full disambiguation is absolutely required to obtain a document representation more effective than the bag of words model. We discuss three works specially relevant.

- Scott and Matwin [15] have tested a text representation in which WordNet synsets corresponding to the words in documents, and their hypernyms, were used as indexing units with the rule learner Ripper on the Reuters-21578 test collection. The results of the experiments were discouraging, probably due to the fact that no disambiguation at all is performed, and to the inability of Ripper to accurately learn in a highly dimensional space.
- Fukumoto and Suzuki [16] have performed experiments extracting synonyms and hypernyms from WordNet nouns in a more sophisticated fashion. First, sysnets are not used as indexing units; instead, words extracted from synsets whose words occur in the documents are used. Second, the height to which the WordNet hierarchy is scanned is dependent on the semantic field (location, person, activity, etc.), and estimated during learning. These experiments were performed with Support Vector Machines on the Reuters-21578 test collection, and their results are positive, with special incidence on rare (low frequency) categories. Notably, no sense disambiguation was performed.
- Petridis *et al.* [17] used WordNet synsets as indexing units with several learning algorithms on the SemCor text collection. In this collection, all words and collocations have been manually disambiguated with respect to WordNet synsets. The lazy learner k-Nearest Neighbors, the probabilistic approach Naive Bayes, and a Neural Network were tested on several text representations. The concept indexing approach performed consistently better than the bag of words model, being the Neural Network the best learner.

The work by Scott and Matwin suggests that some kind of disambiguation is required. The work by Fukumoto and Suzuki allows to suppose that no full disambiguation is needed. Finally, the work by Petridis *et al.* demonstrates that perfect disambiguation is effective, over a limited number of learning algorithms and an correctly disambiguated text collection. Positive evidence is scarce yet, and our own recent experiments with this model show mixed results [18].

Much less work has been devoted to the query expansion method in ATC. As far as we know, there is only our previous work [19, 20], followed by [21]. In [19], we proposed a model for learning-based ATC, in which the category names are enriched with WordNet synonymy information, under optimal WSD conditions, and focusing a class of linear learning algorithms. We further automated the WSD step using a heavy knowledge-based algorithm in [20]. In these works, we have employed the LDB WordNet, and the standard ATC test collection Reuters-21578. From these works, it can be stated that the query expansion method is effective, and that perfect or complex WSD methods work for the problem. The work has been adapted to other semi-supervised learning algorithms in [21]. Regarding this method, we address here the simplification of the WSD, adopting a simple dictionary-based WSD method, with positive results.

### 4.1 Current Methods in WSD

Following [22], there two basic methods for WSD:

- Dictionary based WSD, in which only the information in a dictionary (or lexical-semantic resource) is used. The most simple method involves labeling each word occurrence in a context, with the sense in which the definition is more similar to the context.
- Supervised WSD, in which the dictionary is taken only as a reference (if any), and the main source of information is a training collection of sense usage samples. A word occurrence is labeled by an automatic classifier, trained on these samples.

Recent SENSEVAL competitions (specially in the All-Words English Task) demonstrate that:

- Supervised methods perform better than dictionary methods, but there is an increasing interest on integrating both methods.
- Top performing systems are only performing slightly better than statistical baseline methods, still far from from human performance.

Instead of using highly complex methods, as those tested in SENSEVAL, we make use of two simple WSD methods, a dictionary based one for the Query Expansion integration, and a baseline statistical approach in the Concept Indexing method. Therefore, as our results are positive, we demonstrate that the characteristics of the ATC task make perfect WSD not needed.

## 5 Query Expansion Method

In this section, we describe the Query Expansion method for integrating WordNet information in ATC, and we test it on the standard Reuters-21578. We make use of a simple dictionary method for WSD, showing that full WSD is not needed.

### 5.1 Description

In ATC, categories are best described by the set of documents they contain. However, and since categories are mostly provided for human use (like in Web Directories, or the subject keywords in libraries), they have attached a name and sometimes, a short description. We interpret these names as queries in IR, adapting the Query Expansion method the following way[3]:

- For each category, its name is extracted, and searched in the LDB WordNet for its senses (synsets). In the case of multi-word names, if the whole name is not found, it is divided in words, and each of them searched individually. For each category, a set of potential synsets is obtained.
- The correct synset is identified for each ambiguous expression or word. This is a WSD process we detail below. From this step, we get a synset per category, or per word in multi-word categories.
- The words obtained are converted into a term weight vector, each word with its IDF weight in the training collection. For each category, a vector is produced.
- We take the term-weight vector for each category, as the initial vector for a linear learning algorithm (the Rocchio one in our experiments). We run the algorithm on the training collection, generating a prototype vector.

For the learning step, training documents are represented as term-weight vectors according to the Vector Space Model: terms are word stems occurring in between the 1% and 10% of documents; weights are TF.IDF like. When a new document is to be classified, it is represented a as term-weight vector, and its cosine similarity to each category prototype is computed. The document is assigned to the category if the similarity exceeds a predefined threshold.

The WSD step is quite simple. For each of the potential synsets, its neighborhood is found (considering semantic links in WordNet as a graph). Each candidate synset is associated to a set of words extracted from the closest synsets, and we compute the overlap between this set of words and the whole set of words of documents in the category. The synset that shows higher overlap is selected.

This is a very simple WSD based on dictionaries, that exploits the information available for the category in the form of training documents, and the information available for senses, in the form of words in closest synsets.

### 5.2 Experiments

In order to test this WSD method, we have extended previous experiments to this WSD algorithm for this work. We make use of the Rocchio linear learning algorithm as described in [19]. We work on the Reuters-21578 test collection, ModApte Split.

The Reuters-21578 collection consists of 21,578 newswire articles from Reuters collected during 1987. Documents in Reuters deal with financial topics, and were

[3] This method is fully described in [19, 20].

classified in several sets of financial categories by personnel from Reuters Ltd. and Carnegie Group Inc. Documents vary in length and number of categories assigned, from 1 line to more than 50, and from none categories to more than 8. There are five sets of categories: TOPICS, ORGANIZATIONS, EXCHANGES, PLACES, and PEOPLE. As others before, we have selected the 90 TOPICS for our experiments. In the ModApte Split, there are 9,603 documents in the training collection, and 3,299 in the test collection.

In the Table 1, we present the results of our experiments, comparing the usage of WordNet with the dictionary WSD method, and a baseline learning method based only on the Rocchio algorithm. We show the $F_1$[4] values obtained by macro (MF1) and micro averaging (mF1), grouping the categories by the number of training documents available for them (Docs/Topic). This way, we also show how WordNet information is specially valuable in the case of less popular categories, on which learning is limited by insufficient training data. Also, we provide the number of categories (#Topics), the average number of documents in the training (AvgTr) and test (AvgTs) collections, for each of the groups of categories.

**Table 1.** Query Expansion and dictionary WSD evaluation results.

| | | | | No-WN | | WN | |
|---|---|---|---|---|---|---|---|
| Docs/Topic | #Topics | AvgTr | AvgTs | MF1 | mF1 | MF1 | mF1 |
| 1–5 | 20 | 1,15 | 1,55 | 0,140 | 0,179 | 0,397 | 0,464 |
| 6–19 | 16 | 7,56 | 4,13 | 0,163 | 0,202 | 0,373 | 0,419 |
| 21–50 | 19 | 20,11 | 10,89 | 0,191 | 0,191 | 0,545 | 0,562 |
| 51–99 | 14 | 46,79 | 19,21 | 0,329 | 0,327 | 0,601 | 0,595 |
| 100–999 | 19 | 199,47 | 71,89 | 0,629 | 0,647 | 0,645 | 0,651 |
| 1000– | 2 | 2262,50 | 903,00 | 0,864 | 0,864 | 0,812 | 0,805 |
| TOTAL | 90 | 105,51 | 41,61 | 0,303 | 0,696 | 0,517 | 0,712 |

The results shown in the table are very encouraging. Comparing the four right columns, it can be seen that macro-averaged improvements are bigger than micro-averaged ones. Thus, those categories with less learning documents are get more benefit from the WordNet enrichment. This is also confirmed by the fact that performance improvements are achieved for all category groups, except the two more frequent categories. This also suggests that the weights used in the Rocchio algorithm (and in general, the importance given to the information extracted from WordNet, versus the information in the training collection) can be balanced in proportion with the number of training documents.

In general, ATC effectiveness is substantially improved. In our opinion, perfect WSD is not required; the method can tolerate the mistakes performed by a simple dictionary WSD method, still improving effectiveness.

[4] $F_1$ is the standard quality metric in ATC. Is averages recall and precision. Macro-averaged $F_1$ gives equal importance to all categories, while micro-averaged $F_1$ gives more importance to popular categories. It is important to show both [5].

## 6 Concept Indexing Method

In this section, we present the Concept Indexing method for integrating lexical-semantic information in learning based ATC. Also, we detail our baseline statistical WSD method, and test it in a systematic way. The results of our experiments show that no perfect WSD is required for alleviating language variability problems in ATC.

### 6.1 Description

The basic model for Concept Indexing, sketched above, consist of using concepts (WordNet synsets in our case) as indexing units in text representation. The rest of the process for learning based ATC is kept as usual. Instead of this simple formulation, and following [8] ideas, we have appended two representations: a word based representation, and a concept based one. This one, a text document is represented as a term and concept weight vector.

We let the selection step to decide which are the most representative indexing units; depending on the category, there can be more synsets than words, or the opposite, as indexing units. This way, we exploit the selection methods and the information contained both in the training document collection, and the lexical-semantic resource.

### 6.2 Word Sense Disambiguation Method

This method requires WSD, because it is needed to identify the correct sense (synset) for each word in a document. In previous experiments [18], we found some evidence supporting that perfect WSD was required if only concepts were used as indexing units. Since we use here synsets and words, we can use a simple WSD based on sense frequency and POS tags. For each word, we assign it the most frequent sense for the POS tag it has in the context. This method is used as baseline in SENSEVAL experiments, and many sophisticate WSD are unable to reach its effectiveness.

### 6.3 Experiments

We have performed several series of experiments, using the SemCor text collection. The SemCor text collection is a Semantic Concordance, a corpus tagged with WordNet senses in order to supplement WordNet itself (for instance, for researching or showing examples of sense usage). However, SemCor has been adapted and used for testing IR by Gonzalo *et al.* [11], and used for evaluating TC by Petridis *et al.* [17]. Moreover, there are not other collections tagged with conceptual information in depth, and so, indexing with "perfect" disambiguation can hardly tested without SemCor. However, we have changed manually labeled references to second and other senses to sense one, simulating the statistical WSD method proposed above. SemCor has 15 genre oriented classes, and

**Table 2.** Concept indexing and statistical WSD evaluation results.

| | Synsets | | Words | | Combined | |
|---|---|---|---|---|---|---|
| Alg. | MF1 | mF1 | MF1 | mF1 | MF1 | mF1 |
| NB | 0,631 | 0,739 | 0,635 | 0,750 | 0,702 | 0,837 |
| C45 | 0,258 | 0,391 | 0,270 | 0,382 | 0,280 | 0,396 |
| SVM | 0,502 | 0,773 | 0,482 | 0,730 | 0,467 | 0,763 |
| ABNB | 0,638 | 0,759 | 0,638 | 0,760 | 0,682 | 0,824 |

147 documents. Given this scarce information, we have used the 10-fold cross validation evaluation methodology [5].

In the Table 2, we present the results of our experiments. We have tested a synset based representation, a word based representation, and the combined approach, using a representative range of high performance learning algorithms. From those tested in the literature [5], we have selected the following ones[5]: the probabilistic approach Naive Bayes (NB); the decision tree learner C4.5 (C45); the Support Vector Machines kernel method (SVM); and the AdaBoost meta-learner applied to Naive Bayes (ABNB). We present macro (MF1) and micro averaged (mF1) $F_1$ values.

Given the results, it is clear that the top performing method is NB operating on the combined representation. While AdaBoost usually improves its base method, and this is observed on the synset and word based representations, it has not been able to do so on the combined representation. This may be due to the fact that AdaBoost improves *weak* methods, but Naive Bayes is proved very accurate on this particular problem. Also, and more in general, the combined representation improves synset and word based representations for all the algorithms, except for Support Vector Machines. The decision tree learner can be considered a weak baseline for this problem.

Under the light of these results, we believe that no advanced WSD is needed, and still ATC can be greatly improved. Still, bigger improvements may be achieved by using the nearest synsets to those occurring in the documents (through hyponymy and meronymy WordNet relations). Also, these results should be confirmed by testing similar representations on a bigger test collection, like the Reuters-21578 one.

## 7 Conclusions

With the aim of reducing language variability, we have sketched two method for integrating lexical-semantic information in learning ATC. These methods, based on traditional work in IR, are called Query Expansion and Concept Indexing. In both methods, WSD plays a key role, either disambiguating categories names, or training and testing documents. We have designed simple WSD methods for each model, based on current work in the area.

[5] We exclude some references for brevity. Please find a sample at [5].

We have conducted several series of experiments with the LDB WordNet and the standard Reuters-21578 and SemCor collections. The results of these experiments demonstrate that no heavy, full WSD is required, still improving ATC effectiveness. This results are very encouraging, and contrast with the mixed results presented in other works in IR and ATC.

## References


1. Zhdanova, A.V., Shishkin, D.V.: Classification of email queries by topic: Approach based on hierarchically structured subject domain. In Yin, H., Allinson, N., Freeman, R., Keane, J., Hubbard, S., eds.: Proceedings of IDEAL-02, 3rd International Conference on Intelligent Data Engineering and Automated Learning, Manchester, UK, Springer Verlag, Heidelberg, DE (2002) 99–104 Published in the "Lecture Notes in Computer Science" series, number 2412.
2. Mladenić, D.: Turning YAHOO! into an automatic Web page classifier. In Prade, H., ed.: Proceedings of ECAI-98, 13th European Conference on Artificial Intelligence, Brighton, UK, John Wiley and Sons, Chichester, UK (1998) 473–474
3. Gómez, J.: Evaluating cost-sensitive unsolicited bulk email categorization. In: Proceedings of SAC-02, 17th ACM Symposium on Applied Computing, Madrid, ES (2002) 615–620
4. Hepple, M., Ireson, N., Allegrini, P., Marchi, S., Montemagni, S., Gómez, J.: NLP-enhanced content filtering within the POESIA project. In: Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC 2004). (2004)
5. Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys **34** (2002) 1–47
6. Van Rijsbergen, C.J.: Information Retrieval. Butterworths, London (1979)
7. Miller, G.A.: WordNet: A lexical database for English. Communications of the ACM **38** (1995) 39–41
8. Voorhees, E.M.: Using wordnet to disambiguate word sense for text retrieval. In: Proceedings of SIGIR-93, 16th ACM International Conference on Research and Development in Information Retrieval, Pittsburgh, US (1993) 171–180
9. Voorhees, E.M.: Query expansion using lexical-semantic relations. In Croft, W.B., van Rijsbergen, C.J., eds.: Proceedings of the 17th Annual International Conference on Research and Development in Information Retrieval, London, UK, Springer Verlag (1994) 61–70
10. Voorhees, E.: Using WordNet for text retrieval. In: WordNet: An Electronic Lexical Database. MIT Press (1998)
11. Gonzalo, J., Verdejo, F., Chugur, I., Cigarrán, J.: Indexing with WordNet synsets can improve text retrieval. In: Proceedings of the COLING/ACL Workshop on Usage of WordNet in Natural Language Processing Systems. (1998)
12. Salton, G.: Automatic text processing: the transformation, analysis, and retrieval of information by computer. Addison Wesley (1989)
13. Yang, Y., Pedersen, J.: A comparative study on feature selection in text categorization. In: Proc. Of the 14th International Conf. On Machine Learning. (1997)
14. Yang, Y., Liu, X.: A re-examination of text categorization methods. In Hearst, M.A., Gey, F., Tong, R., eds.: Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval, Berkeley, US, ACM Press, New York, US (1999) 42–49

15. Scott, S.: Feature engineering for a symbolic approach to text classification. Master's thesis, Computer Science Dept., University of Ottawa, Ottawa, CA (1998)
16. Fukumoto, F., Suzuki, Y.: Learning lexical representation for text categorization. In: Proceedings of the NAACL 2001 Workshop on WordNet and Other Lexical Resources. (2001)
17. Petridis, V., Kaburlasos, V., Fragkou, P., Kehagias, A.: Text classification using the $\sigma$-FLNMAP neural network. In: Proceedings of the 2001 International Joint Conference on Neural Networks. (2001)
18. Gómez, J., Cortizo, J., Puertas, E., Ruíz, M.: Concept indexing for automated text categorization. In: Natural Language Processing and Information Systems: 9th International Conference on Applications of Natural Language to Information Systems, NLDB 2004, Salford, UK, June 23-25, 2004, Proceedings. Lecture Notes in Computer Science, Vol. 3136, Springer (2004) 195–206
19. de Buenaga Rodríguez, M., Gómez Hidalgo, J., Díaz Agudo, B.: Using wordnet to complement training information in text categorization. In Nicolov, N., Mitkov, R., eds.: Recent Advances in Natural Language Processing II: Selected Papers from RANLP'97. Volume 189 of Current Issues in Linguistic Theory (CILT)., John Benjamins (2000) 353–364
20. Ureña-López, L.A., Buenaga, M., Gómez, J.M.: Integrating linguistic resources in TC through WSD. Computers and the Humanities **35** (2001) 215–230
21. Benkhalifa, M., Mouradi, A., Bouyakhf, H.: Integrating external knowledge to supplement training data in semi-supervised learning for text categorization. Information Retrieval **4** (2001) 91–113
22. Manning, C., Schütze, H.: 16: Text Categorization. In: Foundations of Statistical Natural Language Processing. The MIT Press, Cambridge, US (1999) 575–608

# Combining Biological Databases and Text Mining to Support New Bioinformatics Applications

René Witte[1] and Christopher J.O. Baker[2]

[1] Institute for Program Structures and Data Organization (IPD)
Universität Karlsruhe (TH), Germany
witte@ipd.uka.de
[2] Department of Computer Science and Software Engineering
Concordia University, Montréal (Québec), Canada
baker@encs.concordia.ca

**Abstract.** A large amount of biological knowledge today is only available from full-text research papers. Since neither manual database curators nor users can keep up with the rapidly expanding volume of scientific literature, natural language processing approaches are becoming increasingly important for bioinformatic projects.
In this paper, we go beyond simply extracting information from full-text articles by describing an architecture that supports targeted access to information from biological databases using the results derived from text mining of research papers, thereby integrating information from both sources within a biological application.
The described architecture is currently being used to extract information about protein mutations from full-text research papers. Text mining results drive the retrieval of sequence information from protein databases and the employment of algorithmic sequence analysis tools, which facilitate further data access from protein structure databases. Complex mapping of NLP derived text annotations to protein structures allows the rendering, with 3D structure visualization, of information not available in databases of mutation annotations.

## 1 Introduction

Biological researchers today have access to vast amounts of research data. Unlike in many other disciplines, these results are not only published in research papers, but additionally in a structured form within several publicly accessible databases. This data describes a unique array of information on biological entities such as DNA, proteins, and small molecules. A large proportion of salient information is however still hidden within individual research papers. Moreover, the rate at which new findings are being published is much higher than individual scientists or engineers can cope with, which is hindering further research and the development of industrial applications. For this reason, NLP techniques are progressively being applied in the area of biology.

Existing work in the area of biological text mining systems so far has focused on delivering extraction-based systems for biological research and database curation projects. Examples for such systems are: (1) The *BioRAT* system [3], which

A. Montoyo et al. (Eds.): NLDB 2005, LNCS 3513, pp. 310–321, 2005.
© Springer-Verlag Berlin Heidelberg 2005

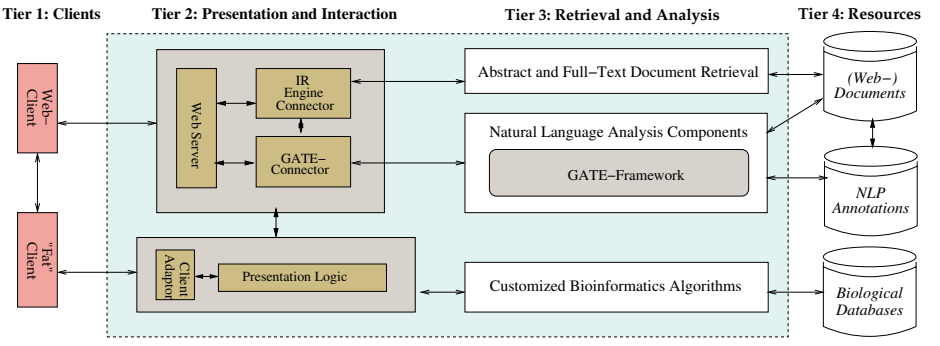



**Fig. 1.** System architecture for the integration of NLP with biological databases.

combines an information retrieval engine with an information extraction component based on user-definable templates (regular-expression based grammars). Users can then view extracted text segments instead of reading the full-text papers. (2) *ProFAL* (PROtein Functional Annotation through Literature) [4] is a system that annotates entries in biological databases with information found in the scientific literature, supporting manual database curation by proposing information from texts as supplementary data for biological entities. (3) *Textpresso,* an "ontology-based information retrieval and extraction system for Biological Literature" [9], which also aims at supporting biological database curation tasks.

While we also aim at supporting biologists through text mining, our work differs in that we want to provide a foundation for new biological applications by directly linking information obtained through NLP with the available biological databases. In other applications, like BioRAT or Textpresso, the textual results are meant for human consumption, so the often low precision of text mining systems is less critical. In our case, however, there is no human step between the NLP system and further application-specific processing. To ensure a reliable cross-linking is possible, NLP-derived results must be more rigorously structured, filtered, and analyzed before being used for bioinformatics applications. We demonstrate the feasibility of this idea with a biologically relevant application for the visualization of protein 3D structures.

## 2 Connecting Biological Databases and Text Mining

In this section we present an architecture for combining text mining results with biological databases and in-silico algorithms. It follows a standard multi-tier information system design, similarly to the application discussed in [13]. Figure 2 shows the main components, which we now discuss in detail.

*Tier 1: Clients.* The first tier provides access to the system, typically used by humans, but potentially also for other automated clients. Most services and data will be delivered through a web browser, while some programs could require additional "fat clients" or *Java* applets, like a 3D-visualization component.

*Tier 2: Presentation and Interaction.* Tier 2 is responsible for information presentation and user interaction. In our architecture, it has to deal with both service access and content visualization. A connector for an information retrieval engine allows the dispatch of user queries to an IR system to obtain documents. Retrieved documents can then be queued for processing by an NLP system. Finally, it allows for the control of specialized in-silico applications, the interaction between the user agent, the processed NLP results, and the bioinformatics algorithms.

*Tier 3: Retrieval and Analysis.* Tier 3 provides all the document analysis and retrieval functions discussed above. In order to access biological documents, the architecture can be equipped with a stand-alone information retrieval engine like *Lucene* or a web-spidering component. The natural language analysis part is based on the GATE *(General Architecture for Text Engineering)* framework [5], one of the most widely used NLP tools. Since it has been designed as a component-based architecture, individual analysis components can be easily added, modified, or removed from the system. Finally, application-specific algorithms are needed to process the NLP-derived results, filtering and supplementing them with data from biological databases. These algorithms, in turn, can reference standard bioinformatics tools like BLAST [1] or CLUSTAL W [12].

*Tier 4: Resources.* Input documents (research papers) either come directly from the Web (or some other networked source, like emails), or a full-text database. Results from the NLP component are stored as annotations to the original documents in their own database. They can be queried for specific keywords (for example, finding all references to a particular protein), or exported to XML for exchange with other applications. Finally, in order to verify, process, and supplement the text mining results the architecture needs access to various biological databases, for example the PDB, Brenda, or *Entrez*.

## 3 Case Study: The MutationMiner System

In this section we present the *MutationMiner* system we have developed within the architecture described above. It combines text mining results from protein engineering literature with biological databases to support enhanced 3D structure visualizations of proteins [2]. We give preliminary results and outline areas for further improvement, which are discussed in more detail in section 4.

### 3.1 Biological Background

The motivation for this work is the ever-increasing amount of scientific literature detailing the effects of mutations to proteins. A bio-engineer working on the improvement of an enzyme, for example for its use within an industrial process, needs an understanding of the impact of all mutations carried out on the particular protein family. This requires a complex mapping of sequence mutants to a common protein structure. Currently the protein mutation database (PMD) [7] and associated visualization tools can provide this capability. The content of this

database is limited however by the speed at which newly published papers can be processed. In 1999 the PMD authors reported a three-year backlog of unprocessed publications. Since the arrival of high-throughput sequence modification techniques, such as directed evolution, a greater number of mutant sequences are produced along with information about their improved performance under precisely defined conditions.

Our goal, therefore, is to develop text mining tools that automatically scan literature and extract information about protein mutations. The extracted information can then be used to access protein sequence information from biological databases for use by a sequence alignment algorithm, which in turn queries protein structure information needed for 3D visualization. A protein engineer can then view structural representations of proteins (obtained from protein databases) combined with annotations describing mutations and their impacts (extracted through text mining from publications).

*Protein Mutations in the Literature.* Enzymes are proteins that catalyze specific biochemical reactions. Each enzyme family carries out conversions of distinct chemical substrates to chemical products. Within an enzyme family each individual enzyme has different physiochemical operating parameters, like temperature optimum, pH optimum, or thermal stability. Mutation of protein sequences has in many cases resulted in the production of enzymes with altered properties and is a common approach to enzyme improvement. Such mutations are typically the change of amino acids of the protein sequence achieved using molecular biology techniques such as site directed mutagenesis or directed evolution. The properties of the amino acids at specific positions on the protein sequence are the determining factor, however which amino acids and which positions are responsible for particular enzyme properties is not always known. For this reason, protein engineers routinely mutate residues and document their impacts on enzyme characteristics of special interest in scientific publications.

*Structure Visualization.* The complex structure of a protein is intrinsically related to its function and the elucidation and manipulation of protein structures to enhance protein function has valuable practical benefits. Protein structure visualization tools allow the protein engineer to view and rotate three-dimensional images in various representations. This in turn allows for the interpretation of experimental or computational results in a spatial context and facilitates the generation of hypotheses concerning the mechanistic interactions of the protein with substrate ligands. For these reasons, it is important to be able to link text mining results in an automated fashion to such 3D visualizations.

### 3.2 System Architecture and Implementation

The system, as outlined above, needs to integrate document retrieval, NLP-based text analysis, protein sequence database access, protein sequence analysis, and output format generation within a single architecture. Figure 2 shows the enhanced architecture based on the design presented in section 2.

Users interact with the system using a standard web client (tier 1). A web server (tier 2) receives a query (e.g., for a protein family) and dispatches it to an

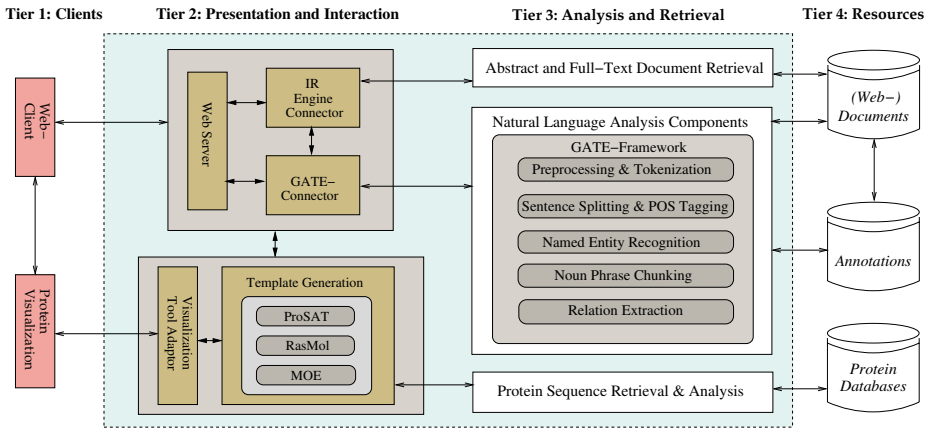



**Fig. 2.** MutationMiner System Architecture.

IR subsystem (tier 3), which retrieves relevant texts from the Web (e.g., NCBI's PubMed) or a local database (tier 4). Retrieved abstracts or full-length papers (if available) are then run through the NLP subsystem (tier 3) to identify mutations and extract relevant information. This information is then used by another tier 3 component to search *Entrez* in order to identify protein accessions and retrieve protein sequences from a biological sequence database. Mutated residues located on eligible sequences are then combined with the information extracted from the documents and converted into tool-specific output formats (tier 2). The user can then access the combined information through a protein visualization tool like ProSAT. In the remainder of this section, we discuss the individual components in more detail and show preliminary results.

**Text Mining Subsystem.** The NLP step needs to identify the proteins being mutated so that the corresponding amino acid sequence can be retrieved from a database. To do this the retrieved documents are run through an NLP subsystem that extracts proteins, host organisms, mutations, their interrelations, as well as provided accession numbers. A full text or abstract, once retrieved and converted into a suitable input format, is run through a so-called processing pipeline of GATE components, which we describe in more detail below.

*Preprocessing and Gazetteering.* After dividing the input stream into individual tokens in the *tokenization* step, a lookup phase identifies words and expressions based on a number of precompiled lists, like person names, companies, measurements, and biomedical-related lists, like chemicals, drugs, genetic structures, or protein names. Based on these lists, a *Gazetteer* component annotates words with a major and minor type, which forms a two-level hierarchy, similar to a (very simple) ontology. For non-biomedical information, we rely on lists contained in the ANNIE information extraction system that comes with GATE. Biomedical lists use the same resources as the BioRAT system described in [3]: lists of entries

extracted from the MeSH hierarchy and SwissProt, together holding more than five million words in roughly 650,000 entries.

*Sentence Splitting and POS Tagging.* The next two components split the input text into individual sentences and then, for each sentence, annotate each word with its *part-of-speech (POS)* tag using the Hepple tagger.

*Named Entity Recognition.* In the next stage, several finite-state transducers combine individual tokens into more complex named entities (NEs), based on regular-expression grammars and specialized tokenizers, which are run over the annotations generated by the previous steps. Examples for entities we detect are *persons* (containing a first name, last name, and possibly initials), *protein expressions*, or *database accession identifiers*. At this stage we also identify *mutation expressions*, which can occur in many different formats.

*Noun Phrase Chunking.* Another JAPE (finite-state transducer) grammar analyses the text and builds up more complex grammatical structures, so-called *noun phrases*, which include determiners, modifiers, and head nouns. For example, the words *"The specific enzyme activity"* will be identified as a single noun phrase (NP) with its words marked up as *"The/DET specific/MOD enzyme/MOD activity/HEAD."* An important feature of our NP chunker is its ability to incorporate the named entities detected above in addition to using POS tags. This allows us to alleviate some of the problems that result from using standard POS taggers, which are statistically trained on more general domains like newspaper articles, for biomedical documents. Finally, we mark all those noun phrase structures that contain a biological named entity.

*Relation Detection.* The last step is the correct identification and interpretation of relations between entities. For our task, we need to be able to identify two kinds of relations: between *proteins* and *mutations,* that is, which protein has been mutated within the described experiment; and between *proteins* and *taxonomic origin,* which we need to correctly retrieve amino acid sequences from protein sequence databases. For the protein-mutation identification, we currently extract all sentences that contain mutation expressions as identified by the corresponding NE grammar. We then scan these sentences for the protein expression, making the simple assumption that the protein mentioned together with the mutations must be the one that has been mutated. For example, in the sentence: *"Wild-type and mutated xylanase II proteins (termed E210D and E210S) were expressed in S. cerevisiae grown in liquid culture."* we identify two mutations, E210D and E210S, and one protein expression, *"xylanase II proteins,"* which we then assume is the protein being mutated. As this approach is quite simplistic, it might fail in a number of cases, especially when more than one protein mutation is described within a single paper. However, since we only extract those mutations where we can identify a corresponding host organism, this approach has been shown to work reliably within our case study on selected xylanase papers. For extracting the second (protein-host) relation we use a template-based approach that matches certain NP-NP patterns where one noun phrase contains the protein expression identified as the one being mutated (e.g., *xylanase II*), with NPs containing an expression marked as an organism (e.g., algae or fungi).

**1: P36217. Reports Endo-1,4-beta-xyl...[gi:549461] BLink, Domains, Links**

```
>gi|549461|sp|P36217|XYN2_TRIRE Endo-1,4-beta-xylanase 2 precursor
MVSFTSLLAASPPSRASCRPAAEVESVAVEKRQTIQPGTGYNNGYFYSYWNDGHGGVTYTNGPGG
QFSVNWSNSGNFVGGKGWQPGTKNKVINFSGSYNPNGNSYLSVYGWSRNPLIEYYIVENFGTYNP
```

**Fig. 3.** Protein sequence data in FASTA format for *xylanase 2* retrieved from *Entrez* using protein names and organisms obtained by NLP analysis.

**CLUSTAL W (1.82) multiple sequence alignment**

```
                  10        20        30        40        50
1  YRF-TGTYK-GTVKSDGGTYDIYTTTRYNAPSIDGD-RTTFTQYWSVRQS   gi|139865|sp|P09850|XYNA_BACCI
1  YRF-TGTYK-GTVKSDGGTYDIYTTTRYNAPSIDGD-RTTFTQYWSVRQS   gi|640242|pdb|1BCX|Xylanase
1  YRF-TGTYK-GTVTSDGGTYDVYQTTRVNAPSVEG--TKTFNQYWSVRQS   gi|17942986|pdb|1HIX|BChain
1  YRF-TGAYK-GSFYADGGTYDIYETTRVNQPSIIG--IATFKQYWSVRQT   gi|1351447|sp|P00694|XYNA_BACP
1  YNPSTGATKLGEVTSDGSVYDIYRTQRVNQPSIIG--TATFYQYWSVRRN   gi|549461|sp|P36217|XYN2TRIRE
1  YNFCSSAITSLGTVYSDGSTYQVGTDTRTNEPSITG--TSTFTQYFSVRES   gi|465492|sp|P33557|XYN3_ASPKA
1  RGVPLDGVGFQSHLIVG---QVPGDFRQNLQRFADLGVDVRITELDIRMR   gi|121856|sp|P07986|GUX_CELFI
1  RGVPIDGVGFQSHFNSGS--PYNSNFRTTLQNFAALGVDVAITELDIQG-   gi|6226911|sp|P26514|XYNA_STRL
1  RGVPIDGVGFQGHFINGMSPEYLASIDQNIKRYAEIGVIVSFTEIDIRIP   gi|139886|sp|P10478|XYNZ_CLOTM
```

**Fig. 4.** Alignment of *xylanase* sequences obtained from the *Entrez* database.

**Biological Database Integration and Protein Sequence Analysis.** As outlined above, information retrieved from documents is used to access various biological databases for the retrieval of protein sequence and structure data, which in turn is used for further processing steps. The end product of our application is a combined data set for protein 3D-structure visualization containing information from both scientific publications and databases. In the following paragraphs, we discuss how data obtained in the text mining step can be processed by in-silico bioinformatics tools and linked to databases.

*Protein Sequence Database Access.* The second step in the process is the retrieval of protein sequences from a sequence database for each protein/organism combination detected in the text mining subsystem. For this, we access the *Entrez* databases in order to identify protein accessions and retrieve protein sequences in FASTA format [10]. *Entrez* is the integrated, text-based search and retrieval system used at *National Centre for Biotechnology Information (NCBI)* for the major databases, including PubMed Scientific Literature, Nucleotide and Protein Sequences, Protein Structures, Complete Genomes, and Taxonomy[1]. The key needed for a successful retrieval is a correct protein/organism pair. Figure 3 shows an example of a retrieved *xylanase* sequence in FASTA format (for programmatic purposes the sequence is obtained in XML format).

*Sequence Analysis.* The sequence analysis component takes the sequences obtained in the previous step and processes them for similarity. Outlying and duplicated sequences are identified using multiple sequence alignment (MSA) and statistical scoring with user-specified threshold criteria. Figure 4 shows an excerpt of a MSA with CLUSTAL W [12]. A list of candidate sequences for which

[1] The complete list of Entrez databases can be viewed at `http://www.ncbi.nlm.nih.gov/Database/index.html`

| | |
|---|---|
| *Title* | Crystallographic Analyses Of Family 11 Endo–1,4-Xylanase Xyl1 |
| *Classification* | Hydrolase |
| *Compound* | Mol_Id: 1; Molecule: Endo-1,4–Xylanase; Chain: A, B; Ec: 3.2.1.8; |
| *Exp. Method* | X-ray Diffraction |

```
JRNL TITL 2 ENDO-[BETA]-1,4-XYLANASE XYL1 FROM STREPTOMYCES SP. S38
JRNL REF  ACTA CRYSTALLOGR.,SECT.D V. 57 1813 2001
JRNL REFN ASTM ABCRE6 DK ISSN 0907-4449
...
DBREF  1HIX A    1   190  TREMBL Q59962   Q59962
DBREF  1HIX B    1   190  TREMBL Q59962   Q59962
...
ATOM 1 N   ILE A 4 48.459 19.245 17.075 1.00 24.52 N
ATOM 2 CA  ILE A 4 47.132 19.306 17.680 1.00 50.98 C
ATOM 3 C   ILE A 4 47.116 18.686 19.079 1.00 49.94 C
ATOM 4 O   ILE A 4 48.009 17.936 19.465 1.00 70.83 O
ATOM 5 CB  ILE A 4 46.042 18.612 16.837 1.00 50.51 C
ATOM 6 CG1 ILE A 4 46.419 17.217 16.338 1.00 51.09 C
ATOM 7 CG2 ILE A 4 45.613 19.514 15.687 1.00 54.39 C
ATOM 8 CD1 ILE A 4 46.397 17.045 14.836 1.00 46.72 C
ATOM 9 N   THR A 5 46.077 19.024 19.828 1.00 40.65 N
...
MASTER   321  0  0  2  28  0  0  9 3077  2  0 30
END
```

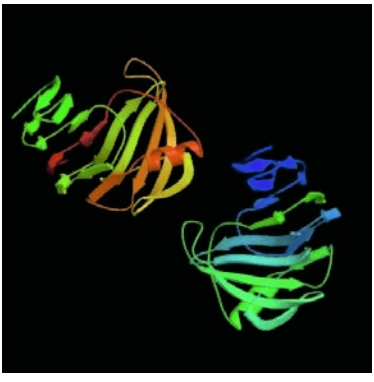


**Fig. 5.** Protein Data Bank (PDB) record for 1HIX and its 3D-visualization.

protein mutation annotations from the papers may be written to a structure visualization tool input format is generated. Before annotations are written to an input format the sequences are further evaluated for a number of features. Domain complexity is evaluated using CDD *(Conserved Domain Database)* search tools [8] and non-target domains are trimmed. Mutated residues are located on the retrieved sequences and only sequences bearing the declared wild type residues at the specified coordinates with the correct offset between multiple mutations are eligible for subsequent sequence-structure alignment.

*Structure Selection.* The choice of a protein structure for mapping and visualization of mutations can be generated dynamically or is user-defined. A dynamically selected structure is the top hit obtained when the consensus sequence of all eligible sequences is pairwise aligned using BLAST against the database of sequences of structures contained in the Protein Data Bank. The structure of the selected sequence (top hit) is used as the template to render the mutations and associated annotations from a variety of sequence mutations described in publications. The mapped coordinates of the mutated residues on the structure sequence are identified by pairwise BLAST alignment. More details on the sequence analysis algorithm can be found in [2].

*3D-Structure Database Access.* We now retrieve the corresponding structure from the Protein Data Bank (PDB). This database is the single worldwide repository for the processing and distribution of 3D biological macromolecular structure data. The name of the structure, e.g. 1HIX, identified in the selection step is the key term entered in the PDB query engine and facilitates the direct download

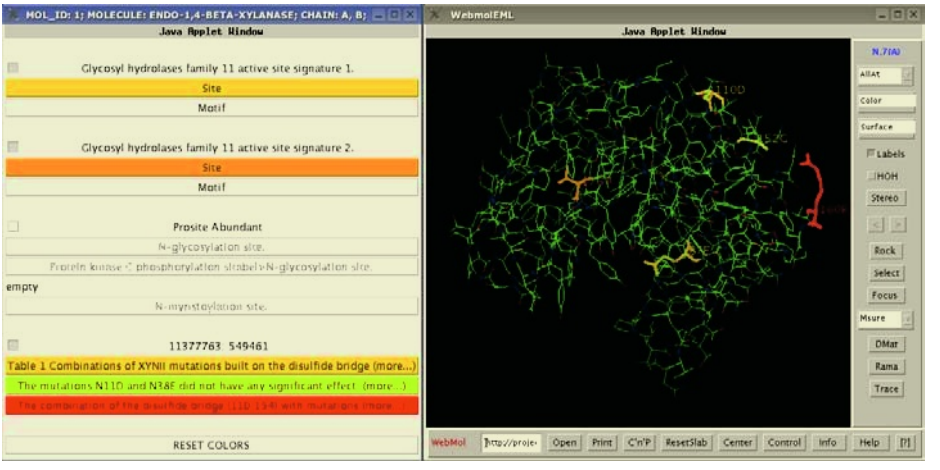



**Fig. 6.** ProSAT showing a 3D (Webmol) visualization of the endo-1,4-$\beta$-protein with mutations extracted through text mining, selected with the interface on the left (sections of the extracted information is displayed on the buttons).

of the structure file. Figure 5 shows an example of a `pdb`[2] file containing atom coordinates and the corresponding amino acids used by a variety of structure visualization tools for rendering images of protein structures in 3D. Amino acid residue identity and coordinates, columns 4 and 6, facilitate mapping of mutation annotations extracted through text mining.

**Application Integration.** After sequence analysis has legitimized the transfer of annotations from a particular text to a residue on the structural homolog, ranking and formatting of sentences is necessary. Formatted annotations are produced depending on the input format for a particular visualization tool.

Currently, only the ProSAT template [6] with additional provision for non-database annotations is employed, while other tools could be enhanced for this purpose as well. The annotations, along with the Genbank protein Identifier (GI) and PubMed ID (PMID) for the originating publication, are uploaded to the ProSAT server and rendered alongside the structural homolog through a Webmol interface. Coloured mutated residues are highlighted in structure and described in a corresponding annotation panel, as shown in Figure 6.

### 3.3 Case Study and Results

Improvement of enzyme features is particularly relevant when the enzyme of interest catalyses an industrially relevant reaction. In our case study we have chosen to mine texts describing mutations to *xylanase* enzymes. Such enzymes depolymerise the plant cell wall component *xylan* that is partly responsible for

[2] Further information on the standard PDB data format can be found at `http://www.rcsb.org/pdb/`.

**Table 1.** NLP subsystem partial evaluation results.

| | Abstract only | | Full paper | |
|---|---|---|---|---|
| | Protein/Organism | Mutations | Protein/Organism | Mutations |
| Precision | 0.88 | 1.00 | 0.91 | 0.84 |
| Recall | 0.71 | 0.85 | 0.46 | 0.97 |
| F-Measure | 0.79 | 0.92 | 0.61 | 0.90 |

dark colour of unbleached paper. Chlorine based oxidizing chemicals are typically used to bleach paper and result in considerable effluent problems for the pulp and paper industry. Xylanases are now used to remove xylan, which results in less chlorine being required for bleaching. Xylanases have been specifically improved to perform well under industrial conditions (high temperature, alkaline) required by the pulp bleaching process.

For our first system evaluation we selected twenty papers on xylanase mutations. Table 1 shows the results of a preliminary (manual) evaluation of the NLP subsystem. We evaluated (a) whether the system found the correct protein-organism pairs (i.e., it must have identified the protein, the organism, and correctly assigned the protein to its host organism) and (b) how many mutations it found. We are currently preparing more extensive, automated evaluations of the NLP subsystem, the sequence analysis component, and the overall system. However, with respect to the NLP part, the most problematic entities are currently author-invented abbreviations.

### 3.4 Summary

The case study has addressed a complex biological data integration problem and highlighted the feasibility of integrating literature-derived annotations with in-silico biology. The extent to which the text mining systems combined with sequence analysis tools and existing biological data can provide additional insight to structural biology and protein engineering will be determined from the future employment of the prototype software by expert protein engineers knowledgeable of specific protein families. We consider the use of text mining to drive protein structure visualization as an innovative approach that provides the protein engineer with enhanced access to the knowledge reported by other investigators without the need for time consuming manual literature searches.

## 4 Future Work

From an application perspective, the inclusion of further information describing enzyme characteristics of wild type proteins for contrasting with the improvements to particular features of these enzymes described in the literature is of additional value to the protein engineer. Such descriptions in the literature often refer to fold increases without necessarily providing units of measurement. Bringing wild type data together with mutation induced improvements is clearly

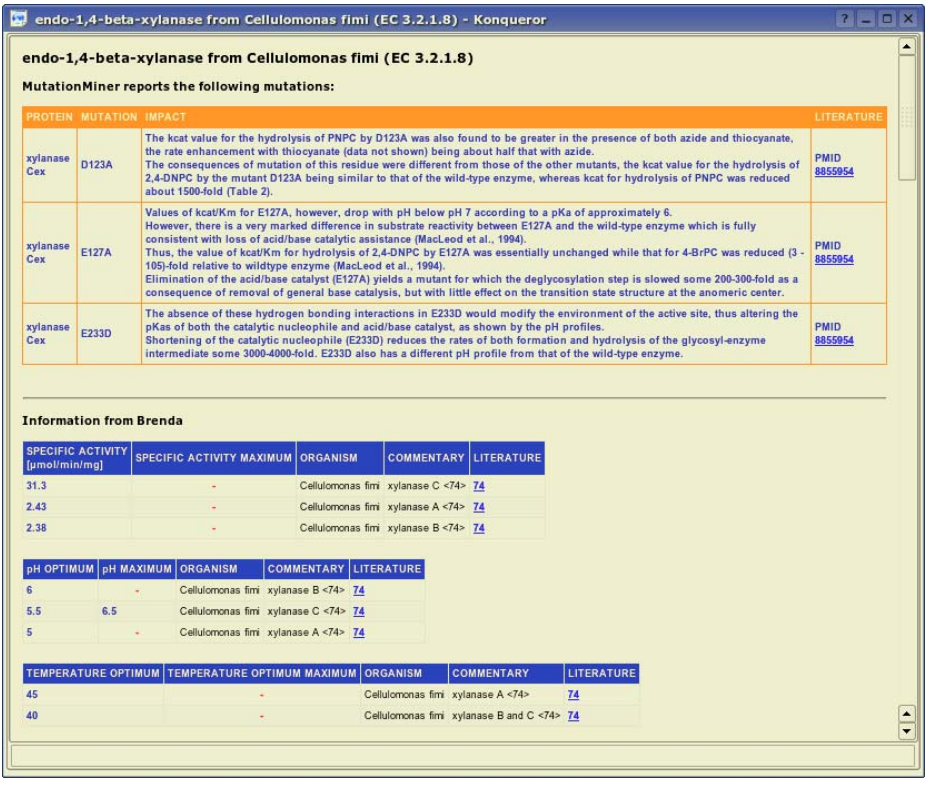



**Fig. 7.** A simulated screenshot of connecting text mining results of a mutated protein with its corresponding wild-type information from the *Brenda* database.

valuable and information of this kind is of great value in decision making for future investigations. For example, before embarking on major mutational studies to improve an enzyme for a particular property it is important to know if there is a precedent of such an achievement within any protein family. Text mining of mutation literature accompanied with wild type information provided by database searching complements this need and can be achieved by the architecture we describe. Figure 7 shows an example by simulating a connection between our system and the *Brenda* database [11], which we plan to automate in the future.

## 5 Conclusions

In this paper we present an architecture enabling new biological applications by linking biological databases with text mining results from research papers.

The protein mutation example shows that text mining results of scientific literature can provide enough information to access and link numerous biological databases to build or enhance in-silico bioinformatics applications.

An important insight of our work is that the often imprecise and incomplete results from natural language processing techniques can be automatically filtered through bioinformatics algorithms and supplemented with information from existing databases.

## Acknowledgements

Vladislav Ryzhikov implemented and evaluated significant parts of the NLP subsystem. The authors would like to thank Razif R. Gabdoulline and Rebecca Wade for their help and collaboration in adapting their ProSAT system to accept textual annotations.

## References

1. S. F. Altschul, W. Gish, W. Miller, E. W. Meyers, and D. J. Lipman. Basic Local Alignment Search Tool. *Journal of Molecular Biology*, 215(3):403–310, 1990.
2. Christopher J. O. Baker and René Witte. Enriching Protein Structure Visualizations with Mutation Annotations Obtained by Text Mining Protein Engineering Literature. In *The 3rd Canadian Working Conference on Computational Biology (CCCB'04)*, Markham, Ontario, October 4 2004. Co-located with IBM CASCON.
3. D.P.A. Corney, B.F. Buxton, W.B. Langdon, and D.T. Jones. BioRAT: extracting biological information from full-length papers. *Bioinformatics*, November 22 2004.
4. Francisco M. Couto, Mario J. Silva, and Pedro Coutinho. ProFAL: PROtein Functional Annotation through Literature. In *JISBD*, pages 747–756, 2003.
5. H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the 40th Anniversary Meeting of the ACL*, 2002.
6. R. R. Gabdoulline, R. Hoffmann, F. Leitnern, and R. C. Wade. ProSAT: functional annotation of protein 3D structures. *Bioinformatics*, 19(13):1723–1725, 2003.
7. Takeshi Kawabata, Motonori Ota, and Ken Nishikawa. The protein mutant database. *Nucleaic Acid Research*, 27(1), 1999.
8. A. Marchler-Bauer, A. R. Panchenko, B. A. Shoemaker, P. A. Thiessen, L. Y. Geer, and S. H. Bryant. CDD: a database of conserved domain alignments with links to domain three-dimensional structure. *Nucleic Acids Research*, 30(1):281–283, 2002.
9. Hans-Michael Müller, Eimear E. Kenny, and Paul W. Sternberg. Textpresso: An Ontology-Based Information Retrieval and Extraction System for Biological Literature. *PLoS Biology*, 2(11):1984–1998, November 2004. `www.plosbiology.org`.
10. W. R. Pearson and D. J. Lipman. Improved tools for biological sequence comparison. *Proc. of the National Academy of Sciences of the USA*, 85(8), 1988.
11. I. Schomburg, A. Chang, C. Ebeling, M. Gremse, C. Heldt, G. Huhn, and D. Schomburg. BRENDA, the enzyme database: updates and major new developments. *Nucleic Acids Research*, 32, 2004.
12. J. D. Thompson, D. G. Higgins, and T. J. Gibson. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, positions-specific gap penalties and weight matrix choice. *Nucleic Acids Research*, 22(22):4673–4680, 1994.
13. René Witte. An Integration Architecture for User-Centric Document Creation, Retrieval, and Analysis. In *Proceedings of the VLDB Workshop on Information Integration on the Web (IIWeb)*, pages 141–144, Toronto, Canada, August 30 2004.

# A Semi-automatic Approach to Extracting Common Sense Knowledge from Knowledge Sources

Veda C. Storey[1], Vijayan Sugumaran[2], and Yi Ding[1]

[1] Department of Computer Information Systems, J. Mack Robinson College of Business
Georgia State University, Box 4015, Atlanta, GA 30302
vstorey@gsu.edu, yding1@cis.gsu.edu

[2] Department of Decision and Information Sciences, School of Business Administration
Oakland University, Rochester, MI 48309
sugumara@oakland.edu

**Abstract.** Common sense knowledge based systems are developed by researchers to enable machines to understand ordinary knowledge and reason intelligently as a human would. The knowledge repositories of such systems are usually developed manually by a knowledge engineer or by users. Building a knowledge base of common sense knowledge such as that possessed by an average human being would be a very time-consuming, if not impossible, task. Some aspects of real world knowledge have already been captured and organized into various repositories such as the World Wide Web, WordNet, and the DAML ontology library. However, the extraction and integration of common sense knowledge from those sources remains a challenge. To address this challenge, an architecture for a Common Sense Knowledge Extractor is proposed that serves as an intermediary tool to extract common sense knowledge from several knowledge sources in order to develop a common sense repository. The design of the system as an extension of prior research on intelligent query processing is presented.

## 1 Introduction

There has been growing interest in capturing and using domain-specific and other forms of real-world knowledge to aid in conceptual modeling and query processing [4]. Knowledge is recognized as a significant organizational resource that is possessed in the mind of individuals. It is personalized information (which may or may not be new, unique, useful, or accurate) related to facts, procedures, concepts, interpretations, ideas, observations, and judgments [1].

Knowledge based systems have been developed to provide support for organizations to store, retrieve, and disseminate this resource (considered as intellectual capital) across internal groups of people. However, there are many challenges to doing so, the most important of which is the need to design knowledge retrieval strategies that provide timely and easy access to knowledge while avoiding information overload. "Intelligent" information systems that have a deeper and more meaningful understanding of common sense knowledge would aid in resolving knowledge management issues and have more effective query processing.

This research, then, is motivated by the need to build a base of common sense knowledge that will serve as a valuable resource for intelligent responses to queries

A. Montoyo et al. (Eds.): NLDB 2005, LNCS 3513, pp. 322–332, 2005.
© Springer-Verlag Berlin Heidelberg 2005

and for use in projects that are subsequent to the one where the knowledge is obtained. Although various attempts have been made to build such a base, they are not of practical use. One reason is that the systems are built manually, which is time-consuming, if not impossible. There are, however, many existing knowledge sources such as the World Wide Web, WordNet, and the DAML Ontology library.

The objective, of this research, therefore, is to: *develop an intermediary common sense tool with functional and data access interfaces to existing sources of knowledge*. Since the existing sources have different knowledge representations, interfaces to each are needed. The contribution of the research is to provide a way to develop a common sense knowledge base and populate it in a semi-automated way. The research should also provide some insights into how to utilize and integrate existing knowledge sources and reasoning techniques to build up a practical base of common sense knowledge. Specifically, this research builds upon our prior research on intelligent query processing to augment an existing system that incorporates different knowledge forms, namely, ontologies, a natural language lexicon and user profiles. An architecture is proposed for obtaining common sense knowledge as extracted from multiple, existing knowledge sources.

## 2 Related Work

### 2.1 Common Sense Knowledge

Common sense knowledge refers to the kinds of facts, ideas, and theories that most of us know [12]. However, most systems lack common sense knowledge. This is particularly obvious in the area of query processing. Consider, for example, the famous query [2].

> Mom needs to have a series of physical therapy sessions. Biweekly or something…. Set up the appointments

Common sense knowledge would predict, for example, that if Mom is a residence of the United States, then, it is important to find out what type of health insurance plan Mom has and whether the provider is "within network." The phrase "within network" has a specific meaning, namely, that the medical provider and insurance company have an agreement regarding which services are offered and the price structure for these services. If Mom resides in Canada, however, then, the selection of medical personal is not restricted by an insurance company and it would be important to first check for availability.

Some common sense may be explicitly coded into systems. Systems can identify explicit mistakes such as invalid numbers or dates. More work is required for systems to identify implicit mistakes such as scheduling a concert at 3:30 am or sending a birthday gift card for Christmas. With enough common sense knowledge and skills, and methods for applying the knowledge, computer systems would be much more intelligent, not only to identify implicit mistakes from user input but also to understand more complicated information from sources such as reports or books. Common sense reasoning was originally proposed to make a computer system reason as a child does [11]. However, common sense reasoning still faces challenging issues such as

the enormous amount of knowledge required, the complexity of the structure of natural language and understanding processes [10].

### 2.2 Common Sense Knowledge Projects

There have been several attempts to build up a common sense reasoning system. The CYC project [7], for example, was developed with the goal of enabling a variety of knowledge-intensive products and services by constructing a foundation of basic "common sense" knowledge. The ThoughtTreasure project [17] was developed with a common sense knowledge base and architecture for natural language processing that uses multiple representations including logic, finite automata, grids, and scripts. Its knowledge base contains 51,000 common sense assertions and about 27000 concepts. Most of those knowledge pieces in ThoughtTreasure were added by the project developers [17].

WordNet is an online lexical reference system whose design is based on theories of human lexical memory [20]. WordNet contains separate databases for nouns, verbs, adjectives, and adverbs, represented as lexicons [5]. It lacks a root concept and top-level ontology for organizing all concepts. For example, there is no direct connection between the concept of "Dog" and concept of "Pet" although a dog is associated with pet in most situations.

The Open Mind Common Sense Project [13] builds a common sense knowledge base by involving public users. To date, it has collected 300,000 concepts and 1.6 million binary-relational assertions. Part of this project is the ConceptNet, a semantic network with its framework extended from WordNet. ConceptNet, as an open source project, also consists of a set of tool-kits for researchers to access its ready-to-use knowledge base, which extracted its knowledge from Open Mind Common Sense Corpus.

Although those existing common sense knowledge projects have been developed, much more progress is needed for a practical use of common sense knowledge. This is especially true since many of these projects have had a large manual component However, there are still a large number of existing knowledge sources such as the World Wide Web. A great deal of common sense information is contained in those semi-structured or free text web pages, or relational databases, which contains its own implicit common sense information (concepts, relations among attributes). Acquiring common sense knowledge from these sources should be more effective than the current, manual effort required to accumulate a useful amount of common sense knowledge.

### 2.3 Ontologies

There has been much work carried out on ontology development as a way to capture and organize knowledge about the real world. In fact, it has been proposed that ontologies will be most effective in providing the needed domain-specific knowledge to process queries effectively and perform other tasks [4]. An ontology generally con-

sists of terms of an application domain and the relationships among them [6], although many different definitions and applications of ontologies exist [18].

Consider, for example, the auction domain ontology that has been used previously in the development of an ontology management system for database [16], and shown in Figure 1. The ontology captures the standard requirements of an ontology, namely, the terms of a domain and the relationships between them. Included also are explicit mention of the business rules for this application domain and the constraints that exist in that domain. These business rules and constraints provide indications of what a useful set of common sense knowledge might be. Thus, examples of the accompanying common sense knowledge might be:

- Required (account, transactions) (An account is required for a transaction.)
- Required (registered, bid) (One must be registered to bid).
- Not equal (buyer, seller, item) (For an item, buyer and seller must be different.)
- Required (item, bid) (For a bid, one must have an item).

**Table 1.** Partial Acution Domain Ontology.

| Term | Synonym | Description | Business Rule | Related to |
|---|---|---|---|---|
| Item | Product | Bought and sold and bid on | | Category, Customer, Shipper |
| Category | | Classification of Product | | Item |
| Customer | Buyer, Seller | Person who buys or sells | | Item, Account |
| Account | | Prerequisite for Transaction | Customer Needs to open account for transaction | Customer |
| Shipper | Vendor | Company that delivers the items to the buyer | | Item |
| Bid | auction price | Current price willing to pay | | Item, Customer |
| Bidder | Buyer Customer | Bids for item or product on sale | Has to be registered to bid | Buyer, Customer, Item |
| Buyer | Customer | Buys item or product that is for sale | | Item |
| Buy | Purchase | Buy or acquire and Item | Inverse transaction of Sell | Customer, Buyer, Item |
| Sell | Liquidate | Selling Items | Inverse of Buy transaction | Buyer, Customer, Item |
| Seller | Customer | Puts Item up for sale | | Item |
| Shipper | Vendor | Company that delivers items | | Item |

| Term | Pre-requisite Constraint | Temporal Constraint | Mutually Inclusive Constraint | Mutually Exclusive Constraint |
|---|---|---|---|---|
| Bid | Item, Account, Bidder | Account | Item, Buyer, Seller | |
| Seller | Item, Initial Bid | Account | Item | Buyer |
| Buyer | Item, Bid | Bid | Bid | Seller |

### 2.4 Prior Research on Query Processing

Our prior research developed the ISRA system [15] that accepts a query from the user, augments it with knowledge in the form of subset/superset terms from an ontology library and WordNet. That research has demonstrated that different knowledge forms are needed to process queries that will provide more "intelligent" and appropriate responses to user queries. These knowledge forms are: 1) lexical knowledge (e.g., WordNet), 2) ontology and domain-specific knowledge, 3) personal, or user profile knowledge, and 4) common sense knowledge. For example, suppose one is attempting to process the following query on the World Wide Web:

"Find the names of good restaurants in Summerside."

To effectively process this query, the knowledge needed is:

- Lexical knowledge: synonym for "good" or "surrogate" for good (e.g., ratings by an agency, etc.)
- Ontological knowledge: restaurant is a place to eat. There are different forms of restaurants (family, café, cafeteria, etc.)
- Personal knowledge: The user is from Prince Edward Island, Canada. Therefore, Summerside, Oklahoma is irrelevant. Prince Edward Island is a small place. Therefore "good" will mean "standard" or not very fancy. The user has children.
- Common sense knowledge: People who have children and are intending to take those children with them to the restaurant might like to consider a family restaurant. People who have a lot of children might want to make a reservation. Many family type restaurants do not take reservations.

Thus, without an understanding of what "good" means, it is difficult to process this query. It could refer to a rating by some agency or by the user based upon past experience or, possibly, on price. The exact location of Summerside (country, state, or province) is not specified. Searches on Google provide the results shown in Table 2. As expected, the number of hits dramatically decreases as the query is refined. Of course, some of the hits result in incorrect or undesired semantics. For example, with one keyword, a hit of Summerside in 'Oklahoma' is not relevant. Rather, "Summerside," a town in Prince Edward Island, Canada is the correct semantics.

**Table 2.** Query Results from Google.

| Search Keywords | Hits |
|---|---|
| Summerside | 163,000 |
| Summerside, restaurant | 9,900 |
| Summerside restaurant Prince Edward Island | 4,850 |
| Summerside restaurant Prince Edward Island gourmet | 127 |

## 3 Common Sense Knowledge Extractor as Intermediate Tool

This section proposes an architecture for a system that could be used to incorporate common sense reasoning by taking advantage of existing sources of common sense knowledge.

### 3.1 Purpose

The system will be an intermediary tool. The contribution, therefore, is to bridge existing knowledge sources and internal common sense knowledge based systems together. Unfortunately, existing knowledge bases are in a variety of formats so it is unlikely for an internal common sense knowledge base system to be able to extract the common sense knowledge directly from those sources. Thus, an intermediary tool that is compatible with these different formats could extract the knowledge from those sources. Figure 1 demonstrates the different sources of common sense knowledge and the need for an intermediate tool. It also takes a broad view of common sense knowledge. Included for example, are many diverse aspects such as emotion, self-reflection, self-imaging, and representation. This research focuses on extracting, using, and augmenting these existing sources and applying them to business applications.

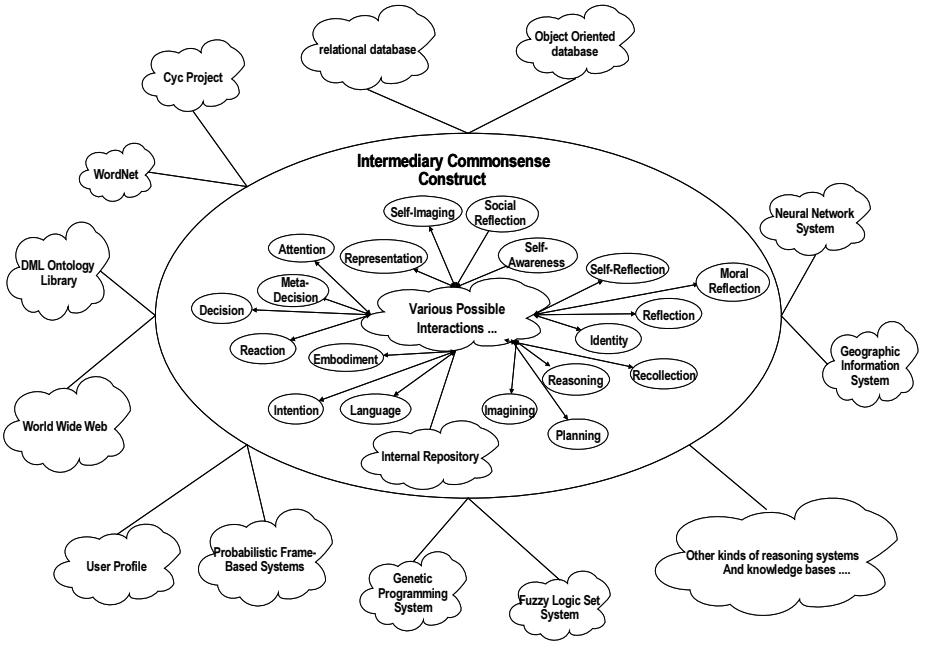



**Fig. 1.** Sources of Common Sense Knowledge.

### 3.2 Knowledge Extractor Tool Architecture

The common sense knowledge extractor will serve as an intermediary tool for generating a base of common sense knowledge. The sources of common sense knowledge appear in different formats: free-text, semi-structured, entity relationship based, formal logic based, semantic network based, etc. This tool will be added to the ISRA system in an attempt to provide an integrated solution to transfer appropriate knowledge/information from the knowledge sources into the common sense knowledge

repository. The architecture of the tool is shown in Figure 2 and consists of the following components: a) knowledge middleware and reasoning engine, b) common sense knowledge repository, c) repository maintenance tools, and d) repository APIs for applications, each of which is described briefly below.

### 3.2.1 Knowledge Middleware and Reasoning Engine

This component provides the interface to the different sources of common sense knowledge. Since the sources use different formats and representations, it is often difficult to combine the knowledge elements extracted from these sources in problem solving. The knowledge middleware component attempts to eliminate this problem by providing a unified interface. This knowledge middleware maintains meta-information about various knowledge sources in terms of how to access them, how to search for specific knowledge elements and convert them into a common representation to be used by a knowledge application. The inference engine facilitates knowledge resolution, i.e., integrating the various knowledge elements into a consistent set of common sense knowledge elements related to a particular topic. It also aids in resolving conflicts. The inference engine helps to derive new knowledge based on individual knowledge elements extracted from different sources.

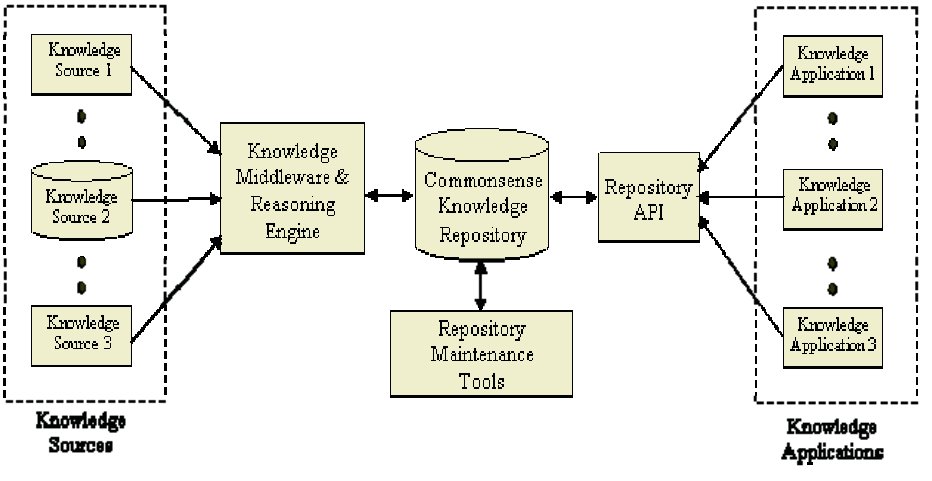



**Fig. 2.** Architecture for Common Sense Knowledge Extractor Tool.

E.g., Source 1: Auction items require bids before transaction (from auction ontology)
Source 2: Auction transaction requires account (from auction ontology)
Source 3: Account belongs to either a person or a company (from business process knowledge.
Inference: Person or company makes auction transaction

The system's common sense knowledge base system is the internal common sense knowledge repository. The outside systems are connected through the intermediary common sense tool's data and functional interfaces. Through these data interfaces,

the various knowledge pieces from outside knowledge sources can be extracted and transferred into the common sense knowledge. The functional interfaces enable the common sense knowledge retrieval commands to be transferred into the appropriate knowledge retrieval commands to the outside knowledge systems. This construct provides flexibility and scalability to allow any new knowledgebase systems to be added by extending the plug-in data and functional interfaces to support the knowledge access to that particular knowledge system.

#### 3.2.2 Common Sense Knowledge Repository

The common sense knowledge repository is similar to ConceptNet [9] whose framework is based on the extension of WordNet's semantic network. The lexicon node in WordNet has been extended into compound concepts (e.g. 'buy food', 'drive to store') in ConceptNet. The semantic relations in WordNet such as is-a and part-of have been extended into twenty semantic relations such as EffectOff (causality), SubeventOf (event hierarchy), CapableOf (agent's ability), etc. A major advantage of choosing ConceptNet as a model is that it has a natural language processing tool-kit that can provide the needed technical functions such as recognizing, tokenizing, tagging, chunking, lemmatizing, and semantically interpreting any input paragraph of text in either semi-structured or free text format, which are the dominant format for many real world documents.

#### 3.2.3 Repository Maintenance Tools

The common sense knowledge repository evolves over time with the repository maintenance tools enabling the user to ensure that the repository is consistent. These tools facilitate adding new knowledge to the repository, deleting existing knowledge elements, and updating knowledge elements as new knowledge is derived. The maintenance tools also allow users to transform knowledge elements from one representation to another so that they can be used in various applications.

#### 3.2.4 Repository APIs

The repository APIs (Application Program Interfaces) provide the interface for the common sense knowledge repository so that knowledge applications can effectively use the repository. The APIs include standard methods for accessing different parts of the repository, searching and retrieving relevant knowledge elements and integrating these knowledge elements into the application by converting them into the specific format needed within the application. For example, when searching for concepts in the repository, the application can use boolean expressions or semi-structured expressions and perform pattern matching to retrieve appropriate and related concepts.

### 3.3 Development Process

The intermediary tool will be developed in three stages: 1) initial prototype 2) extension; and 3) evaluation. Initially, the World Wide Web will be used as an existing knowledge source to acquire common sense knowledge. The World Wide Web has

numerous development tools that can be used to facilitate the extraction process. These tools will be augmented by the following procedure.

- Retrieve the selected web page through web extension tools such as Google Web API service, which allow developers to query more than 8 billion web pages from their own computer program.
- Apply NLP rules such as regular expressions and syntactic-semantic constraints to extract common sense knowledge assertions in various forms such as binary relationships and facts.
- Refine the assertions through appropriate methods such as Normalization
- Produce additional 'intermediate' knowledge such as semantic and lexical generalizations by applying heuristics that will improve the connectivity of the knowledge base
- Further mine the Web page content and extract the common sense knowledge pieces by performing inferencing.
- Store the extracted common sense knowledge into the internal knowledge base.

To validate the process, it will be integrated with the ISRA system which has been already implemented and assessed [15].

## 4 Examples

It has long been an assumption that common sense knowledge can be effectively applied to a variety of applications and problems. Since the ISRA system focuses on query processing and the WWW, Table 3 illustrates some examples that show how the addition of common sense knowledge could help processing queries on the web.

**Table 3.** Query Results with Simulated Common Sense Knowledge.

| Sample Query | Google results | Common Sense knowledge missing | Common Sense Knowledge Added |
|---|---|---|---|
| Find oboe reeds | 120,000 | Oboe is woodwind instrument, reeds have quality, purchase required | 5,500 |
| Get sour recipe | 3,940,000 | Nutrition, soup type (e.g., tomato) | 199,000 |
| Find gourmet decaffeinated coffee | 80,800 | Buy coffee | 12,100 |

The above examples illustrate first that the number of search results can be greatly reduced by including a minimal amount of knowledge. Some of the knowledge is generic across the application domains. For example, "find" usually implies the desire to buy. That, in turn, suggests that "sales" is an important part of the query. It is possible that portions of the common sense knowledge might be embedded in ontologies. For example, the fact that an oboe is a woodwind instrument and, in turn, a double-reed instrument would be a typical subtype/supertype relationship one would expect

to find in an ontology. Similarly, there are various types of soup with different nutritional values.

## 5 Conclusion

This research has attempted to highlight the need for the incorporation of different types of knowledge that can capture and use common sense knowledge in information systems. An architecture for the development of a common sense knowledge module that would serve as an intermediary to capture and synthesize existing stocks of common sense knowledge has been proposed. The intermediary tool will augment an existing system to process more effectively "intelligent queries." Four types of knowledge forms have been identified as effective in extracting more useful queries: lexical, ontology, user profile and common sense knowledge. Further research is needed to implement the common sense module and integrate it with the existing system.

## References

1. Alavi, M. and Leidner, D. E., "Review: Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues," MIS Quarterly, vol. 25, pp. 107-136, 2001.
2. Berners-Lee, T., Hendler, J., and Lassila, O. "The Semantic Web," Scientific American, May 2001, pp. 1-19.
3. Calvanese, D., Giacomo, G., Lenzerini, M., Nardi Daniele, and Rosati, R., "Information Integration: Conceptual modeling and reasoning support," Conference on Cooperative Information Systems, 1998.
4. Embley, D., "Toward Semantic Understanding An Approach Based on Information Extraction Ontologies," Proceedings of the Fourteenth Australian Database Conference, Dunedin, New Zealand, 18-22 January, 2004, pp.3-12.
5. Fellbaum, C., "WordNet: An Electronic Lexical Database," MIT Press, 1998.
6. Gruber, T.R. "A Translation Approach to Portable Ontology Specifications," Knowledge Acquisition (5), 1993, pp. 199-220.
7. Lenat, D., "A Large-Scale Investment in Knowledge Infrastructure," Communication of the ACM, vol. 38, pp. 32-38, 1995.
8. Levy, A., "Logic-based techniques in data integration," presented at Workshop on Logic-Based Artificial Intelligence, Washington, DC, 1999.
9. Liu, H. and Singh, P., "ConceptNet - a practical common sense reasoning tool-kit," BT Technology Journal, vol. 22, 2004.
10. McCarthy, J., Minsky, M., and Sloman, A., "An architecture of diversity for common sense reasoning," IBM Systems Journal, vol. 41, 2002.
11. McCarthy, J., "Program with Common Sense," presented at Proceedings of the Symposium on Mechanizations of Thought Process, London, 1958.
12. Minsky, M., The Emotion Machine. New York: Pantheon, forthcoming.
13. OpenMind, http://commonsense.media.mit.edu/cgi-bin/search.cgi
14. Sowa, J. F., "The Challenge of Knowledge Soup," Vivo Mind Intelligence, Inc, 2004.
15. Storey, V.C., Burton-Jones, A., Sugumaran, V., and Purao, S. "Making the Web More Semantic: A Methodology for Context-Aware Query Processing," Working Paper, 2004.

16. Sugumaran, V. and Storey, V.C., “Ontologies for Conceptual Modeling: Their Creation, Use and Management,” Data and Knowledge Engineering, Vol.42, No.3, November 2002. pp.251-271.
17. ThoughtTreasure, http://www.signiform.com/tt/htm/tt.htm
18. Weber, R. "Ontological Issues in Accounting Information Systems," In Researching Accounting as an Information Systems Discipline, S. Sutton and V. Arnold (eds.), American Accounting Association, Sarasota, FL, 2002.
19. WordNet, http://www.cogsci.princeton.edu/~wn/
20. Miller, G.A., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K.J. "Introduction to WordNet: An On-line Lexical Database," International Journal of Lexicography (3:4), 1990, pp. 235-244.

# A Phrasal Approach to Natural Language Interfaces over Databases

Michael Minock

Department of Computing Science
Umeå University, Sweden 90187
Tel.: +46 90 786 6398 Fax: +46 90 786 6126
mjm@cs.umu.se

**Abstract.** This short paper introduces the STEP system for natural language access to relational databases. In contrast to most work in the area, STEP adopts a phrasal approach; an administrator couples phrasal patterns to elementary expressions of tuple relational calculus. This 'phrasal lexicon' is used bi-directionally, enabling the generation of natural language from tuple relational calculus and the inverse parsing of natural language to tuple calculus. This ability to both understand and generate natural language enables STEP to engage the user in clarification dialogs when the parse of their query is of questionable quality or is open to multiple interpretations. An on-line demonstration of STEP is accessible at http://www.cs.umu.se/~mjm/step.

## 1 Introduction

To get a grip on the problem of reliable natural language interfaces, we focus on the case in which the information of user interest is housed within relational databases and interaction is purely textual; user requests are single sentences of natural language and answers are multiple sentences of natural language. Clarification dialogs, when necessary, are limited to multiple choice, yes/no responses from the user. Historically projects with such assumptions have aroused great interest within the relational database and computational linguistics communities [1, 4] which has continued, albeit with less fervor, into the current period (see for example [2, 13, 14]). RENDEZVOUS [3] was perhaps the first project proposed[1] and, to the point here, Codd was very explicit about requiring his system to engage the user in clarification dialogs; single shot systems where requests are parsed to a formal language and then immediately applied to a back-end database were deemed likely to be misinterpreted and misprized by users. At the very least a system should be able to paraphrase the user's query back to them during answer presentation or ambiguity resolution. In short, *any practical natural language interface over databases must have query paraphrasing capabilities.*

[1] It is doubtful whether RENDEZVOUS was completed given the limited facilities of the time.

A. Montoyo et al. (Eds.): NLDB 2005, LNCS 3513, pp. 333–336, 2005.
© Springer-Verlag Berlin Heidelberg 2005

It would be wrong to claim that other projects did not address the paraphrasing problem, some did [6, 7]. But as approaches switched from semantic grammars to those using domain independent grammars [5], the focus on generating query paraphrases tended to be discarded. Though the idea of doing a full, bi-directional integration of domain independent grammars is appealing, one has to wonder if this is currently feasible for interfaces to databases; the problems of ambiguity in large scale grammars, of configuring mappings between domain independent logical form and database relations, of capturing idiosyncratic domain language and of making the whole system bi-directional give one pause.

In response to this, STEP adopts a phrasal approach to the configuration and maintenance of linguistic knowledge. Specifically an administrator authors a *phrasal lexicon* by coupling phrasal patterns to elementary expressions of a class of tuple calculus. This phrasal lexicon is used bi-directionally, enabling the generation of natural language from tuple relational calculus and the inverse parsing of natural language to tuple calculus. This ability to both understand and generate natural language enables STEP to engage the user in clarification dialogs when the parse of their query is of questionable quality or is ambiguous. The details of STEP are documented in a recent technical report [12] available at `http://www.cs.umu.se/~mjm/step`.

## 2 The Web Interface

STEP is currently about 10,000 lines of LISP code, run as a server and accessed via standard Internet browsers. STEP issues satisfiability queries to the SPASS theorem prover and relational queries to a PostgreSQL database. Recently WordNet [9] has also been integrated into the system. STEP supports concurrent users and has responses times in the neighborhood of 2 to 5 seconds for queries over a geography database [8]. Figure 1 shows the web-based interface. Note that the current database is the `Geography` database, which has a simple canned paragraph description. Users enter their requests on the input field and obtain answers in the area immediately below. Figure 2 shows the response to a user request that is open to multiple interpretations.

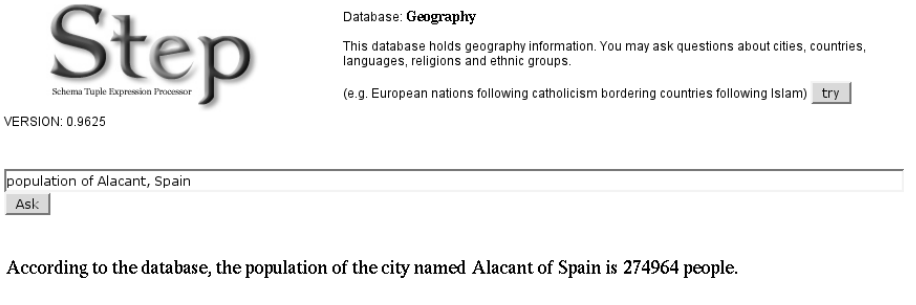



**Fig. 1.** Basic Querying.

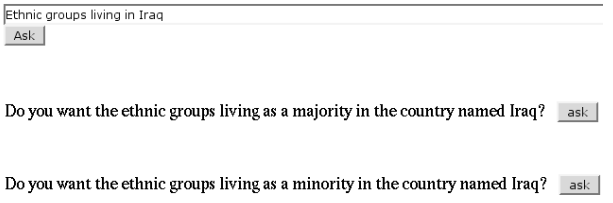



**Fig. 2.** A Clarification Dialog.

A full demonstration of STEP over a geography database, complete with a link to its complete configuration, has been continuously available for anonymous querying since June 2004. Over this period approximately 150 different visitors (myself not included) have issued in the neighborhood of 1000 queries to STEP. The purpose of this initial testing was not to carefully measure the accuracy of STEP, but was rather to build up a relatively large sample of real requests and in doing so help determine where system development efforts should be placed; naturally STEP has undergone extensive development and refinement over this period. Still, based on a cursory analysis of more recent system logs, STEP has done a reasonably accurate job of satisfying user requests. In the future more systematic studies will be undertaken over different domains to quantify this.

## 3 Discussion

The architectural of STEP is somewhat unusual. STEP does not use a domain independent grammar for syntactic analysis, but rather a phrasal lexicon authored specifically over the underlying database. Input sentences are parsed via a closed set of inference rules [12]. These inference rules employ a Montague like strategy where 'logical form' is incrementally built up as the input is scanned from left to right. These inference rules also cause special *fudging operations* which, at a cost, deliberately add, drop or alter words in the input sentence to help find a parse, albeit one of less confidence. To prepare for query paraphrasing, STEP compiles the phrasal lexicon into a subsumption hierarchy. Paraphrasing a given logical query comes down to semantically sorting the query into this hierarchy and gathering phrasal attachments of the sorted query's immediate parents [11].

'Logical form' in STEP consists of expressions in a class of tuple calculus with attached pragmatic features. The actual class of tuple calculus is decidable for emptiness, containment and equivalence [10]. Such capabilities are used in paraphrasing and in reasoning to support cooperative responses [11]. The relations used within these expressions are over database as well as pragmatic and conceptual relations. Such an extended vocabulary of relations makes it possible that STEP might provide meta level responses for queries over meaningful entities and relationships not (yet) covered in the underlying database.

The advantage of STEP's phrasal approach is that it avoids many of the difficulties associated with ambiguity in large scale domain independent grammars

and maps directly to the underlying database relations. Additionally, via fudging operations, STEP finds acceptable parses for many non-grammatical inputs, a common occurrence in practice. Finally a phrasal approach allows for easier specification of idiomatic and idiosyncratic domain language. A disadvantages of STEP's approach is that a specific phrasal lexicon must be authored for each new database. That said, we envision tools to assist in this process and we note that entries in the phrasal lexicon are relatively well structured and can be compiled into a subsumption hierarchy which makes their semantic relationships explicit. A question of course, is how well will STEP handle the syntactic complexities of real language over more complex databases. Of course it is still to early to answer this question fully, but the thesis here is that through better integrating large electronic dictionaries into the parsing process and through relying on clarification dialogs, STEP will ultimately work well enough to be practical.

## References

1. I. Androutsopoulos and G.D. Ritchie. Database interfaces. In R. Dale, H. Moisl, and H. Somers, editors, *Handbook of Natural Language Processing*, pages 209–240. Marcel Dekker Inc., 2000.
2. A. Blum. Microsoft english query 7.5: Automatic extraction of semantics from relational databases and OLAP cubes. In *Proc. of VLDB*, pages 247–248, 1999.
3. E. Codd. Seven steps to rendezvous with the casual user. In *IFIP Working Conference Data Base Management*, pages 179–200, 1974.
4. A. Copestake and K. Sparck Jones. Natural language interfaces to databases. *The Natural Language Review*, 5(4):225–249, 1990.
5. B. Grosz, D. Appelt, P. Martin, and F. Pereira. Team: An experiment in the design of transportable natural-language interfaces. *AI*, 32(2):173–243, 1987.
6. J. Ljungberg. Paraphrasing SQL to natural language. In *Proc. of RIAO 91*, Barcelona, 1991.
7. B. Lowden and A. de Roeck. The REMIT system for paraphrasing relational query expressions into natural language. In *Proc. of VLDB*, pages 365–371, 1986.
8. W. May. Information extraction and integration with FLORID: The MONDIAL case study. Technical Report 131, Universität Freiburg, Institut für Informatik, 1999.
9. G. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller. Five papers on WordNet. Technical report, Princeton University, Princeton, N.J, 1993.
10. M. Minock. Knowledge representation using schema tuple queries. In *Proc. of KRDB*, pages 51–62, Hamburg, Germany, 2003. IEEE Computer Society Press.
11. M. Minock. Modular generation of relational query paraphrases. *Journal of Language and Computation special issue on Formal Aspects of NLG*, 2005. To appear.
12. M. Minock. A phrasal approach to natural language access over relational databases. Technical Report 05.09, Umeå University, Umeå, Sweden, March 2005.
13. A. Popescu, O. Etzioni, and H. Kautz. Towards a theory of natural language interfaces to databases. In *Intelligent User Interfaces*, 2003.
14. B. Thalheim and T. Kobienia. Generating DB queries for web NL requests using schema information and DB content. In *Proc. of NLDB*, pages 205–209, 2001.

# Information Extraction for User's Utterance Processing on Ubiquitous Robot Companion*

Hanmin Jung[1], Choong-Nyong Seon[2], Jae Hong Kim[3],
Joo Chan Sohn[3], Won-Kyung Sung[1], and Dong-In Park[1]

[1] Information System Division, KISTI, 52 Eueon-dong, Yuseong-gu, Taejeon, Korea
jhm@kisti.re.kr
[2] Research Institute, DiQuest Inc., Korea
[3] Knowledge and Inference Research Team, ETRI, Korea

**Abstract.** Information Extraction originally tries to statically acquire information from various formatted documents in accordance with user-defined schema. Further, it can expand its application areas into the processing of purposeful user's utterance in natural language including various linguistic phenomena such as syntactic transformation and colloquial expression with frequently omitted words/phrases. We basically adopt verified lexico-semantic framework to obtain meaningful information from user's utterance and divide extraction phase into the two: the first is to extract and revise arguments and the other is to extract a predicate, which is an utterance meaning type of the input sentence.

## 1 Introduction

By interacting with human being, intelligent service robots understand human's utterance and emotion, and provide various services such as education, entertainment, and information activities. Particularly, user's utterance processing as a front-end system in the robot is critical to successfully grasp and react to the utterance meaning. We introduce information extraction technology to acquire meaningful information from user's utterance. Information Extraction originally tries to statically pick out information from various formatted documents in accordance with user-defined schema. Further, it can significantly expand its application areas into the analysis of purposeful user's utterance in natural language including various linguistic phenomena such as syntactic transformation and colloquial expression with frequently omitted words/phrases. It obviously overcomes the limitation of current robots' understandability[1]. POSIE [1], DQIE [3], and WHISK [4] are the information extraction systems that have ability to manage free texts. WHISK is prominent in comparison with other previous information extraction systems in that multi-slots can be extracted by the rules on unstructured documents. However, it should learn or describe all possible permutations between the slots within a rule because its rule is described in the way of tightly-coupled slot relations. It also requires additional syntactic analysis which

* This research was operated by *the project of Ubiquitous Robotics Companion technology development* as a national project by the Ministry of Information and Communication Republic of Korea.

[1] For example, AIBO understands only over 100 simple words and even users should learn the word list.

A. Montoyo et al. (Eds.): NLDB 2005, LNCS 3513, pp. 337–340, 2005.
© Springer-Verlag Berlin Heidelberg 2005

usually shows a low performance on free texts, and uses inefficient rule description without systematically-designed semantic knowledge. POSIE overcomes the defects of WHISK by introducing dynamic slot grouping and lexico-semantic approach. Despite it shows an outstanding performance over other systems for job offering and continuing education domains, POSIE still has a limitation related with multi-slots because its dynamic slot grouping supposes that the boundaries between two slots be shared, but the instances for each slot usually appear in extended order on free texts and colloquial sentences. DQIE mainly focuses on free texts, especially e-mails. It partially introduces POSIE framework and stratifies its rules into three levels for efficient rule description. DQIE showed that the performance on e-mails be over 92 for F-measure with only robust lexico-semantic pattern-based linguistic processing. Its variation for dynamic user's query analysis on car navigation domain also remarkably compensated speech recognition errors [2]. In this paper, we basically adopt verified DQIE framework to obtain meaningful information from user's utterance and update its rule structure to support intelligent robot services. We divide extraction phase into the two: the first is to extract and revise arguments and the other is to extract a predicate. A-extraction rules help to find all of the arguments by matching the input utterance with their own lexico-semantic patterns. Revision rules modify previously extracted arguments by referring contextual information. With P-extraction rules, we extract the predicate of the utterance.

## 2 URC: A Network-Based Intelligent Service Robot

URC (*Ubiquitous Robotic Companion*) is a network-based ubiquitous robot providing emotion-based interaction services and intellectual/physical labor services for human being regardless of his/her location. As Ubiquitous Robotic Companion adds network into conventional robots, it becomes possible to enhance mobility and human interface, and to make use of various user-friendly Web services for distributing its functions. It composes of *User Interface*, *User's Utterance Processing*, *Context Processing*, *Service Composition*, *Service Discovery*, and *Service Execution*. *User Interface*, including speech recognition and text-to-speech, interacts with users to accept their requests and to provide the result of internal processing. *User's Utterance Processing* analyzes user's question/command and generates a query structure, which is a formal description. The processing on information extraction framework is the focus of this paper. *Context Processing* interprets user's context referring to situation board. *Service Composition* and *Service Discovery* establish a service request through inquiring of Web services brokers. *Service Execution* calls the best Web service or robot built-in API with the request made by *Service Composition*.

## 3 Information Extraction for User's Utterance Processing

### 3.1 System Requirements

Utterance processing is to extract meaningful information to establish a target description which would be used for the service planning of Ubiquitous Robotic Com-

panion. We define our system requirements as the three major functions: information extraction, conceptual mapping, and domain portability. Information extraction should be able to recognize and to distinguish named entities and field information (*We call them arguments.*). Conceptual mapping should separate the extracted information into an utterance meaning type (*We call it predicate which is utterance meaning type*) and arguments. Finally, new domains should be easily added into existing system in the manner of revising rules. To proficiently satisfy the above, we departmentalize extraction rules into A-extraction rules for acquiring arguments, revision rules for correcting the arguments, P-extraction rules for obtaining a predicate.

### 3.2 Lexico-Semantic Approach to Extracting Predicate-Arguments

A lexico-semantic pattern (LSP) is a sequence with lexicons, part-of-speech tags, and semantic tags to abstract input sentence. Semantic tags consist of the three types: conceptual words (*like "%weather"*), instances (*like "@weather"*), and arguments (*like "#weather"*). Lexico-semantic approach enhances the coverage of rules by information abstraction through many-to-one mapping between surface forms and a lexico-semantic pattern, and provides the way to process and to compensate utterances only with partial textual fragments not full texts [2]. Our system basically adopts the verified lexico-semantic framework of DQIE [3] to keep up its high extraction performance and robust linguistic processing. Named entity recognition discovers all possible semantic tags for each input word. All of the words without semantic tags, except verbs and adjectives for discrimination, are substituted with part-of-speech tags. Sentence-LSP transfer makes an LSP with lexicons, part-of-speech tags, and semantic tags for an input sentence. It makes easier to match with LSP-based rules such as A-extraction rules, revision rules, and P-extraction rules by abstracting the input sentence. Argument candidate extraction finds all possible ones from 22 arguments by matching with A-extraction rules. As the result of this match, surface forms corresponding with matched LSPs are replaced with arguments. As A-extraction rules do not refer surrounding contexts, extracted arguments sometimes need to be revised by the contexts. An argument or a part-of-speech tag is replaced with new argument in the case of matching with a revision rule. Predicate is the utterance meaning type of an input sentence. We extract a predicate from 54 ACTION and QUESTION predicates by referring context-based P-extraction rules.

### 3.3 Extraction Rules

For an efficient rule description to extract meaningful information, we separate rules into non-contextual and contextual. Non-contextual A-extraction rules do not see any left and/or right context, but describe argument itself while revision rules reference surrounding contexts. An A-extraction rule consists of an LSP which consists of lexicons, part-of-speech tags, and semantic tags (*without for arguments*), and a semantic tag for argument. After successful matching between an input LSP and the left-hand side of an A-extraction rule, the input would be assigned to its right-hand side, i.e. an argument. Contextual revision rules additionally introduce semantic tags

for arguments into their left-hand sides and operations for argument modification into their right-hand sides. As the separated rules are based on contextual information, we can accurately verify extracted arguments efficiently reduce the number of rule entries. A contextual P-extraction rule consists of an extended LSP and a predicate. It describes maximal context to acquire best-fitted predicate for the input.

## 4 Experimental Result

We manually constructed 2,116 Korean sentences, 6,726 domain named entities, and 545 rules on weather, traffic, and home automation domains as the first year task. For measuring a system performance, we experimented with the above sentences in the manner of closed-test, and acquired unexpected results for several sentences originated from synonyms and omitted postpositions. We currently compensate the results in *Service Composition* by introducing additional predicate-arguments which make the same service query as original predicate-argument.

## 5 Conclusion

We applied linguistic processing based on lexico-semantic patterns into user's utterance processing for Ubiquitous Robotic Companion. Our patterns cover lexical to semantic matching, and promise greater adaptability due to the hybrid linguistic analysis and the pattern-matching characteristics. The LSP-based linguistic processing does not require deep analysis that sacrifices robustness and flexibility, but sufficiently handles delicate natural languages. We basically adopted verified DQIE framework and functionally divided its rules into the four: A-extraction rules, revision rules, and P-extraction rules to easily port into other domains and to epochally manage dynamic user's utterance.

## References

1. Jung, H., Yi, E., Kim, D., and Lee, G.: Information Extraction with Automatic Knowledge Expansion. Information Processing and Management, Vol. 41, No. 2 (2005)
2. Jung, H., Min, K., Kim, W., Sung, W., Park, D.: Query Analysis Using Context-Based Information Extraction on Navigation Domain. Proceedings of the 30th Annual Conference of the IEEE Industrial Electronics Society (2004)
3. Min, K., Jung, H., Seo, J.: Context-based Information Extraction on E-mail Texts. Proceedings of Asia Information Retrieval Symposium (2004)
4. Soderland, S.: Learning Information Extraction Rules for Semi-structured and Free Text. Machine Learning, Vol. 34 (1999)

# Investigating the Best Configuration of HMM Spanish PoS Tagger when Minimum Amount of Training Data Is Available*

Sergio Ferrández and Jesús Peral

Grupo de Investigación en Procesamiento del Lenguaje y Sistemas de Información
Departamento de Lenguajes y Sistemas Informáticos
University of Alicante, Spain
{sferrandez,jperal}@dlsi.ua.es

**Abstract.** One of the important processing steps for many natural language systems (information extraction, question answering, etc.) is Part-of-speech (PoS) tagging. This issue has been tackled with a number of different approaches in order to resolve this step. In this paper we study the functioning of a Hidden Markov Models (HMM) Spanish PoS tagger using a minimum amount of training corpora. Our PoS tagger is based on HMM where the states are tag pairs that emit words. It is based on transitional and lexical probabilities. This technique has been suggested by Rabiner [11] –and our implementation is influenced by Brants [2]–. We have investigated the best configuration of HMM using a small amount of training data which has about 50,000 words and the maximum precision obtained for an unknown Spanish text was 95.36%.

## 1 Introduction

Tagging is the task of classifying words in a natural language text with a respect to a specific criterion. PoS Tagger is the basis of many higher level natural language processing task. There are some statistical [2, 9, 10, 12] and knowledge-based [4, 7] implementations of PoS taggers, also there are some systems that combine different methods with a voting procedure [8]. Our implementation follows Brants [2].

One of the main sources of errors in Natural Language Systems is the incorrect resolution of PoS ambiguities (lexical, morphological, etc.). HMM as presented by Rabiner [11] are the standard statistical approach to try to properly resolve such ambiguities.

Unambiguously tagged corpora are expensive to obtain and require costly human supervision. Consequently, the main objective of this paper is to study the behavior of HMM applied to PoS tagging using a minimum amount of training data.

The next section shows our approach. Section 3 presents the evaluation of our approach and Section 4 shows some conclusions and further work.

* This research has been partially funded by the Spanish Government under project PROFIT number FIT-340100-2004-14.

A. Montoyo et al. (Eds.): NLDB 2005, LNCS 3513, pp. 341–344, 2005.
© Springer-Verlag Berlin Heidelberg 2005

## 2 Our Approach: The HMM PoS Tagger

Tagging is the task of marking words in natural language text, the tagger has to choice a tag to unknown word from a defined finite tag set.

The forward-backward algorithm is used for unsupervised learning and relative frequencies can be used for supervised learning . The Viterbi algorithm [14] is used to find the most likely sequence of states for a given sequence of tags. Our implementation follows Brants [2].

The parameters of our model are initial state probabilities, state transition probabilities and emission probabilities: (a) $\pi_i$ is the probability that a complete sentence starts at state $s_i$, $\pi_i = P(q_1 = s_i)$, where $q_1$ is an initial state. (b) $g_i(k)$ is the emission probability of the observed word $w_k$ from state $s_i$, $g_i(k) = P(w_k = s_i)$. (c) $a_{ij}$ is the transition probability from state $i$ to state $j$, $a_{ij} = P(q_{t+1}|q_t = s_i)$, where $q_t$ is the state in time $t$.

We have to compute this probability estimation for each initial state, transition probability or emission probability in the training corpora in order to generate the HMM.

Given a complete sentence (sequence of words $w_1...w_n$), we want to look for the sequence of tags $T^*$ that maximizes the probability that the words are emitted by the model, we want to select the single most probable path through the model. This is computed using the Viterbi algorithm [14].

Mérialdo [9] shows that the Viterbi criterion optimizes sentence accuracy while the maximum likelihood criterion optimizes word accuracy. The maximum likelihood criterion selects the most probable tag for each word individually by summing over all paths through the model.

The next equation uses the Markov assumption that the transition and output probabilities only depend on the current state but not on earlier states.

$$T^* = arg \max_{t_1,...,t_n} \prod_{i=1}^{n} P(t_i|t_{i-1}, ..., t_1)P(w_i|t_i, ..., t_1) \tag{1}$$

$$\approx arg \max_{t_1,...,t_n} \prod_{i=1}^{n} P(t_i|t_{i-1}t_{i-2})P(w_i|t_i)$$

It is necessary to look for estimations that assign a part of the probability mass to the unseen events. To do so, there are many different smoothing techniques, all of them consisting of decreasing the probability assigned to the seen events and distributing the remaining mass among the unseen events. In our tagger the transition probabilities are smoothed using deleted interpolation [2].

$$P(t_i|t_{i-1}t_{i-2}) = \lambda_1\widehat{P}(t_i) + \lambda_2\widehat{P}(t_i|t_{i-1})\lambda_3\widehat{P}(t_i|t_{i-2}t_{i-1}) \tag{2}$$

For unknown words, a successive abstraction scheme is employed which look at successively shorter suffix. We borrow the calculation of the parameter $\theta$ from Brants and use a single context-independent value.

$$P(t|c_{n-i+1}, ..., c_n) = \frac{\widehat{P}(t, c_{n-i+1}, ..., c_n) + \theta P(t, c_{n-i+2}, ..., c_n)}{1 + \theta} \tag{3}$$

## 3 Evaluation

To realize the implementation of a PoS Tagger based on HMM we have to design tag set that show the syntactic category of a word in the context of a sentence and outstanding morphological information (our approach has 39 tags). We assume a trigram model, the states represent pairs of tags.

In order to ascertain the best configuration of HMM Spanish PoS Tagger, we have carried out different experiments when a minimum amount of training corpora is available.

### 3.1 Training Phase

Our system has been trained using a fragment of the *Lexesp* (CLiC- TALP Corpus) corpora [6] which contains 43 Spanish fragments (about 50,000 words) from different genres and authors. These corpora have been supervised by human experts and they belong to project of "Departamento de Psicología de la Universidad de Oviedo" and have been developed by "Grupo de Lingüística Computacional de la Universidad de Barcelona" with the collaboration of "Grupo de Procesamiento del Lenguaje de la Universidad Politécnica de Cataluña". Having worked with different genres and disparate authors, we felt that the applicability of our HMH PoS Tagger to other texts is assured.

In the experiments done using ten fold cross-validation a precision of **94.67%** was obtained. In order to find out the best precision of HMM Spanish PoS Tagger, we have done experiments using different values of parameters of equations used to calculate the probabilities of unknown words (see equation 3).

The best results for the different experiments, with a precision of **96.03%**, have been obtained when using 4 as a maximum suffix length and 0.5 as suffix backoff theta.

### 3.2 Evaluation Phase

In this section we have evaluated our approach. This task has been done using an unknown Spanish text which has about 10,000 words obtaining a precision of **95.36%**.

We have done experiments expanding the tag set until 259 which contains more morphological information that we will be useful for many Natural Language Systems. Obviously, we have observed that the precision diminish until 94.86%.

Generally, the obtained precision of the statistical models is between 95% and 97% (for example Freeling [5], Brill's Tagger [3], and Padró [10] for Spanish and TreeTagger [13] for English); on the other hand, the knowledge-based implementations (MACO [1] for Spanish) have a precision higher than 97%. Therefore, our approach proves to obtain a competitive result with a minimum corpora.

## 4 Conclusions and Further Work

We have proposed a Spanish Pos Tagger based on HMM that obtains competitive results using a minimum amount of training corpora which has 50,000 words.

Our computational system can be integrated inside other Natural Language Applications, due to PoS tagging is the basis of many higher level NLP tasks.

The main advantage of our Spanish tagger is to be able to obtain admissible results using a small training data. The evaluation result has been a precision of 95.36% using an unknown Spanish text.

As a future aim, we want to perform PoS Tagger of other languages (English, Italian, German, etc.) using the same core of HMM. Also, we plan to expand the morphological information of tags (with the use of dictionaries and rules).

## References

1. J. Atserias, J. Carmona, I. Castellón, S. Cervell, M. Civit, L. Márquez, M.A. Martí, L. Padró, R.Placer, H. Rodríguez, M. Taulé, and J. Turmo. Morphosyntactic Analysis and Parsing of Unrestricted Spanish Text. *First International Conference on Language Resources and Evaluation, LREC'98*, pages 1267–1272, 1998.
2. T. Brants. Tnt- a statistical part-of-speech tagger. *Proceedings of the 6rd Conference on Applied Natural Language Procesing, ANLP*, pages 224 – 231, 2000.
3. E. Brill. Transformation-based error-driven learning of natural language: A case study in part of speech tagging. *Computational Linguistics*, 21:543 – 565.
4. E. Brill. A corpus-based Approach to Language Learning. 1993.
5. X. Carreras, I. Chao, L. Padró, and M. Padró. Freeling: An open-source suite of language analyzers. *Proceedings of the 4th International Conference on Languge Resources and Evaluation, LREC'04*, pages 1364 – 1371, 2004.
6. M. Civit. Criterios de etiquetación y desambiguación morfosintáctica de corpus en español. *PhD thesis, Linguistics Department,Universitat de Barcelona*, 2003.
7. W. Daelemans, J. Zavrel, P. Berckand, and S. Gillis. A memory-based part-of-speech tagger generator. *Proceedings of the 4th Workshop on Very Large Corpora*, pages 14 – 27, 1996.
8. G. Figuerola, F. Zazo, E. Rodríguez, and J. Alonso. La Recuperación de Información en español y la normalización de términos. *Revista Iberoamericana de Inteligencia Artificial*, VIII(22):135 – 145, 2004.
9. B. Mérialdo. Tagging English text with a probabilistic model. *Computational Linguistics*, 20(2):155 – 171, 1994.
10. M. Padró and L. Padró. Developing Competitive HMM PoS Taggers Using Small Training Corpora. *ESPAÑA for NATURAL LANGUAGE PROCESSING, EsTAL*, pages 127 – 136, 2004.
11. L.R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257 – 286, 1989.
12. A. Ratnaparkhi. A maximum entropy part-of-speech tagger. *Proceedings of the 1st Conference on Empirical Methods in Natural Language Procesing, EMNLP*, pages 16 – 19, 1996.
13. H. Schmid. TreeTagger — a language independent part-of-speech tagger. *Institut fur Maschinelle Sprachverarbeitung, Universitat Stuttgart*, 1995.
14. A.J. Viterbi. Error bounds for convolutional codes and asymptotically optimal decoding algorithm. *IEEE Transactions on Inf. Theory*, pages 260 – 269, 1967.

# An Approach to Automatic Construction of Lexical Relations Between Chinese Nouns from Machine Readable Dictionary

Yi Hu, Ruzhan Lu, and Xuening Li

Department of Computer Science and Engineering
Shanghai Jiaotong University, Shanghai, China, 200030
{huyi,rz-lu,xuening_li}@cs.sjtu.edu.cn

**Abstract.** In this paper, a machine readable dictionary is utilized to acquire Chinese noun pairs satisfying five lexical relations. For low accuracy of current Chinese parser, our method is different from the traditional ones that use parsing firstly. The new method is designed to be a three-step procedure. Firstly, it annotates the paraphrase of some nominal entries that are used as training data. Secondly, patterns that denote lexical relations between nouns are defined and the applicability of the patterns is learnt from training Maximum Entropy model. At last, these patterns are applied to the remaining portion of the dictionary. A relatively satisfying result is achieved.

## 1 Introduction

In Chinese, words lack in morphological variety, the liberty of conceptual combination and the ambiguity of syntax structure are all more obvious than Indo-languages. For instance, nouns in Chinese keep their morphologies in all cases even when they are used as adjectives. This determines that constructing relationships between words is a very important task to build up semantic resources in Chinese.

WordNet [1], as a lexical database, attempts to capture a psychological model of the semantic interconnection within English words and has received the most attention in NLP. In HowNet[2], a Chinese word is expressed by some "atomic" semantic items. These resources were all created by hand. Another effort [3] is to automatically extract semantic information from on-line dictionaries, which can be divided into two steps. The first is parsing the definition in the dictionary. The second is using the syntactic information to improve the accuracy of pattern identification.

Because of the low accuracy of current Chinese parser, the usual two-step methods are not adapted. Our method is a new three-step procedure. Firstly, the paraphrase of some nominal entries is annotated by the relations and linguistic features in context. Secondly, patterns that denote lexical relations are defined and their applicability is learnt from training Maximum Entropy (ME) model. At last, these patterns are applied to the dictionary. We pay sufficient attention to the patterns and explore the conditions of applying these patterns by ME. In our work, Chinese parser only assists in extracting features in context.

A. Montoyo et al. (Eds.): NLDB 2005, LNCS 3513, pp. 345–348, 2005.
© Springer-Verlag Berlin Heidelberg 2005

## 2 Lexical Knowledge

There are two categories of lexical knowledge: intensional attributes and lexical relations. Intensional attributes compose the natural meaning of a word. For example, to the word "man", its intensional attributes ought to include <adult>, <male>, etc. Lexical relations represent the interrelations of words, and they are the beginning of automatic acquirement of lexical knowledge.

In this paper, a pattern-based method is used to automatically extract five basic lexical relations between Chinese nouns, which are hypernymy, hyponymy, synonymy, antonymy and meronymy. WordNet provides classical relations (examples in Chinese): hypernymy ("食品" (food)/"面包" (bread) ), hyponymy ("羌族" (Qiang clan)/"少数民族"(minority)), meronymy ("身体"(body)/"腿"(leg)), antonymy ("胜利"(success)/"失败" (failure)), and synonymy ("计算机"(computer)/"计算机"(computer)). Because the data of the other lexical relations is sparse, we just deal with the five basic lexical relations between Chinese nouns.

## 3 Pattern Description

### 3.1 Formal Definition

For instance, the definition of "红色 (redness) " is "像鲜血或石榴花一样的颜色 (like the color of blood or guava flower)". Here, "颜色 (color)" is the hypernymy word of "红色 (redness)". Thus, "像" + "一样" (like) can combine to form a pattern denoting the hypernymous relation. A pattern ***P*** in our work is formalized as:

$$[\#1]Part^{(1)}[\#2]Part^{(2)}.....[\#m]Part^{(m)}[\#m+1] \ . \qquad (1)$$

The description means that $P$ is composed of $m$ parts marked as $Part^{(loc)}$ (loc = 1~m). There exist m+1 positions tagged with [#Position] between the parts where the contextual features appear. "[#1]像$^{(1)}$[#2]一样$^{(2)}$[#3]" is an example.

### 3.2 Pattern Ambiguity

In order to improve the recall of extracting, our patterns do not strictly satisfy Chinese sense. So it is necessary to use a statistical method to decide whether a pattern is really suitable even if its form is matched (see the following example).

The definition of "情夫 (leman)" is "与已经有配偶的女子发生和保持不正当关系的男子 (a man who has and keeps illegal relation with a married woman)". There is a pattern "[#1]有$^{(1)}$ [#2]和$^{(2)}$ [#3]" that can be matched in this paraphrase, and this pattern usually shows that the two headwords (nouns) in position [#2] and [#3] might have hyponymous relation with the entry "情夫 (leman)". But "女子(woman)" and "男子(man)" can not be regarded as the hyponymy words of "情夫(leman)".

### 3.3 Disambiguating Function $\psi$

For using natural language is uncertain, it needs to decide whether a pattern $P$ is suitable for extracting according to its Contextual Feature Set ($\sigma$). We introduce the function $\psi$ for disambiguation:

$$\psi(P) = \arg\max_{c' \in \{c, \bar{c}\}} \Pr(c' \mid \sigma) \cdot \tag{2}$$

Where $c' \in \{c, \bar{c}\}$, $c$ means suitable, and $\bar{c}$ means unsuitable.

### 3.4 Contextual Features

The contextual features generally come from syntactic information and word forms, etc. Each feature reflects a simple fact of the language composition (LC) in current [#Position], so the types of feature value are usually Boolean. Partial features used in our work are listed in table 1:

**Table 1.** Partial Feature Examples.

| Feature Name | Feature Explanation |
|---|---|
| is_a_vp | whether current LC is a Chinese verb phrase |
| is_empty | whether current LC is empty |
| is_adv_yiban | whether current LC can be the adverb "一般(usually)" |
| Contain_dunhao | whether current LC contains punctuation "、" |

## 4 Disambiguating Strategy for Pattern Application

Determining the applicability of a patter $P$ can be treated as an issue of classification. In another word, if $\psi(P) = \mathrm{c}$, the pattern may be regarded as suitable in current context. Otherwise, unsuitable. We employ ME algorithm to solve this problem.

ME classification is a popular technique which has proven effective in many NLP application. The underlying philosophy is we should choose the model making the fewest assumptions about data while consistent with the foregone knowledge [4].

To pattern $P$, its estimate of $\Pr(c' \mid \sigma)$ takes the following exponential form:

$$\Pr(c' \mid \sigma) := \pi(\sigma) \cdot \exp(\sum_i \alpha_i F_i(c', \sigma)) \cdot \tag{3}$$

Where $\pi$ is a normalization function, which is calculated by (4):

$$\pi(\sigma) = (\sum_{c'} \exp(\sum_i \alpha_i F_i(c', \sigma)))^{-1} \cdot \tag{4}$$

For the importance of feature position that is sensitive to language, we introduce positional constraint to calculate the feature function $F_i$ for feature $f_i$ and class $c'$:

$$F_i(\tilde{c}, \sigma) := \begin{cases} 1, & \tilde{c} = c' \text{and } f_i \text{appears between Part}^{(a)} \text{and Part}^{(b)} \\ 0 & otherwise \end{cases} . \tag{5}$$

The $\alpha_i$ is feature weight parameter, and the definition of ME shows that the larger $\alpha_i$ is, the more strongly $f_i$ is considered to be an indicator to class $c'$.

## 5 Result and Conclusion

Our lexical resource is a machine readable dictionary, *Applied Chinese Dictionary*. 76 patterns are defined with statistical information and 11,696 noun pairs satisfy the five basic lexical relations. The precisions are listed in table 2.

**Table 2.** Precisions of Extracting The Five Lexical Relations.

| Type | Hyper. | Hypo. | Syno. | Anto. | Mero. |
|---|---|---|---|---|---|
| Preci. (%) | 85.2 | 73.6 | 77.9 | 91.3 | 63.0 |

Seen from the result, the algorithm achieves certain coverage. The precisions of hypernymy and antonymy are relatively high, which indicates that the two relations have steady context. To meronymy, the generalized ability of its patterns is limited for its sparse data and insufficient learning from contextual features.

Many noun pairs cannot be assigned a suitable relation yet. So our next work ought to further explore the pattern-based method. On the other hand, another significant work in future is to automatically extract intensional attributes of a word [5].

## Acknowledgement

This work is supported by National Natural Science Foundation of China (NSFC) (No.60496326) and National 863 Project (No. 2001AA114210-11).

## References

1. Miller, G.A., R. Beckwith, et al. 1990. Introduction to wordnet: An on-line lexical database. Journal of Lexicography, 3(4): 235-244.
2. Dong Zhendong. HowNet. http://www.keenage.com/
3. Jensen, K., and J.L. Binot. 1987. Disambiguating prepositional phrase attachments by using on-line dictionary definitions. Computational Linguistics, 13(3-4): 251-260.
4. Stephen Della Pietra, Vincent Della Pietra, John Lafferty. 1997. Inducing features of random fields. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(4): 380-393.
5. Lu Ruzhan, Jin Guangjin. 2003.The new angle of view to modern Chinese research. The third national workshop on language and literal application.

# Automatic Acquisition of Adjacent Information and Its Effectiveness in Extraction of Bilingual Word Pairs from Parallel Corpora

Hiroshi Echizen-ya[1], Kenji Araki[2], and Yoshio Momouchi[1]

[1] Dept. of Electronics and Information, Hokkai-Gakuen University,
S26-Jo, W11-Chome, Chuo-ku, Sapporo, 064-0926 Japan
Tel: +81-11-841-1161, Fax: +81-11-551-2951
{echi,momouchi}@eli.hokkai-s-u.ac.jp

[2] Graduate School of Information Science and Technology, Hokkaido University,
N14-Jo, W9-Chome, Kita-ku, Sapporo, 060-0814 Japan
Tel: +81-11-706-6534, Fax: +81-11-709-6277
araki@media.eng.hokudai.ac.jp

**Abstract.** We propose a learning method for solving the sparse data problem in automatic extraction of bilingual word pairs from parallel corpora. In general, methods based on similarity measures are insufficient because of the sparse data problem. The essence of our method is the use of this inference: in local parts of bilingual sentence pairs (*e.g.*, phrases, not sentences), the equivalents of words that adjoin the source language words of bilingual word pairs also adjoin the target language words of bilingual word pairs. Our learning method automatically acquires such adjacent information. The acquired adjacent information is used to extract bilingual word pairs. As a result, our system can limit the search scope for the decision of equivalents in bilingual sentence pairs by extracting only word pairs that adjoin the acquired adjacent information. We applied our method to two systems based on Yates' $\chi^2$ and AIC. Results of evaluation experiments indicate that the extraction rates respectively improved 6.1 and 6.0 percentage points using our method.

## 1 Introduction

Manual extraction by humans of bilingual word pairs of various languages is costly. For that reason, automatic extraction of bilingual word pairs from parallel corpora is effective. Many similarity measures [1] are used to extract bilingual word pairs automatically from parallel corpora with various languages because they are language independent. However, they are insufficient. That is, when several bilingual word pairs with close similarity values candidates exist, the system based on similarity measures falls into the sparse data problem. This problem is common among the methods based on similarity measures.

To overcome the sparse data problem, we use the hypothesis that, in local parts of bilingual sentence pairs (*e.g.*, phrases, not sentences), the equivalents of words that adjoin the **s**ource **l**anguage (SL) words of bilingual word pairs also

A. Montoyo et al. (Eds.): NLDB 2005, LNCS 3513, pp. 349–352, 2005.
© Springer-Verlag Berlin Heidelberg 2005

adjoin the **t**arget **l**anguage (TL) words of bilingual word pairs. Such adjacent information is effective to solve the sparse data problem. That is, the system can limit the search scope for the decision of equivalents in bilingual sentence pairs by extracting only those word pairs that adjoin the adjacent information.

Moreover, the adjacent information is acquired automatically for learning [2]. These features allow the application of our method to parallel corpora with various languages. We call this learning method **A**djacent **I**nformation **L**earning (AIL). In this paper, we applied AIL to two systems based on Yates' $\chi^2$ [3] and **A**kaike's **I**nformation **C**riterion (AIC) [4] to extract bilingual word pairs from parallel corpora. Evaluation experiments using parallel corpora with five different languages indicated that the extraction rates respectively improved 6.1 and 6.0 percentage points through the use of AIL. Therefore, we confirmed that AIL is effective to solve the sparse data problem in extraction of bilingual word pairs from parallel corpora.

## 2 Acquisition of Adjacent Information

The system obtains bilingual word pairs and templates using two bilingual sentence pairs. In this paper, the templates are rules that possess the adjacent information for extracting new bilingual word pairs. The system determines the templates using common parts between two bilingual sentence pairs. Moreover, the system assigns similarity values between SL words and TL words using the Dice coefficient for the obtained bilingual word pairs and templates. Figure 1 shows an acquisition example of adjacent information.

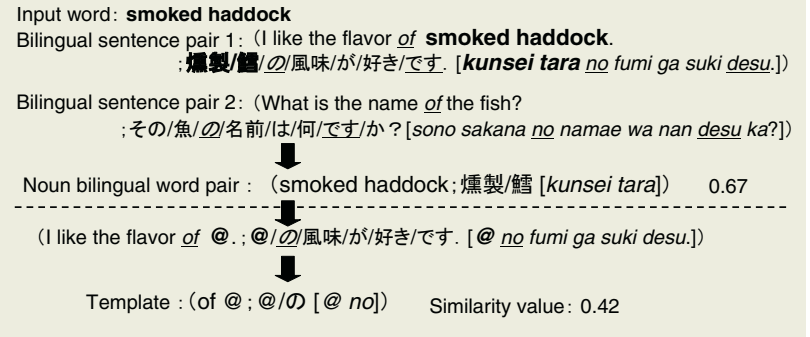



**Fig. 1.** An example of acquisition of adjacent information.

First, the system selects bilingual sentence pair 1, for which the SL word "smoked haddock" exists. Furthermore, the system selects bilingual sentence pair 2, for which "of" that adjoins "smoked haddock" exists and for which "の [*no*]" and "です [*desu*]" exist as common parts between two TL sentences. The system extracts "燻製/鱈 [*kunsei tara*]", which exists on the left side of the common part "の [*no*]", from the TL sentence of bilingual sentence pair 1. On the other hand,

"風味/が/好き [*fumi ga suki*]", which exists between "の [*no*]" and "です [*desu*]", is also extracted. However, it does not correspond to a noun, verb, adjective, adverb, or conjunction. Consequently, only (smoked haddock; 燻製/鱈 [*kunsei tara*]) is obtained as a correct noun bilingual word pair. Moreover, the system acquires the template (of @;@/の [*@ no*]) by replacing "smoked haddock" and "燻製/鱈 [*kunsei tara*]" with the variable "@" in bilingual sentence pair 1. Similarity values in (smoked haddock; 燻製/鱈 [*kunsei tara*]) and (of @;@/の [*@ no*]) are calculated using the Dice coefficient. The system chooses the most suitable bilingual word pairs and templates using their similarity values when several candidates of bilingual word pairs and templates exist.

The template (of @;@/の [*@ no*]) has the information that the equivalents of words that adjoin the right side of "of" exist on the left side "の [*no*]" in TL sentences. This fact indicates that the system using AIL can limit the search scope for the decision of equivalents in bilingual sentence pairs by extracting ONLY word pairs that adjoin the acquired templates. In contrast, the system without AIL must select correct bilingual word pairs from ALL bilingual word pairs that are nouns, verbs, adjectives, adverbs, and conjunctions in bilingual sentence pairs.

## 3 Performance Evaluation and Conclusions

Five kinds of parallel corpora were used in this paper as experimental data. These parallel corpora are for English – Japanese, French – Japanese, German – Japanese, Shanghai-Chinese – Japanese and Ainu – Japanese. They were taken from textbooks containing conversation sentences. The number of bilingual sentence pairs was 1,794. We inputted all 1,081 SL words of nouns, verbs, adjectives, adverbs, and conjunctions into the system based on Yates' $\chi^2$, the system based on Yates' $\chi^2$ in which AIL is applied (herein, we call it the system based on Yates+AIL), the system based on AIC and the system based on AIC+AIL. Initially, the dictionary for bilingual word pairs and the template dictionary are empty. We repeated the experiments for each parallel corpus using respective systems. In addition, we evaluated whether or not correct bilingual word pairs exist in the dictionary and calculated the extraction rate for all SL words.

Table 1 shows experimental results. The respective extraction rates of the systems based on Yates+AIL and AIC+AIL were more than 6.1 and 6.0 percentage points higher than those of the systems based on Yates' $\chi^2$ and AIC. These results indicate that AIL is effective for both Yates' $\chi^2$ and AIC.

Moreover, in the systems based on Yates+AIL and AIC+AIL, the respective extraction rates of the bilingual word pairs for which the frequencies are 1 were more than 9.7 and 9.9 percentage points higher than those of the systems based on Yates' $\chi^2$ and AIC. This fact indicates that AIL is effective to solve the sparse data problem. In some erroneous bilingual word pairs extracted by systems without AIL, their frequencies are 1. The system without AIL extracted such erroneous bilingual word pairs because of the data sparseness problems. Therefore, improvement of the extraction rates of bilingual word pairs for which

**Table 1.** Results of evaluation experiments.

| SL | Yates' $\chi^2$ | Yates +AIL | AIC | AIC +AIL | Number of bilingual word pairs |
|---|---|---|---|---|---|
| English | 53.8% | 59.8% | 53.3% | 58.6% | 169 |
| French | 55.4% | 60.4% | 55.4% | 60.4% | 240 |
| German | 53.3% | 58.5% | 53.8% | 59.0% | 195 |
| Shanghai-Chinese | 57.6% | 62.5% | 58.3% | 62.9% | 264 |
| Ainu | 52.1% | 62.0% | 52.6% | 62.4% | 213 |
| Total | 54.7% | **60.8%** | 54.9% | **60.9%** | 1,081 |

the frequencies are 1 indicates that AIL is effective to solve the sparse data problem.

Among related works, one study [5] has acquired low-frequency bilingual terms using a bilingual dictionary and MT systems for measuring similarity. It is difficult to deal with various languages because of the use of large-scale translation knowledge. On the other hand, one study [6] used co-occurrence of words depending on the number of co-occurrence words and their frequency. That method is insufficient in terms of efficient extraction of bilingual word pairs. In contrast, AIL requires only a one-word string as the co-occurrence word, *i.e.* only "of." Moreover, AIL can extract bilingual word pairs even when the frequencies of the pairs of the co-occurrence words and the bilingual word pairs are only 1. Regarding methods [2, 7] for acquisition templates, such methods require similar bilingual sentence pairs to extract effective templates.

Future studies will apply our method to a multilingual machine translation system.

## References

1. Manning, C. D. and H. Schütze. 1999. Foundations of Statistical Natural Language Processing. MIT Press.
2. Echizen-ya, H., K. Araki, Y. Momouchi, and K. Tochinai. 2002. Study of Practical Effectiveness for Machine Translation Using Recursive Chain-link-type Learning. In *Proceedings of COLING '02*, pp.246–252.
3. Hisamitsu, T. and Y. Niwa. 2001. Topic-Word Selection Based on Combinatorial Probability. In *NLPRS'01*, pp.289–296.
4. Akaike, H. 1974. A New Look at the Statistical Model Identification. *IEEE Transactions on Automatic Control*, AC-19(6):716–723.
5. Utsuro, T., K. Hino, and M. Kida. 2004 Integrating Cross-Lingually Relevant News Articles and Monolingual Web Documents in Bilingual Lexicon Acquisition. In *Proceedings of COLING'04*, pp.1036–1042.
6. Kaji, H. and T. Aizono. 1996. Extracting Word Correspondences from Bilingual Corpora Based on Word Co-occurrence Information. In *Proceedings of COLING'96*, pp.23–28.
7. McTait, K. and A. Trujillo. 1999. A Language-Neutral Sparse-Data Algorithm for Extracting Translation Patterns. In *Proceedings of TMI'99*, pp.98–108.

# Text Mining from Categorized Stem Cell Documents to Infer Developmental Stage-Specific Expression and Regulation Patterns of Stem Cells

Hyun Seok Park[1,2], Min Kyung Kim[1], Eun Jeong Choi[1], and Young Joo Seol[2,3]

[1] Department of Computer Science and Engineering, Ewha University, 11-1 Daehyun-dong, Seodaemun-gu, Seoul 120-750, Korea
{neo,minkykim}@ewha.ac.kr, ejchoi@ewhain.net
[2] Institute of Bioinformatics, Macrogen Inc., Seoul 153-023, Korea
neutrian@macrogen.com
[3] School of Computer Engineering, Sejong University, 98 Gunja-dong, Gwangjin-gu, Gunja-dong, Seoul 143-747, Korea

**Abstract.** Exponentially increasing stem cell data provide means to elucidate the system level understanding of differentiation. Given the existing information on biological networks combined with huge amount of literature data, inferring stem cell information through scientific reasoning of data from on-line documents would get great attention. In this paper, we describe the STEMWAY system for combining known interaction informatics with text mining techniques. Especially, recent advances in natural language processing technique raise new challenges and opportunities for extracting valuable information from literature classified by the developmental stages of stem cells.

## 1 Introduction

Before stem cells can be used to treat patients, scientists need to learn the interaction or regulation patterns of genes in each developmental stage of stem cells. Information about publicly available protein-protein interactions and networks can be a partial clue to system-level understanding (e.g., DIP, http://dip.doembi.ucla.edu/, BIND, http://www.bind.ca/). However, as this information is manually compiled by experts, it stores only well-known interaction data and can not cope with most specific interactions or regulations such as stem cell differentiation. In the stem cell research field, there is an urgent need for an automatic system capable of extracting information not only from general interaction databases but also from the literature. In this paper, we present the STEMWAY system - a complete information extraction system, by integrating and customizing interaction data from various text resources and existing databases (e.g., SCDb, http://stemcell.princeton.edu/) concerned with stem cells.

A. Montoyo et al. (Eds.): NLDB 2005, LNCS 3513, pp. 353–356, 2005.
© Springer-Verlag Berlin Heidelberg 2005

## 2 Text Mining from Stem Cell Research Papers

Nowadays, huge amount of documents related to stem cell research are available on the Internet. Naturally, there is a potentially important marriage between interaction informatics and natural language processing technologies, and this synergy extends beyond the traditional realm of either technology to a variety of emerging applications. We have extracted 78,670 interactions from 121,597 stem cell related abstracts from Medline.

**Classification of Stem Cell Documents:** Initially, we had been using the "stem" word as a key for retrieving abstracts from Medline. But considering that the term "stem" is too abstract, we decided to use more specific combination of keywords such as "pluripotent", "multipotent", organism names and various tissue names to classify the documents. When a stem cell divides as shown in Figure 1, each new cell has the potential to either remain a stem cell or become another type of cell with a more specialized function. Thus, if we subcategorize the retrieved stem cell documents, we might get a clue, in regards to gene expression or regulation patterns or protein interaction mechanisms in a specific tissue or in a specific developmental stage. We have retrieved around 121,597 abstracts from Medline, and classified the documents into several categories. From our classified corpus, we have examined a number of most frequent verbs: "activate", "bind", "interact", "regulate", "encode", "function" and etc. Then, text mining has been carried out repetitively for each categorized document. The result is shown in table 1.

**Parsing Methodology:** Text mining methods range from term recognition to extraction of complex relationships of interaction between the proteins. It is expected that text mining in general will provide tools to facilitate the annotation

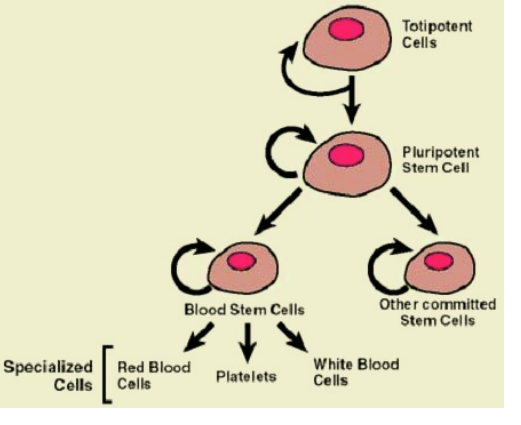



**Fig. 1.** Developmental Stages of Stem Cell (Source: http://www4.od.nih.gov/stemcell/fig2.gif)

**Table 1.** The number of interaction related abstracts and extracted interactions by cell types

| Cell Type | # of Abstracts | # of Interaction Related abstracts | # of Extracted Interactions |
|---|---|---|---|
| Human Pluripotent Cell | 1092 | 851 | 1941 |
| Human Multipotent Cell | 1092 | 577 | 1459 |
| Human Neural Cell | 51611 | 13540 | 21022 |
| Human Blood Cell | 22110 | 9474 | 18002 |
| Human Cardiac Cell | 1427 | 556 | 1301 |
| Mouse Pluripotent Cell | 752 | 404 | 819 |
| Mouse Multipotent Cell | 360 | 206 | 518 |
| Mouse Neural Cell | 6945 | 3098 | 5678 |
| Mouse Blood Cell | 32313 | 13407 | 24206 |
| Mouse Cardiac Cell | 3895 | 1799 | 3724 |

of vast amounts of interactions related to stem cell differentiation. The simplest way to extract protein relations from the literature is to rely on the pattern matching rules or shallow parsing [1, 2]. More promising candidates for a practical information extraction system would be those based on full-sentence parsing [3]. However, as full parsing techniques have some drawbacks, initially we have adopted a shallow parsing technology in place of traditional full parsing.

**User Interfaces:** The user interface of the STEMWAY system is to search the literature and automatically extract information from the abstracts and to provide two essential research support services: accelerating user tasks by partially

**Fig. 2.** Graphic representation of biological network extracted from stem cell documents; automatically extracted interaction data can be manually modified by the expert

automating the process of finding relations between genes and gene products and providing a convenient environment for researchers, annotating biological literature and visualizing the result. For example, automatically extracted interaction data are first displayed in a graphic form as shown in Figure 2 (left). When a node is clicked, specific information about the noun phrases (usually gene names) pops up. When an arch is clicked, the window with the original document is displayed from where relations have been deduced as shown in the figure (right). As relations are annotated automatically, interaction data can be manually modified by the expert when necessary, as shown in the figure (center).

## 3 Conclusion and Future Works

We have used STEMWAY to extract 78,670 interactions between human and mouse proteins from MEDLINE abstracts. To evaluate the quality of interaction data, five biological experts have manually reviewed randomly extracted protein interactions from 1000 source sentences, and founded that 71% of them were correct. We have tested our documents, not completely, but quite extensively, and concluded that MEDLINE is a unique source of diverse stem cell information, which can be extracted in a completely automated way with reasonably high accuracy. Moreover through the STEMWAY, we can choose and classify stage specific and/or organism specific genes. In conclusion, we think our research will be extremely useful in the future, especially in human and mouse microarray data analysis based on stem cell differentiation stages. As we are at the early stage of system development, there will be many modifications to current methods and even system architecture. By adopting relatively deep level analysis approach of the text in the future, we think that we can show more promising results

## Acknowledgments

The STEMWAY project has been partially supported by the Ministry of Information and Communication (IMT2000 ab05) and by the Ministry of Science and Technology ( $21^{st}$ Century Frontier Program of Stemcell Research & BK21) of Korea.

## References

1. Sekimizu,T., Park,H.S. and Tsujii,J.: Identifying the interaction between genes and gene products based on frequently seen verbs in MEDLINE abstracts. Genome Inform. **9** (1998) 62–71
2. Thomas,J., Milward,D., Ouzounis,C.A., Pulman,S. and Caroll,M.: Automatic extraction of protein interactions from scientific abstracts. Pac. Symp. Biocomput. **5** (2000) 541-552
3. Friedman,C., et al.: GENIES: a natural language processing system for the extraction of molecular pathways from journal articles. Bioinformatics **17** (2001) (Suppl. 1) S74-S82

# Simple But Useful Algorithms for Identifying Noun Phrase Complements of Embedded Clauses in a Partial Parse

Sebastian van Delden

Division of Mathematics and Computer Science
University of South Carolina Upstate
800 University Way, Spartanburg, SC, 29303, USA.
svandelden@uscupstate.edu

**Abstract.** Two algorithms for identifying noun phrase complements of embedded clauses in a partial parse are presented. The candidate noun phrases play subject or object roles in (reduced) relative and infinitival clauses. The algorithms are tested on several sources and results are presented.

## 1 Introduction

A detailed set of algorithms which identify noun phrase complements of embedded clauses can play a crucial role in many natural language processing (NLP) tasks[1]. These algorithms are intended for NLP systems which are organized as a linear arrangement of independent components acting in sequence on an input sentence to achieve their goals. Many such systems employ the following two initial components: 1) a part-of-speech tagger [2] which identifies the parts of speech of the words in an input sentence, and 2) a partial parser [3] which identifies clauses and phrases using syntactic and part-of-speech information. A partial parser creates a partial tree structure and avoids making difficult structural decisions such as prepositional phrase attachment or coordination disambiguation which some argue can only be accurately accomplished using complex semantic information. Partial parsers alone may provide enough linguistic information for the NLP system to accomplish its task. However, a more sophisticated system would follow the partial parser with another component which further processes the sentence. The algorithms presented in this paper form the next logical component which can lead to accomplishing several NLP tasks.

Sections two and three of this paper outline the relative and infinitival clause algorithms respectively, and section four concludes the paper with some results.

## 2 Relative Clauses

Figure 1 presents the complete algorithm for identifying noun phrase complements in three types of relative clauses. *RTYPE1* – the relative clause is missing its relative

[1] This work has been partially funded by the University of South Carolina Research and Productivity Scholarship Fund.

A. Montoyo et al. (Eds.): NLDB 2005, LNCS 3513, pp. 357–360, 2005.
© Springer-Verlag Berlin Heidelberg 2005

determiner (or pronoun), subject, and auxiliary verb(s). For example: *The ship ( lost in the battle ages ago ) was finally discovered. RTYPE2* – The relative clause is introduced by a relative determiner (or pronoun) and has no subject. For example: *The ship ( that was lost in the battle ages ago) was finally discovered. RTPYE3* – The relative clause is not introduced by a relative determiner (or pronoun) and has a subject. *The ship ( the British lost in the battle ages ago ) was finally discovered.*

A noun phrase that precedes a verb can play the subject role in an active-voice sentence, but also the object role in a passive-voice sentence. Therefore, instead of identifying subjects and objects, the labels PRENP and POSTNP are used to refer to a noun phrase that directly precedes the verb and a noun phrase that directly follows the verb, respectively.

```
0  IF RTYPE1 THEN
1        IF RTYPE1 introduces the sentence THEN
2                RTYPE1's  PRENP = PRENP of its following clause
3        ELSE
4                RTYPE1's  PRENP candidate(s) is generated by:
5                Searching backwards and saving each POSTNP encountered
6                Stop when a top level verb or sentence start is reached
7  IF RTYPE2 THEN
8        IF RTYPE2 has PRENP but no POSTNP
9                RTYPE2's POSTNP candidate(s) is generated by:
1                        Searching backwards and saving each POSTNP encountered
1                        Stop when a top level verb or sentence start is reached
12       ELSE IF RTYPE2 has no PRENP
13               RTYPE2's PRENP candidate(s) is generated by:
14               Searching backwards and saving each POSTNP encountered
15               Stop when a top level verb or sentence start is reached
16 IF RTYPE3 THEN
17       RTYPE3's  POSTNP candidate(s) is generated by:
18       Searching backwards and saving each POSTNP encountered
19       Stop when a top level verb or sentence start is reached
```

**Fig. 1.** Algorithm for identifying noun phrase complements of embedded relative clauses.

Lines 0 through 6 correspond to RTYPE1 relative clauses. If the relative clause introduces the sentence, then the PRENP is in the clause that follows it in the sentence. Consider the following sentence: *Threatened by the imminence of an uprising, the King finally gave in to the demands. The King* is identified as the PRENP of the relative clause. The relative clause can now be read as its own sentence (in the passive voice): *<The king> <was> threatened by the imminence of an uprising*. If the relative clause does not introduce the sentence, then search backwards in the subsuming clause and compile a list of possible PRENPS. Stop when a verb phrase has been reached in the subsuming clause. Consider the following sentence*: This famous monument on the island, built between 1800 and 1700 B.C., may have been used for religious ceremonies*. The PRENPs for *built* are identified as *the island* and *this famous monument*. It cannot be determined which one is the correct PRENP. In this

case *this famous monument* is the correct PRENP, however, simply replacing *built* with *inhabited* would make *the island* the correct PRENP. Such decisions are not made by these algorithms since they require semantic and verb sub-categorization information.

Identifying RTYPE2 relative clauses (lines 7 through 15) requires two major rules (on lines 8 and 12). For the first case, consider the following sentence: *Peel organized the London police force in 1829 to aid in enforcing the criminal code, which he had revised*. The POSTNPs of *had revised* are identified as: *the criminal code, 1829,* or *the London police force*. The search stops at *the London police force* because the top level verb *organized* is reached. The list of possible complements are ranked in the order that they were discovered in the right to left search. For the second case, consider the following sentence: *Inscriptions and religious texts preserve the earliest Latin, which is sometimes called preliterary Latin*. The PRENP of *is sometimes called* is identified as *the earliest Latin*.

Finally, RTYPE3 relative clauses with no POSTNP are handled by the lines 16 through 19 in the algorithm. For example: The French and British governments instead accepted Hitler's assurance that Sudetenland was the final territorial acquisition he sought. The POSTNP list for sought includes the final territorial acquisition and Hitler's assurance. Sudetenland is not a possibility since it is a PRENP, and accepted is the first top level verb phrase encountered.

## 3 Infinitival Clauses

Figure 2 outlines the infinitival clause algorithm. Lines 0 through 2 recognize the PRENP of a purpose infinitival clause that introduces a sentence simply as the PRENP of the following clause. The following sentence is an example from the Encarta Encyclopedia of a purpose infinitive that introduces a sentence: *(INF To <Hermes> avoid leaving a trail that could be followed), Hermes made shoes from the bark of a tree and used grass to tie them to the cattle's hooves*. The algorithm correctly identified *Hermes* as the PRENP of *avoid*.

The remainder of the algorithm is for infinitival clauses that do not start a sentence and can be purpose or complement infinitives. Lines 3 through 7 are for infinitival clauses that do not have a POSTNP. The PRENP of these types of clauses depend on whether there is a POSTNP in the subsuming clause, for example: *The French forced al-Qadir ( INF to <al-Qadir> become a nomad after 1841 )*. and *Johnson resigned from the university ( INF to <Johnson> accept the position )*. Lines 8 through 14 are for infinitival clauses that have neither a PRENP nor POSTNP. The noun phrase complements that are identified are placed in both the PRENP and POSTNP lists. It cannot be exactly determined which lists the noun phrases belong in because this requires more detailed information about the particular infinitive verb. For example: *I want the book ( INF to <I> read <the book> ).* versus *I want the boy ( INF to <the boy> read <?> ).*

## 4 Results

The algorithms were tested on articles from the Wallstreet Journal Penn Treebank corpus[1], the Encyclopedia Encarta and the WorldBook Encyclopedia. Sentences were first part-of-speech tagged and partially parsed. The results are shown in Table 1.

```
0  IF INF introduces the sentence (purpose INF) THEN
1          INF's PRENP is PRENP of following clause
2          No assignment is made to object
3  ELSE IF INF has a POSTNP THEN
4          IF POSTNP is present in preceding clause THEN
5                  INF's PRENP is POSTNP or PRENP from preceding clause
6          ELSE
7                  INF's PRENP is PRENP from preceding clause
8  ELSE
9          IF POSTNP present in preceding clause THEN
10                 INF's PRENP is PRENP or POSTNP from previous clause
11                 INF's POSTNP is POSTNP from previous clause
12         ELSE
13                 INF's PRENP is PRENP from previous clause
14                 INF's POSTNP is empty
```

**Fig. 2.** Algorithm for identifying noun phrase complements of embedded relative clauses.

**Table 1.** Performance of the Relative and Infinitival Clause Algorithms.

| | Relative Clauses | | | Infinitival Clauses | | |
|---|---|---|---|---|---|---|
| **Source** | **Occur.** | **Precision** | **Recall** | **Occur.** | **Precision** | **Recall** |
| Penn Treebank | 187 | 84% | 91% | 89 | 89% | 90% |
| Encarta | 146 | 86% | 90% | 74 | 91% | 93% |
| WorldBook | 132 | 89% | 93% | 61 | 94% | 94% |

## References

1. Marcus, M., Santorini, B., and Marcinkiewicz, M. 1993. Building a Large Annotated corpus of English: the Penn Treebank. Computational Linguistics, 19:2. Pages 313-330.
2. Ngai, G. and Florian, R. 2001. Transformation-Based Learning in the Fast Lane. North American Chapter of the Association for Computation Linguistics.
3. van Delden, S., and Gomez, F. 2004. Cascaded Finite-State Partial Parsing: A Larger-First Approach. Recent Advances in Natural Language Processing III: Selected Papers from RANLP 2003. Current Issues in Linguistic Theory (CILT) Book Series. Vol. 260, Pp 402-410. John Benjamin Publishers, Amsterdam.

# An Add-On to Rule-Based Sifters for Multi-recipient Spam Emails

Vipul Sharma[1], Puneet Sarda[1], and Swasti Sharma[2]

[1] University of Houston, Department of Computer Science
4800 Calhoun Road, Houston, TX 77204, USA
{vsharma2,ppsarda}@uh.edu
http://www.cs.uh.edu
[2] College of Engineering Roorkee, Computer Science Department
Roorkee, India
sharmaswasti@gmail.com

**Abstract.** The Spam filtering technique described here targets multiple recipient Spam messages with similar email addresses. We exploit these similar patterns to create a rule-based classification system (accuracy 92%). Our technique uses the 'TO' and 'CC' fields to classify an email as Spam or Legitimate. We introduce certain new rules which should enhance the performance of the current filtering techniques [1][4][5]. We also introduce a novel metric to calculate the degree of similarity between a set of strings.

## 1 Introduction

We often find some uninvited emails in our inbox, which have multiple recipients and have addresses starting with one or more common alphabets. It is very unusual for a legitimate email to have the entire recipient list starting with one or more same alphabets like peter@123.edu, peterson@xyz.com, petric@wow.com etc. Sometimes entries in TO and CC fields are blank and unknown to the user and often there is just one unknown entry in TO and rest are in BCC of the email and hence are invisible. We measure these similar trends using our novel similarity metric (DOS) and use the score as a feature used for classification.

## 2 Degree of Similarity

We measure similarity depending on the similar characters at the same location of the strings. In order to measure this similarity we introduce DOS metric below.

### 2.1 DOS Metric

Let $m$ be the number of recipients in the address book of the current recipient. We move them to a separate address list *AddB*. The remaining addresses are arranged in a matrix $M$ where each row contains a single address as shown in fig. 3.1. $R$ & $C$ denote the total number of rows and columns respectively.

A. Montoyo et al. (Eds.): NLDB 2005, LNCS 3513, pp. 361–364, 2005.
© Springer-Verlag Berlin Heidelberg 2005

Let $n$ be the number of distinct letters in each column and $k_j$ be the count of each such letter in the column where $j = 1$ to $n$. The final score S of the matrix is calculated as below

$$S = \sum_c \sum_j (1/n)\, k_j \tag{1}$$

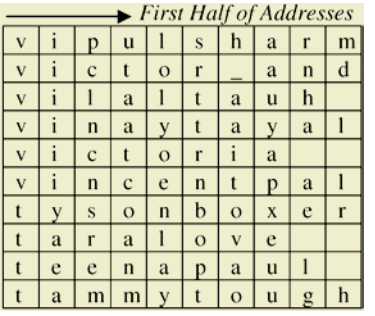



**Fig. 1.** Matrix arrangement of the first half of the email addresses.

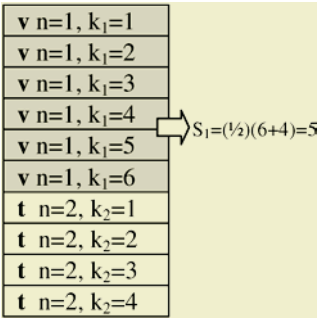



**Fig. 2.** High similarity in a column.

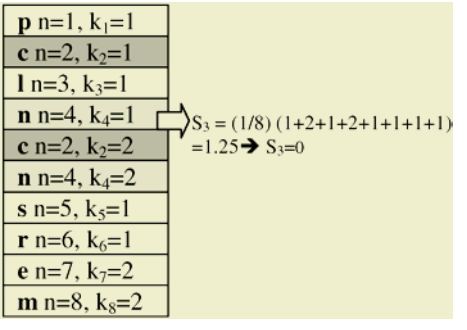



**Fig. 3.** Low Similarity in a column.

The minimum score for $S_j$, where $j$ is any column, can be 1 when all the entries in the column are distinct. For the case when the matrix contains very long and dissimilar addresses; the score for each column $S_j$, is less. However the total score of the matrix $S$ can be large for a large value of C, thus overestimating the score. To overcome this problem we modify $S_j$ by replacing $S_j$ by 0 if $(1<S_j<1.5)$.

### 2.2 Pseudocode of DOS

```
1) Extract the email id from each address
   1.1) If it belongs to the recipient's address book, move
        to AddB else store in the next row of M.

2) For each column in M
   2.1) for each row in the column
      2.1.1) If the letter is in H, increase it's count by 1
             else add to H with count as 1.
   2.2) calculate the score of the column as
        Sum of all counts in H/No. of distinct letters in H
   2.3) if score is less than 1.5 set to 0.
```

3) Compute the score of *M* as the sum of the scores of all columns.

4) The Final Score is
Score – ((No. of email ids in AddB)*Score/Total No. of email ids)

## 3 New Rules

A legitimate email will have less similarity among the addresses. We incorporate this notion in the following rules

**$R_0$ → S> T ➔ Spam** {T is the threshold decided by the learning algorithm}
**$R_1$ → S < T ➔ Legitimate**

The nature of Spam emails suggests that it is less likely for a Spam to have addresses from the current recipient's address book. We use the following rule to represent the influence of the above statement.

**$R_2$ → S = S - (m/R).S** where *m* and *R* are defined above

Another observation is that when the TO and CC fields are blank, the emails are generally spam. We use the following rule to incorporate this knowledge in our system

**$R_3$ → S = S + 50** (the number 50 was experimentally decided as the usual value of degree of similarity for Spam is close to 50)

We have concentrated only on a particular part of the message header and hence have defined rules for the same. These rules can easily be added with the rules of existing Spam filters [1] to enhance the performance of the overall system.

## 4 Results

We tested the real and simulated dataset separately using 10-fold cross validation using score *S* as a feature. The best performance was obtained by John Platt's sequential minimal optimization algorithm for a support vector classifier [2] in Weka: a machine-learning tool [3].

| Data Set | Simulated | Real |
|---|---|---|
| Number of Spam emails | 600 | 179 |
| Number of emails correctly classified as Spam | 600 | 168 |
| Number of emails misclassified as non Spam | 0 | 11 |
| Number of Legitimate emails | 400 | 220 |
| Number of emails correctly classified as non Spam | 400 | 198 |
| Number of false positive | 0 | 22 |
| Accuracy | 100% | 91.9% |

**Fig. 4.** Classification results on Simulated and Real emails.

## 5 Conclusions

Spam email is one of the most irritating problems faced by the entire web community. To enhance the ongoing research to eradicate this problem, we have introduced certain new rules to filter the multi recipient Spam emails. We also introduce a novel metric which can be used to quantify the similarity between multiple sets of strings. The accuracy obtained is satisfactory and should get better when the plug-in is fitted into the existing filtering components.

## References

1. Michael Parker, *Storing SpamAssassin User Data in SQL Databases*, ApacheCon 2004
2. Donghui Wu and Vladimir Vapnik. Support vector machine for text categorization, 1998
3. Witten I. H., Frank E.: Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Academic Press, London U.K (2000)
4. I. Androutsopoulos, G. Paliouras, and E. Michelakis. Learning to filter unsolicited comer cial e-mail. Technical Report, National Centre for Scientific Research Demokritos", 2004
5. G. Sakkis, I. Androutsopoulos, G. Paliouras, V. Karkaletsis, C.D. Spyropoulos, and P. Stamatopoulos. A memory bassed approach to anti-spam filtering for mailing lists. Information Retrieval, 6(1): 49-73, 2003
6. G. Navarro. A guided tour to approximate string matching. ACM Computing Surveys, 33(1):31–88, 2001

# Semantic Annotation of a Natural Language Corpus for Knowledge Extraction

Borja Navarro, Patricio Martínez-Barco, and Manuel Palomar

Grupo de Investigación en Procesamiento del Lenguaje y Sistemas de Información
Departamento de Lenguajes y Sistemas Informáticos
University of Alicante, Spain
{borja,patricio,mpalomar}@dlsi.ua.es

**Abstract.** Knowledge management (ontologies development, disambiguation of words, semantic web, etc.) must extract knowledge from somewhere. The main source of knowledge are natural language texts, in which humans express how they view and conceptualize the world. However, the automatic extraction of knowledge from texts is not a trivial task. In this paper we present a semantic annotated corpus as a source for knowledge extraction. Semantic is the bridge between linguistic input and knowledge (concepts, real world). A corpus with semantic information annotated is a useful resource to extract knowledge from a real context: it is a semi-structured database that offers deep information about human knowledge, concepts and relations between them[1].

## 1 Introduction

From our point of view, a formal model of world, an ontology, is formed by concepts that represent how human view and understand the world [1] [2]. The ontology represents the human knowledge about the world, not the world itself. The ontology, basically, consists of concepts, that is, the mental interiorization of the world and the knowledge about it. This kind of ontology is called "semantic ontology" by [2].

In the automatic construction of open domain ontologies, it is necessary to extract concepts and knowledge from somewhere. The sources in which humans express and conceptualize the world, and from which humans extract knowledge, are natural language texts. We think that they must be the main source for automatic knowledge extraction. Due to the semantic information is the bridge between the language input (words, syntactic relations, etc.) and the conceptual knowledge, it is necessary to deal with the semantic of natural language texts to achieve and extract knowledge. However, it is not a trivial task. Two main problems are, first, how to represent the semantic information and, second, how to deal with the semantic ambiguity.

In this paper we present a resource as a partial solution: a natural language corpus annotated manually with semantic information: the Cast3LB corpus [3].

[1] Research partially funded by Spanish Government FIT-150-500-2002-244, TIC-2003-07158-C04-01 and TIC-2003-7180, and by Valencia Government GV04B-276.

A. Montoyo et al. (Eds.): NLDB 2005, LNCS 3513, pp. 365–368, 2005.
© Springer-Verlag Berlin Heidelberg 2005

The use of corpus semantically annotated solves the main problems in automatic knowledge extraction, included the main one: in a corpus like that, all words are disambiguated.

## 2 Cast3LB Corpus: Semantic Annotation Overview

Cast3LB corpus is a semi-structured knowledge database in which each noun, verb and adjective has been annotated with its unambiguos sense: it is an "all words" corpus with the specific sense (or senses) of nouns, verbs and adjectives. The corpus has 42291 lexical words, where 20461 are nouns, 13471 are verbs and 8543 are adjectives.

For the formal representation of semantic data, a XML DTD has been developed in which, together with the syntactic information, a tag specifies the senses of the words. This specific sense of each word is made by means of the EuroWordNet offset number [4], that is, the identification number of the sense (synset) in the InterLingua Index of EuroWordNet.

Texts of Cast3LB corpus have been extracted from CLIC-TALP corpus (a subsection of LexEsp corpus [5] manually annotated at morphological level -100.000 words-) and from EFE Spanish Corpus -25.000 words-. The corpus contains a large variety of Spanish texts (newspapers, novels, scientific papers, etc.), both from Spain and South-America, so it is a good representation of the current state of the Spanish language[2].

### 2.1 Annotation Steps and Method

In order to overcome ambiguity problems, the annotation process has been carried out in two steps. In the first step, a subset of ambiguous words have been annotated twice by two annotators. With this double annotation we have developed a disagreement typology and an annotation handbook, where all the possible causes of ambiguity have been described and common solutions have been adopted for the rest of cases. In the second step the remaining corpus is annotated following the criteria adopted in the annotation handbook. On other hand, we have followed a semiautomatic annotation process: the human annotator must deal only with the polysemic words; the monosemic words were annotated automatically[3].

It is possible to distinguish two methods for semantically annotate a corpus. The first one is linear (or "textual") method (where the human annotator marks the sentences token by token up to the end of the corpus) and the second one is transversal (or "lexical") (where he/she annotates word-type by word-type, all the occurrences of each word in the corpus one by one)[6]. We have followed in Cast3LB the transversal process. The main advantage of this method is that

[2] Also, the corpus has been annotated previously with morphological and syntactic information. At the discourse level, the coreference of nominal phrases and some elliptical elements are in process of annotation.

[3] The monosemic words have been checked in order to detect wrong senses.

**Table 1.** Interannotation agreement. Experiment 2.

| Part of Speech | Total agreement | Average | Kappa |
|---|---|---|---|
| Adjectives | 74 | 81.08% | k = 0.70 |
| Nouns | 327 | 78.28% | k = 0.77 |
| Verbs | 147 | 70.06% | k = 0.64 |

we can focus our attention on the sense structure of one word and deal with its specific semantic problems: its main sense or senses, its specific senses, etc. Then we check the context of the single word each time it appears and select the corresponding sense. Through this approach, semantic features of each word is taken into consideration only once, and the whole corpus achieves greater consistency. Through the linear process, however, the annotator must remember the sense structure of each word and their specific problems each time the word appears in the corpus, making the annotation process much more complex, and increasing the possibilities of low consistency and disagreement between the annotators[4].

### 2.2 Evaluation

We have developed a experiment in order to measure the annotation agreement between the two annotator. We have used kappa measure [7] [8], that eliminate the agreement by chance. The result is better than the simple agreement average. Once the annotation handbook was written and annotators achieve experience in the annotation process, we compared two files annotated in parallel by each annotator. The results are shown in Table 1).

According to [7], a kappa measure between k = 0.6 and k = 0.8 is good agreement, and a kappa measure higher than k = 0.8 is total agreement. Our results are near this level. They show a good agreement between annotators: the corpus has a consistent and good semantic annotation.

## 3 Conclusions

An "all words" semantic annotated corpus like Cast3LB has several advantages for knowledge extraction: (i) Due to in Cast3LB the sense of each lexical word has been manually disambiguated, the annotation allows the extraction of all meaning possibilities of each word in its specific context. Also, due to the annotated senses are based on a hierarchy taxonomy like WordNet, it is possible to extract semantic generalizations and to achieve a more abstract level of characterization than the specific word sense. (ii) Cast3LB corpus has been annotated

[4] Nevertheless, the transversal method finds its disadvantage in the annotation of large corpus, because no fragment of the corpus is available until the whole corpus is completed. To avoid this, we have selected a fragment of the whole corpus and annotated it by means of the linear process.

with morphological and syntactic information, so it is possible to extract relations established between nouns and verbs in real texts. (iii) An ontology must be language independent. Cast3LB corpus is formed by Spanish texts, however, the same methodology has been used to annotate other corpora like catalan corpus Cat3LB and basque corpus Eus3LB. The semantic representation is the same for three corpora. (iiii) Finally, Cast3LB corpus is formed by real texts: they have been extracted from real communicative situations.

Due to the high amount of knowledge that must be extracted for the development of ontologies, the main problem in the use of annotated corpora is their small size. However, the corpus shows the real state of the language and knowledge for a specific domain. It is more objective the extraction of knowledge from different texts (and from different authors) than the formalization of knowledge by only one person.

To conclude, in this short paper we have argued that a deep management of knowledge needs the development of linguistic resources in which data are explicitly annotated. A corpus with semantic information is a useful resource to know how the knowledge is organized in a real context, and to extract it: the corpus offers deep information about how human language conceptualizes and manages knowledge. As a proposal, we have presented a semantic annotated corpus for Spanish (Cast3LB) and the annotation methodology.

## References

1. Bateman, J.A.: On the relationship between ontology construction and natural language: a socio-semiotic view. International Journal of Human-Computer Studies **43** (1995) 929 – 944
2. Nirenburg, S., Raskin, V.: Ontological semantics. MIT Press, Cambridge, Massachusetts (2004)
3. Navarro, B., Civit, M., Martí, M.A., Fernández, B., Marcos, R.: Syntactic, semantic and pragmatic annotation in Cast3LB. In: Proceedings of the Shallow Processing of Large Corpora. A Corpus Linguistics WorkShop, Lancaster, UK (2003)
4. Vossen, P.: EuroWordNet: Building a Multilingual Database with WordNets for European Languages. The ELRA Newsletter **3** (1998)
5. Sebastián, N., Martí, M.A., Carreiras, M.F., Cuetos, F.: 2000 LEXESP: Léxico Informatizado del Español. Edicions de la Universitat de Barcelona, Barcelona (2000)
6. Kilgarriff, A.: Gold standard datasets for evaluating word sense disambiguation programs. Computer Speech and Language. Special Use on Evaluation **12** (1998) 453–472
7. Carletta, J.: Assessing agreement on classification tasks: the kappa statistics. Computational Linguistics **22** (1996) 249–254
8. Cohen, J.: A coefficient of agreement for nominal scales. Educational Psychological Measurement **20** (1960)

# mySENSEVAL: Explaining WSD System Performance Using Target Word Features

Harri M.T. Saarikoski

Helsinki University, Department of General Linguistics
KIT Language Technology Doctorate School
PB 9, F-00014 Helsinki, Finland
Harri.Saarikoski@helsinki.fi

**Abstract.** Word sense disambiguation (WSD) is an unsolved problem in NLP. The field has produced a variety of methods but none of them potent enough to reach high, human-tagger accuracy in demanding NLP applications. Our contribution to WSD is mySENSEVAL, an error analyzer using SENSEVAL evaluation scores (in mySQL database) to find significant correlations between WSD system types and lexico-conceptual features (from WordNet and SUMO).

## 1 Introduction

Recent SENSEVAL-3 evaluation clinched the fact that WSD systems have reached a standstill in progress [3,6]. Level of accuracy obtained (72% to WordNet senses in SENSEVAL-3) is not reliable enough for demanding NLP applications, such as ontology learning or knowledge acquisition. The standstill reflects the difficulty of WSD system optimizing [2], which arises from not fully understanding the complex interrelations of the variables (words, senses, contexts) on systems.

There is an avenue in WSD research that has not been sufficiently explored: SENSEVAL system evaluation scores[1]. When thoroughly analyzed, we believe these scores can contribute to a more accurate system based on mapping types of words to types of systems. To that end, we have developed a tool to find out implicit answers to the open question of why some system types resolve some words (in some contexts) better than other systems. The findings obtained so far are described in [5]. This paper describes the user interface.

## 2 mySENSEVAL

This section presents our error analyzer.

### 2.1 User Interface

mySENSEVAL user interface is operated via HTML user interface (see Figure 1). It enables the viewing and/or adding and/or visualizing of:

[1] SENSEVAL evaluation sets (corpora and scores) can be found via http://www.senseval.org/. SENSEVAL-3 systems are described in [3,6] and SENSEVAL-2 systems in [1] and by developers themselves at http://www.cogs.susx.ac.uk/lab/nlp/mccarthy/SEVALsystems.html.

A. Montoyo et al. (Eds.): NLDB 2005, LNCS 3513, pp. 369–371, 2005.
© Springer-Verlag Berlin Heidelberg 2005

- Scores (and confusions) at all words, single words and all senses of one word
- Feature typologies of systems, words and senses
- Correlations between the features

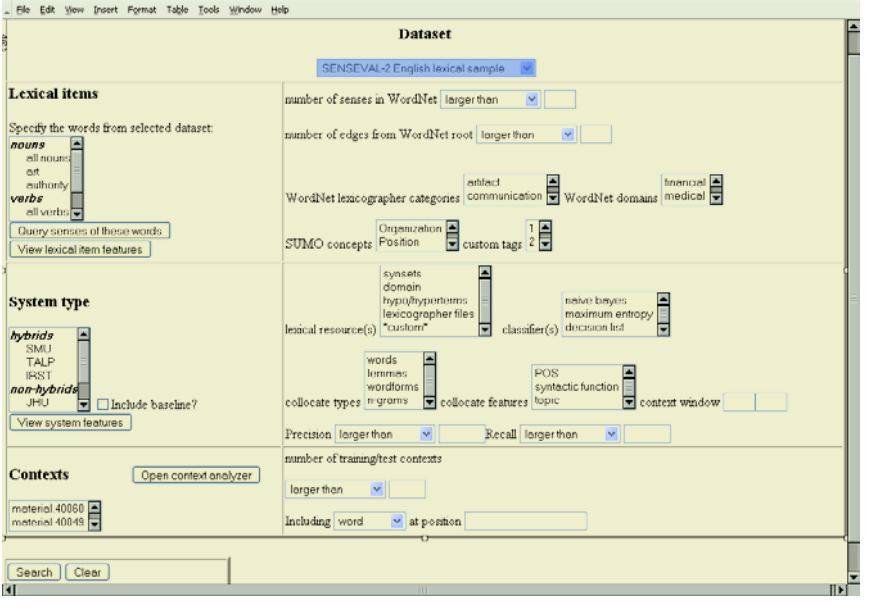



**Fig. 1.** Select Lexical items and Systems (on the left) and constraints (on the right).

## 2.2 Database Features

**Table 1.** Features of some senses, including SUMO concepts [4], WordNet lexicographer files and SENSEVAL-2 coarse-grain sense inventory [1] (not showing here).

| Sense ID | Definition | SUMO | Lexfile |
|---|---|---|---|
| Sense%1:10:00 | Meaning of e.g. word | Abstract | communication |
| Sense%1:09:02 | As in sense of e.g. timing | Capability | cognition |
| Art%1:09:00 | As in art of e.g. disguise | hasSkill | cognition |

**Table 2.** Semi-formalization of a few supervised systems, showing some key differences.

| System[a] | Context features | Algorithm heuristics |
|---|---|---|
| SMU | ▫ Features: nouns around target word<br>↓<br>▫ Window: 10 nouns | ▫ Classifiers and target words: Single classifier (instance-based learner) for all words.<br>▫ Knowledge: Uses monosemous context words (and their close relations in WordNet) for bootstrapping target words. |
| JHU | ▫ Features: words/lemmas, syntactic features and POS<br>↓<br>▫ Window: 'bags' (global) and n-grams (local) | ▫ Classifiers and target words: Voting pool of six classifiers, each with a different set of context features.<br>▫ Knowledge: Does not use any external knowledge. |

[a] SMU and JHU were the top contenders in S2ENLS [1].

## 3 Conclusion

This paper has presented mySENSEVAL, an error analyzer for defining the amount of correlation between words (in contexts) and systems. MySENSEVAL aims to help define the nature of guessing errors and facilitate the manual efforts entailed in WSD:

(1) by giving system developers a statistical scope into how different 'method families' (e.g. hybrid vs non-hybrid, single-classifier vs voting pool) perform with different target words

(2) giving lexico-conceptual resource designers a perspective into what lexical knowledge should be favored in order to cater for WSD systems

## References

1. Edmonds, P. and Kilgarriff, A. 2002. Introduction to the Special Issue on evaluating word sense disambiguation programs. Journal of Natural Language Engineering 8(4).
2. Hoste, V, Hendrickx, I., Daelemans, W. and van den Bosch, A. 2002. Parameter optimization for machine-learning of word sense disambiguation. Journal of Natural Language Engineering, 8(4).
3. Mihalcea, R., Kilgarriff, A. and Chklovski, T. 2004. The SENSEVAL-3 English lexical sample task. In Proceedings of SENSEVAL-3 Workshop at ACL-2004, Barcelona, Spain.
4. Niles, I., and Pease, A. 2001. Towards a standard upper ontology. In Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001), Chris Welty and Barry Smith, eds, Ogunquit, Maine, October 17-19, 2001.
5. Saarikoski, H. 2004. Using SENSEVAL system scores to optimize lexical resources for WSD and knowledge Acquisition. Proceedings of IBERAMIA, IX Ibero-American Conference for Artificial Intelligence, Puebla, Mexico.
6. SENSEVAL-3 evaluation workshop of WSD systems 2004. Proceedings of SENSEVAL-3 Workshop at ACL-2004, Barcelona, Spain, July 2004.

# Information Extraction from Email Announcements*

Viktor Pekar

Research Group for Computational Linguistics, University of Wolverhampton,
Stafford Street, WV1 1SB Wolverhampton, UK
V.Pekar@wlv.ac.uk
http://www.wlv.ac.uk/~in8113/

**Abstract.** Public email announcements present a number of unique challenges for an Information Extraction (IE) system, such as the presence of both free and semi-structured text, inconsistent document layout and widely varying formats of template fillers. In this paper we describe a study of parametrisation of an IE method to determine settings that best suit the specifics of the task at hand.

## 1 Introduction

Previous successful IE methods have been concerned with extracting information from consistently structured web pages and from free text. Reliable performance can be achieved by systems that operate on web pages with regular layout, such as those providing a web interface to a database, e.g., [1]. These techniques rely on the analysis of the HTML structure of pages to learn how to locate relevant strings of text in them. Considerable progress has also been made in developing methods for extracting information from free text, e.g., [3]. The cues these systems use are the output of different NLP techniques such as tokenization, syntactic parsing, and named entity recognition. The present paper is concerned with IE from public email announcements, an important medium for distributing information on the web. Email announcements share many characteristics with both types of documents just described, but also have their own peculiarities that create specific challenges for an IE system. The major ones are, firstly, the fact that since announcements come from diverse sources, there is a considerable variation in the manner of laying out an announcement. As a result, these documents are hardly amenable to rigid extraction patterns firing on domain-specific structural constraints. Secondly, fillers typically have different formats such as named entities, terms, dates, free word phrases, and even sentences. Thirdly, the fillers may be found both in free and semi-structured text.

In this paper we present a study of applying an IE method which uses a machine learning procedure to extract information from this type of documents. We describe an experimental study of parametrisation of the IE method to determine parameter settings that best suit the specifics of this task. The particular kind of announcements the study is concerned with are conference announcements in the area of NLP and AI.

* The study was supported by a ESRC grant RES-000-23-0010.

A. Montoyo et al. (Eds.): NLDB 2005, LNCS 3513, pp. 372–375, 2005.
© Springer-Verlag Berlin Heidelberg 2005

## 2 Information Extraction Method

Previous research has developed a range of IE methods for tasks that require filling one template per document, e.g., [1], [3]. Here, we opt for an IE method similar to the one proposed by [2]. This method learns two distinct classifiers from an annotated corpus: one operating on the level of sentences and one on the level of words. First, the sentence-level classifier scans the document for sentences that potentially contain template fillers. This is achieved with high accuracy since the classifier uses features derived from broad contexts, namely tokens from the entire sentence. After that, the word-level classifier attempts to extract fillers in the relevant sentences by looking at the local context of each of its words.

Given the specifics of our task, namely unclear boundaries of fillers and their varying formats, instead of trying to learn a word-level classifier, we use only the first-level classifier and experiment with a variety of ways to delimit document fragments. The solution we are after is an optimal balance between the size of the document fragments and the effectiveness of classifications.

We study three types of document fragments:

- *Sections*, i.e. each fragment is made up of text between two section headings;
- *Sentences*, i.e. each fragment is delimited by the boundaries of paragraphs, tables, itemisation lists; full sentences inside paragraphs are further split into sentences;
- *Lines*, i.e. each fragment corresponds to text between two line break tags.

When fragmenting a document, special care was taken not to break documents across multiword NEs. If, for example, a line break tag appeared inside text marked up as a NE, the NE was added to the previous fragment, so that the fragment ended directly after the closing NE tag.

## 3 Evaluation

To represent each document fragment, the following steps were performed: all words found in it were stemmed, NEs substituted for their semantic label, and stopwords removed. Each text token appearing inside the fragment formed a feature, its numerical value being its frequency of occurrence. A separate feature was introduced that represented the relative position of the fragment inside the document.

The class label of the fragment was obtained by looking at which IE filler was present inside it. Since a fragment may contain more than one filler, the data contained instances with multiple class labels. To perform multilabel classification, a binary classifier was trained for each class and each test instance was classified by $n$ classifiers, where $n$ is the number of unique class labels in the training data. In our experiments we used the WEKA implementation of the multinomial Naïve Bayes learner [4]. Annotated data for training classifiers was prepared by downloading 100 calls for papers from the Elsnet list on-line archive. The evaluation method used was 10-fold cross-validation.

## 4 Results

Table 1 characterises the fragments resulting from each of the methods: *Sections*, *Sentences* and *Lines*. Because the *Lines* method produces the smallest fragments, it is the most preferable one, but it may also be the hardest, since instances prepared from these fragments contain the least number of features. The *Sections* method, on the contrary, is least preferable, but seems to present the easiest classification problem. The accuracy attained by the methods is shown in Table 3.

**Table 1.** The size of fragments and the number of fragments per document resulting from each of the three fragmentation methods.

| | Sections | Sentences | Lines |
|---|---|---|---|
| Words per fragment | 68.59 | 18.85 | 7.32 |
| Fragments per document | 15 | 54.2 | 142.4 |

**Table 2.** The precision (*P*), recall (*R*), and F-measure (*F*) rates for the *Section*, *Sentences*, and *Lines* methods (the best scores across the methods appear in bold).

| | Sections | | | Sentences | | | Lines | | |
|---|---|---|---|---|---|---|---|---|---|
| | P | R | F | P | R | F | P | R | F |
| AREA | .589 | .832 | .69 | .567 | .83 | .674 | **.861** | **.914** | **.887** |
| CITY | .384 | .718 | .5 | .417 | **.726** | **.53** | **.491** | .552 | .52 |
| EMAIL | .357 | **.883** | .508 | .396 | **.876** | .546 | **.529** | .565 | **.547** |
| COUNTRY | **.35** | **.683** | **.463** | .302 | .667 | .416 | .32 | .352 | .335 |
| DATES | .408 | .743 | .527 | **.425** | **.806** | **.557** | .381 | .444 | .41 |
| DEADLINE | .385 | **.884** | .536 | .405 | **.865** | .552 | **.456** | .806 | **.583** |
| FINALCOPY | .385 | .925 | .544 | .494 | **.95** | **.65** | **.563** | .768 | .649 |
| NAME | .469 | .744 | .576 | .501 | .792 | .614 | **.606** | **.801** | **.69** |
| NOTIFICATION | .357 | .932 | .516 | .483 | **.978** | **.647** | **.53** | .787 | .633 |
| ORGANISER | .34 | .795 | **.476** | .329 | **.832** | .472 | **.348** | .301 | .323 |
| PCMEMBER | .447 | .854 | .587 | .51 | .943 | .662 | **.856** | **.96** | **.905** |
| URL | .348 | .813 | .486 | .416 | **.884** | .566 | **.514** | .687 | **.588** |
| All | .401 | .817 | .534 | .437 | .845 | .573 | **.537** | .661 | **.589** |

The results show that the *Lines* method actually proves the most effective one in terms of the overall F-measure, achieving slightly better performance than the *Sentences* method. For individual extraction fields, *Lines* outperforms others by a wide margin for AREA and PCMEMBER and does a little better or is on pare for most of other fields.

Looking at the precision scores, *Lines* is clearly the best: its advantage over *Sentences* is 10 and over "Sections" 13 points in terms of overall precision. In terms of overall recall, however, the *Sentences* method has a significant advantage over *Lines* (18 points) and is a bit more effective than *Sections* (3 points). *Sentences* also happens to be the best performer in terms of recall for individual fields.

These results indicate that if one is to use the fragment classification method on its own for extracting information from a document, the *Lines* method is the most advis-

able choice. It not only delineates the smallest fragments with potentially useful text, but is also the most effective method overall. However if, as in [2], a further procedure is to be applied to the relevant fragments to more precisely identify filler instances, then the *Sentences* method should be preferred, since it achieves the highest recall.

Since we are using a separate classifier for each extraction field, it is possible to create a combination of classifiers to achieve the highest possible results in terms the desired evaluation measure. Thus, using the *Sentences* method for all fields except AREA, NAME, and PCMEMBER (for which *Lines* is used), we can achieve the greatest recall overall of .85.

## 5 Conclusion

In this paper we investigated ways to suit an IE method to the specifics of extracting information from email announcements. In order to identify the fillers in a document, we attempt to locate the smallest possible text fragments that are likely to contain the filler. We find that fragments of a document, relevant to IE, are best identified by dividing it into small fragments such as lines of text.

The obtained results can be taken to be optimal only for some of the template fields, e.g. AREA (F=.88) and PCMEMBER (F=.9), but not others (e.g., COUNTRY and ORGANISER, F=.47). Despite that, the approach achieves quite high recall rates, overall (.85) and for individual fields. In our future work we are going to focus on techniques to pinpoint the exact filler more precisely in the text fragments identified as relevant by the approach described here.

## References

1. Kushmerick, N., Weld, D., Doorenbos, R.: Wrapper Induction for Information Extraction. In: Proc. of IJCAI-97 (1997) 729–737.
2. De Sitter, A., Daelemans, W.: Information Extraction via Double Classification. In: Proc. of the ECML/PKDD 2003 Workshop on Adaptive Text Extraction and Mining. Cavtat-Dubrovnik, Croatia (2003)
3. Soderland, S.: Learning Information Extraction Rules for Semi-structured and Free Text. Machine Learning, Vol. 34. (1999) 233–272.
4. Witten I., Frank, E.: Data Mining – Practical Machine Learning Tools and Techniques with Java Implementations. Morgan-Kaufmann. San Francisco, CA (2000)

# An Application of NLP Rules to Spoken Document Segmentation Task

Rafael M. Terol, Patricio Martínez-Barco,
Fernando Llopis, and Trinitario Martínez

Departamento de Lenguajes y Sistemas Informáticos
Universidad de Alicante
Carretera de San Vicente del Raspeig - Alicante - Spain
Tel.: +34965903772, Fax: +34965909326
{rafamt,patricio,llopis,tme}@dlsi.ua.es

**Abstract.** One of the main differences between Spoken Document Retrieval (SDR) systems and Text Retrieval systems is the need of a segmentation process that detects the story boundaries. However, until now, SDR researchers have not paid attention in building correct segments more than considering slidding windows of a fixed size in time. In this paper, new methodology for evaluating segments to SDR task, and the evaluation of three possible strategies are presented over the TREC-9 SDR collection. Moreover, the justification of each strategy is discussed.

## 1 Introduction

One of the main problems of SDR systems is to obtain a text representation of spoken documents. Our SDR system [2] employs a simplified spoken document segmenter based on NLP techniques with the aim to accomplish this goal. We designed and developed two advanced spoken document segmenters with introduces the application of NLP techniques more refined than the simplified spoken document segmenter: the first one incorporates the application on windowing in the simplified spoken document segmenter, and the second one is a more complex spoken document segmenter which design and main features will be showed in this paper.

Next section presents the background of the segmentation task that well-know spoken document retrieval systems apply in their global process. Following we describe the motivation that permit us the development of our segmenter toolkit based on Natural Language Processing rules and the main features about this segmenter toolkit. Finally the evaluation task and results of our segmenter toolkit are also presented.

## 2 Background

To solve this segmentation problem in Spoken Document Retrieval, different approaches have been applied such as:

A. Montoyo et al. (Eds.): NLDB 2005, LNCS 3513, pp. 376–379, 2005.
© Springer-Verlag Berlin Heidelberg 2005

- CL-SDR System by Univ. of Chicago [3] produced 30 seconds segments based on the word recognition time stamps using 10 second step to create overlapping segment windows.
- CL-SDR System by ITC-irst [3] produced segments with a shifting time-window of 30 seconds, moved with steps of 10 seconds. In this case, segments were also truncated if a silence period longer than 5 seconds was found.
- Cambridge University SDR system at TREC-9 [4] was based on windowing of 30 seconds, but in this case, with an inter-window shift of 15 seconds. Moreover, they used the commercial tags to filter out words thought to have originated in commercial breaks. By this way, the transcriptions were first filtered, removing all words which occurred within periods labelled as commercial in the non-lexical file.
- THISL SDR System [5] is a probabilistic text retrieval system based on overlapping rectangular windows of the audio stream with defined both frame length and frame shift.
- LIMSI SDR System [6] segmented the audio stream into overlapping documents of a fixed duration. After some tests, they chose a 30s window duration with a 15s overlap. As there were many stories significantly shorter than 30s in broadcast shows, they concluded that it may be of interest to use a double windowing system in order to better target short stories.

## 3 Spoken Document Segmenter

This section shows, as baseline, the simplified spoken document segmenter based on temporal pauses observed inside speaker speech. Likewise, this baseline release of the spoken document segmenter is used as base of the window spoken document segmenter and the complex spoken document segmenter. The features of these two segmenters are also presented in this section.

### 3.1 Simplified Spoken Document Segmenter

The aim of this simplified spoken document segmenter is to obtain structured text in sentences from original spoken document. The decision about which words belong to one sentence and which others belong to another sentence is only based on the length of temporal pauses between the pronunciation of the words. Therefore, this simplified release of the spoken document segmenter acquires a parameter used as temporal upper limit inside speech speaker to determine if a word belong to one sentence or, alternatively, if the same word is the initial word of another sentence.

### 3.2 Window Spoken Document Segmenter

This spoken document segmenter is based on sliding windowing of the sentences obtained from the application of the simplified spoken document segmenter to spoken documents. Different tests applied to this spoken document segmenter

based on spoken document retrieval demonstrated that 10 is the best size of the window and 3 is the best size of slide.

The design consideration based on temporal pauses in the speech of the speaker of these two spoken document segmenters can produce, as instance, syntactic mistakes. An example of one of these syntactic mistakes is that the last word of a sentence would be an auxiliar verb:

*Sentence 1:* ***... police now are.***
*Sentence 2:* ***trying to determine which ...***

These kind of mistakes and another ones originated that we also focused our research effort in the design and development processes of a more complex release of this spoken document segmenter. Following subsection shows all details about it.

### 3.3 Complex Spoken Document Segmenter

This segmenter is able to extract the different news for each one of the spoken documents and to assign each sentence to one of the extracted news.

With the aim to produce a more natural segmentation, we have designed a set of rules to apply in the segmentation process of spoken documents. These rules consider, apart of temporal pauses between words and grammar rules, other aspects such as: the word position in the text, the number of the word occurrences in the text, the frequency of occurrences of the word in the text, TFIDF, and so on.

## 4 Evaluation and Conclusions

In the evaluation task we have considered as reference collection the document collection for the CL-SDR track at CLEF 2003 that is composed only by 21.754 documents with known boundaries. Nevertheless, we have handled as test collection the document collection for the same track at CLEF 2004 that was the same as previous one but in this last one the boundaries between documents are unknown and it also includes noise such as commercials, filler, etc. In order to perform an evaluation to know the goodness of the segmentation, it is need to count on both document collections: the reference collection (RC) and the test collection (TC). This fact produces that the evaluation task consists in to verify for each one of segments obtained from the test collection the goodness of the assignation with a segment of the test collection.

Then it is considered that a segment $t_i$ in test collection is matched with a segment $r_j$ in reference collection that maximizes the function $\mathbf{Card}(t_i \cap r_j)$ when it is accomplished:

- $r_{j1} \leq t_{im} < r_{(j+1)1}$ where:
  - $r_{j1}$ is the start time of the segment $r_j$
  - $t_{im}$ is the middle time of the segment $t_i$
  - $r_{(j+1)1}$ is the start time of the contiguous segment to $r_j$

If all these conditions are accomplished then the $t_i$ segment has been rightly segmented. Otherwise this $t_i$ segment has been wrongly segmented. Table 1 shows results obtained for two spoken document segmenters.

**Table 1.** Results of segmentation evaluation.

| | Segments | Documents | Noise | Matched | Precision |
|---|---|---|---|---|---|
| Baseline | 291613 | 231611 | 60002 | 211599 | 0.913 |
| Window | 95398 | 77966 | 17432 | 63850 | 0.819 |
| Complex | 28444 | 24687 | 3757 | 15457 | 0.626 |

We can conclude that the incorporation of more complex NLP techniques produces worse segmentation results than the application of simplified NLP techniques.

## Acknowledgment

This research work has been partially funded by the Spanish Government under project CICyT number TIC2000-0664-C02-02 and PROFIT number FIT-340100- 2004-14 and by the Valencia Government under project numbers GV04B-276 and GV04B-268.

## References

1. The Ninth Text Retrieval Conference (TREC 9), Gaithersburg, Maryland (2000)
2. Llopis, F., Martínez-Barco, P.: Spoken Document Retrieval experiments with IR-n system. In: Proceedings of the CLEF 2003: Workshop on Cross-Language Information Retrieval and Evaluation, Trondheim (2003) 427–433
3. Federico, M., Bertoldi, N., Levow, G.A., Jones, G.J.: (CLEF 2004 Cross-Language Spoken Document Retrieval Track) 631–634
4. Johnson, S., Jourlin, P., Jones, K.S., Woodland, P.: Spoken Document Retrieval for TREC-9 at Cambridge University. [1] 117–126
5. Renals, S., Abberley, D.: The Thisl SDR System at TREC-9. [1] 627–634
6. Gauvain, J.L., Lamel, L., Barras, C., Adda, G., de Kercardio, Y.: The LIMSI SDR System for TREC-9. [1] 335–360

# A Generalised Similarity Measure for Question Answering

Gerhard Fliedner[1,2]

[1] DFKI GmbH, D-66123 Saarbrücken
[2] Computational Linguistics, Saarland University, D-66123 Saarbrücken
fliedner@coli.uni-sb.de

**Abstract.** We define the Generalised Similarity Measure (GSM) as a means of uniformly and efficiently storing linguistic information to search for answers in Question Answering (QA) systems. It computes the similarity between a question representation and those of possible answers in a document collection as a database query. Linguistic knowledge from different sources can be used and combined in the GSM, allowing to find matches even with imperfect representations. To show the viability of the concept, we have implemented the GSM in a proof-of-concept QA system for German, employing information from WordNet and FrameNet. First experiments have been promising, large-scale tests are underway[1].

## 1 Introduction

Question answering systems today mostly do the actual search in their document collection using bag-of-words techniques based the surface words of the questions, known from Information Retrieval (IR), followed by a deeper analysis of documents returned by the search [1, 2]. Even though this solution has proved to be efficient and robust, the recall may suffer in cases where bag-of-words techniques are not able to spot a possible answer. This problem can only partly be solved by techniques like query expansion using synonyms.

Therefore, pre-processing the whole document collection linguistically and doing the actual search on more abstract, preferably semantic, representations of the questions and the document collection would seem attractive. However, it comes with a number of drawbacks.

First, linguistically pre-processing a document collection is time consuming, possibly requiring years of CPU time. Dropping hardware costs, however, have made the parallel processing on dozens of machines a viable solution.

The second big problem comes up at retrieval time: The actual matching between the representations of questions and potential answers calls for a 'vague' matching, allowing for differences in wording, such as use of synonyms (e. g. *buy* vs. *take over*) or nominalisations (*buy X* vs. *takeover of X*), coreference resolution and resolution of implications.

[1] This is ongoing work in the Collate project, funded by the German Ministry for Education and Research, Grant numbers 01 IN A01 B and 01 IN C02.

A. Montoyo et al. (Eds.): NLDB 2005, LNCS 3513, pp. 380–383, 2005.
© Springer-Verlag Berlin Heidelberg 2005

Another drawback of the direct match approach is that, in practice, not all input sentences will receive a representation (due to lack of lexical or grammatical coverage, among other things). Therefore, a search mechanism relying solely on one linguistic representation will often fail. It would be desirable to make use of different levels of linguistic information in parallel.

## 2 A Generalised Similarity Measure

We define the GSM for the search process in a QA system as a method of matching the relational representation of the questions and the underlying document collection. This relational representation assumes that both the question and the fact it asks for are represented as a set of relations between words. In general, each function word introduces one distinct object into the universe of discourse.

Thus, the domain of the relations is exactly this set of linguistic objects. We only consider one place relations (i. e. properties of objects) and two place relations. Relations with greater arity are expressed by a set of two place relations 'anchored' to the same object. Thus, $'love('john,'mary)$ would become $'john(o_1),'mary(o_2),'love(o_3),'subj(o_3,o_1),'obj(o_3,o_2)$.

The GSM is then defined as the similarity between these relations (i. e. the concepts, not the instances). We use a floating point number, ranging between 1 (perfect match, identity) and 0 (no match). The GSMs are pre-computed and stored in a database. Thus, at retrieval time, the GSM table can directly be used for searching by doing a database join over the database search term generated by the query, the GSM table and the actual data in the database.

To allow a generalisation over the relations, we allow different types of relations between the objects, corresponding to different knowledge sources. In our experiments, for example, we have used both GermaNet and FrameNet.

After translating the document collection into our relation format and storing it, searching for the answer of a question is done by first parsing and translating the question into its relational representation and then into a suitable database query. The actual search is thus the task of identifying in the database a set of linguistic objects and the relations that hold between them, where the relations are as similar as possible to those between the words in the question. The similarity of a possible answer to the question is computed as the product of the GSMs of all relations of the question's representation and the corresponding answer representations.

For example, the question *'Whom does John adore?'* would be translated into $'john(q_1),'adore(q_3),'subj(q_3,q_1),'obj(q_3,q_2)$, with $q_2$ being marked as the linguistic object answering the question. In combination with a given similarity between $'love$ and $'adore$ of, say, 0.9, this would match the objects $o_1,\ldots,o_3$ from above, giving *Mary* ($o_2$) as the answer.

The GSM also allows to define a similarity for converse relations. When matching, e. g., $'give$ and $'get$, the relations between the verbs and their arguments can be properly related to each other via the correct GSM similarities.

Using the GSM, it is possible to make use of redundancy by listing as many linguistic facts as possible for the instances in the corpus, as the GSM can

combine them. Thus, even if one level is not available both in the question and the answer representation, a match is still possible based on the remaining ones.

It is important to stress that the GSM in itself does not provide the linguistic knowledge on relational similarity. This must be taken from external sources. In our experiments, we have used (close) hypernym relations from the hand-crafted GermaNet and FrameNet ontologies successfully.

The notion of matching the similarity between both properties and relations is similar to [3]. However, our approach differs in that in encodes different levels of information (here, FrameNet and GermaNet) to allow redundant searches.

## 3 Implementation and Experiments

We have developed the GSM as a technique for efficiently matching linguistically similar answers to questions as part of a QA system for German that is still under development. Our system uses a cascade of comparatively flat parsers to produce linguistic representations of German input text with different layers, not all of which may be found for every sentence, namely a dependency structure enhanced with GermaNet information and a FrameNet layer. The whole setup is described in greater detail in [4].

GermaNet, the German version of WordNet and part of the EuroWordNet [5], is used to provide semantic sortal information, especially for nouns. FrameNet is a lexical database resource containing valency information and 'abstract' predicates. English FrameNet is developed at the ICSI, Berkeley, CA [6]. Development of a German FrameNet is currently underway here at Saarland University [7].

FrameNet is especially suited for the computation of the GSM as described above: It provides properties (namely frames) for words, it defines the suitable relations (namely frame elements) and it also defines relations between frames and their frame elements by so-called frame relations. As a basis for computing the GSM between frames and frame elements, we use their path distance in the FrameNet relation graph. One of the reasons for using FrameNet is that it groups related words together, abstracting away, e. g., over verbs vs. nouns.

GermaNet is used to provide properties and mappings between them. For words that can be found in GermaNet, the GermaNet synset is used as a property. GSMs between GermaNet properties are defined using their path distance in the GermaNet hypernym tree.

GermaNet synsets are combined with the grammatical functions returned by our dependency parser. These are, among other things, normalised over active/passive diathesis, so that the DSub (for Deep Subject) always means the underlying subject.

The correct matching of named entities (NEs) is especially important for QA. NEs are, in general, made up out of more than one word, such as *George W. Bush* or *Lockheed, Inc.* In most cases, different forms of the NE can be used. We assume GSM=1 for perfect match and GSMs<1 for partial overlap of words.

During the linguistic analysis of the documents, coreference is heuristically resolved (anaphora and definite NPs). To be able to use this information at search

time, for all relations between a word and a coreferent expression, this relation is also added between the word and the antecedent and marked as an antecedent link. This allows efficient retrieval, but also reconstructing the original structure.

As a first experiment with the GSM, we have analysed several newspaper articles with our parser and translated the resulting representations into a database for querying. We used articles from the business section of the *Süddeutsche*, a German daily paper, from 1995, comprising approximately 1,700 sentences. The GSM measures were instantiated using close hypernym relations from both the FrameNet and the GermaNet ontologies, as described above.

We formulated factoid questions that came to mind when reading the articles in questions, posed these questions to the system and tested whether the fact that answered the question was among the answers returned by our system. In a first analysis of questions and answers, we found that using the GSM to search for answers helped. Especially important were the possibility to find answers with different parts of speech and close synonyms. Also, coreference resolution played an important role.

## 4 Conclusions

We have presented the Generalised Similarity Measure as a means of uniformly and efficiently storing linguistic knowledge for the use in a direct search in QA.

Future work will include testing the method with larger amounts of data. We think that the GSM method should scale up well. The first experiments have shown that in several respects the data used for the GSM may still be improved: We think that using corpus based similarity measures may improve the overall quality, as these measures would be closer to the actual language use than measures derived from hand-crafted language resources such as GermaNet.

## References

1. Moldovan, D., Harabagiu, S., Girju, R., Morarescu, P., Lacatusu, F., Novischi, A., Badulescu, A., Bolohan, O.: LCC tools for question answering. In: TREC 2002. (2002)
2. Hovy, E., Hermjakob, U., Lin, C.Y.: The use of external knowledge in factoid QA. In: TREC 2001. (2001)
3. Montes-y-Gómez, M., Gelbukh, A., López-López, A., Baeza-Yates, R.: Flexible comparison of conceptual graphs. In: DEXA 2001. Number 2113 in LNCS (2001)
4. Fliedner, G.: Deriving FrameNet representations: Towards meaning-oriented question answering. In: NLDB 2004. Number 3136 in LNCS (2004)
5. Kunze, C., Lemnitzer, L.: Germanet – representation, visualization, application. In: LREC 2002. (2002)
6. Baker, C.F., Fillmore, C.J., Lowe, J.B.: The Berkeley FrameNet project. In: COLING 98. (1998)
7. Erk, K., Kowalski, A., Pinkal, M.: A corpus resource for lexical semantics. In: IWCS 2003. (2003)

# Multi-lingual Database Querying and the Atoms of Language

Epaminondas Kapetanios and Panagiotis Chountas

Health Care Computing Group
School of Computer Science (HSCS)
Univ. of Westminster, London, UK
e.kapetanios@wmin.ac.uk

**Abstract.** The paper presents MDDQL as a query language suitable for multi-lingual conceptual querying of collections of databases from a graphical user interface or from an application programming one. The query language, however, has been specified and implemented with the parametric theory of linguistic diversity in mind such that syntactic and semantic parsing of multi-lingual query expressions becomes quite simple and guarantees identical query results regardless the preferred natural language. We present a parsing algorithm, which shows that it is quite easy to formulate a query regardless the underlying type order of a natural language, be it *Subject-Object-Verb*, *Subject-Verb-Object*, or *Object-Verb-Subject*, etc., and still being able to grasp the meaning of the query at a minimal computational effort possible.

## 1 Introduction

Engineering multi-lingual, natural language interfaces to database systems is faced with inherited complexity [1–3] as imposed by the various grammatical details and issues underlying each natural language. According to the parametric theory of linguistic diversity, however, *parameters* could reconcile our conflicting senses of the sameness and difference of languages [4].

In order to achieve this goal, the MDDQL query language approach [5] comes with a parsing mechanism which leads to conceptually equivalent high-level query trees, regardless the type order, e.g., the order of the basic elements of *subject*, *object*, *verb*, to which families of natural languages normally obeys.

In contrast with current approaches towards the elaboration of huge linguistic dictionaries and ontologies [6], one for each natural language in order to cope with multi-lingual query interfaces, there is an inherited restriction of the maintenance efforts and the overall complexity of the system.

This is due to the fact that semantic parsing takes place in terms of an ontology driven meachanism. This enables the parsing mechanism to block meaningless queries such as *All cars aged more than 40 years, which have been infected by AIDS*, which grammatically [7] is correct.

Given that the ontology also reflects the semantic role of verbs, subjects, objects and other basic elements, the sequence order of query terms can be easily adapted to the type order another particular language might obey, e.g., change from *subject-verb-object*

A. Montoyo et al. (Eds.): NLDB 2005, LNCS 3513, pp. 384–387, 2005.
© Springer-Verlag Berlin Heidelberg 2005

to *object-subject-verb*, and, therefore, accept queries from another family of natural languages, even in their restricted form as sub-languages.

Given also that each query is reflected conceptually on the same MDDQL high level query tree regardless the order of query terms, the semantic parsing result of a query remains unaffected. In addition, adapting of the parsing technique to the semantics of another natural language as far as the NL type order is concerned becomes a matter of adapting the navigational algorihm within the multi-layered conceptual graph in which the ontology is being represented.

*Organization of the paper:* Section 2 roughly introduces the parametric theory of linguistic diversity. Section 3 refers to the parsing services of MDDQL queries and how they relate to the construction of the MDDQL conceptual query tree. A conclusion summarizes the major issues of the MDDQL approach.

## 2 The Parametric Theory of Linguistic Diversity

According to the parametric theory of linguistic diversity [4], it is not the *words*, which are considered as the *atoms of a language*, but rather *parameters* which lead to classification of natural languages according to some *word order type*.

A rough classification of natural languages according to some basic word order types and their distribution is given in the following:

- Subject–[Verb–Object], 42 percent, for example, English, German, Indonesian
- Subject–[Object–Verb], 45 percent, for example, Japanese, Turkish
- Verb–[Subject–Object], 9 percent, for example, Zapotec, Welsh
- etc.

These parameters suffice to bring natural languages closer to each other, though they appear in their presentation to be totally different. For instance, French, Spanish and Welsh belong to the same category as far as the parameter *Subject Placement* within a sentence is concerned. To this extent, English and Welsh are similar in terms of the parameter *Verb Attraction*, a parameter which determines whether tense auxiliaries attract the verb to their position or verbs attract tense auxiliaries to their position.

## 3 The MDDQL Parsing Approach for Multi-lingual Queries

Prior to proceeding with the MDDQL parsing approach, it is worth having a look at the form the conceptual, high level MDDQL query tree takes, regardless the natural language in which a query expression has been formulated. The conceptual MDDQL query tree can be defined by the following constraints:

- The root of the query tree is always a *Class* or an *Instance* term node.
- A *Class* or *Instance* term node might have as children other *Class*, *Instance* or *Property* term nodes.
- A *Datatype Property* term node might have as children other *Datatype Property* term nodes or *Value* term nodes.

- An *Object Property* term node, i.e., relationship between two *agents*, MUST have children, which are *Classes* or *Instance* term nodes.
- An *Object Property* term node, i.e., relationship between two *agents*, might have as children *Property* term nodes.
- A *Value* term node might have as children only *Value* nodes.

For example, consider the intended query *patients having received immediate therapy* as reflected on a conceptual, high level query tree. The query term *have received* is represented by a query tree node and classified on the query tree as an *Object Property* according to its role within the ontology.

A query such as *patients having received immediate therapy* can be passed as an argument of type *String* to the method *QueryParsing* of an object as an instance of the class MDDQL-Queries, e.g., *CurrentQuery.QueryParsing("patients having received immediate therapy")*. The same query, however, if expressed in *Turkish* or *Japanese*, it would have taken the form *CurrentQuery.QueryParsing ('patients immediate therapy having received')*[1]. Similarly, the same query in Welsh could have been expressed like *CurrentQuery.QueryParsing ('having received patients immediate therapy')*, according to the *Verb-Subject-Object* word order type for the family of natural languages to which Welsh belongs.

The semantic parsing technique, however, would have generated out of all three multi-lingual queries an equivalent conceptual MDDQL query tree. In order to do so, an ontology driven parsing technique has been implemented, which also relies on the assignment of the semantic roles of words as representing concepts, relationships, properties, etc., rather than only their semantic roles as part of the underlying grammar, e.g., subject, object, verb, etc.

In other words and given the expression of the roles of the terms at different semantic layers wihin the ontology representation (linguistics, conceptualization, etc.), the lexical and morphological process relies on a different level of connectionism among the terms, which is taken onto consideration, when the type order of the used natural language to express a query is changed.

For instance, assuming that the query expression *CurrentQuery.QueryParsing ('having received patients immediate therapy')* in Welsh is being lexically analyzed, the first matching set of words within the ontology is being searched, i.e., *having received*. The ontology graph traversal algorithm, however, further identifies and returns all those concepts, which are semantically related to the concept *having received*, if one takes into consideration only those ontological roles as assigned to the terms within an ontology.

To this extent, *patients* and *immediate therapy* are accepted as meaningful candidates to connect the term *having received* with and are further expected to be met by the lexical analyzer. However, the word order to be accepted will be *patients immediate therapy* to complete the query, since the pattern *subject-object* is now being expected.

Generally speaking, query parsing or completion relies on both semantic layers of terms: the one expressing the linguistic relationship, and that one expressing the con-

[1] For the sake of simplicity and for all examples, we do not make use of the symbols of a particular language, such as Turkish or Japanese symbols, since these languages are typical representatives of the word order *Subject-Object-Verb*.

ceptual relationships. The construction of and reflection on the corresponding query tree, however, relies only on the conceptual relationships among terms.

For instance, changing natural language for query expressions to one of the type order *Object-Subject-Verb*, the query tree would have taken the same form, i.e., *immediate therapy* → *having received* → *patients*, since both *immediate therapy* and *patients* are concepts, whereas *having received* is a semantic relationship, and the query tree structure complies with the constraints underlying the definition of the query tree.

In general terms, what is actually put on the conceptual query tree, reflects the ontological relationship rather than the order of appearance of words within the sentence. This enables the parsing and generation of database specific queries such as SQL statements at the lowest effort possible [5], however, by taking into consideration a restricted vocabulary, since all we need to adapt are the symbols of a particular language within the ontology while preserving the semantic relationships.

## 4 Conclusion

MDDQL relies on the implementation of a parsing mechanism according to the parametric theory of natural language diversity, which emphasizes the commonalities of natural languages, while minimizing their differences. To this extent, parameters are considered to be the atoms of any language rather than having words fulfilling this role. The system has been implemented in Java and currently provides a platform for querying collections of databases in multi-lingual, natural sub-language. A further important aspect of this approach to be considered is *simplicity* and *scalability* in terms of adding an new natural sub-language, even from a different family of natural languages.

## References

1. Thalheim, B., Kobienia, T.: Generating DB Queries for Web NL Requests Using Schema Information and DB Content. In Moreno, A.M., van de Riet, R.P., eds.: NLDB 2001. Volume 3 of LNI., Madrid, Spain, GI (2001) 205–209
2. Metais, E., Mayr, H.C.: NLDB'99: Applications of natural language to information systems. Journal of Data and Knowledge Engineering **35** (2000) 107–109
3. Ambriola, V., Gervasi, V.: Experiences with Domain-Based Parsing of Natural Language Requirements. In: 4th International Conference on Applications of Natural Language to Information Systems, Klagenfurt, Austria, IOS Press (1999)
4. Baker, M.C.: The Atoms of Language. Oxford University Press (2002)
5. Kapetanios, E., Baer, D., Groenewoud, P.: Simplifying Syntactic and Semantic Parsing of NL Based Queries in Advanced Application Domains. In: 8th Intern. Conf. on Applications of Natural Language to Information Systems, NLDB 2003. Lecture Notes in Informatics, LNI, Cottbus, Germany, Springer Verlag (2003)
6. Düsterhöft, A., Thalheim, B., eds.: Natural Language Processing and Information Systems, 8th International Conference on Applications of Natural Language to Information Systems. LNI, Burg (Spreewald), Germany, GI (2003)
7. Knuth, D.E.: Semantics of Context-Free Languages. In: Mathematical Systems Theory. Volume 2. (1968) 127–145

# Extracting Information from Short Messages

Richard Cooper, Sajjad Ali, and Chenlan Bi

Computing Science, University of Glasgow, 17 Lilybank Gardens, Glasgow G12 8QQ
rich@dcs.gla.ac.uk

**Abstract.** Much currently transmitted information takes the form of e-mails or SMS text messages and so extracting information from such short messages is increasingly important. The words in a message can be partitioned into the syntactic structure, terms from the domain of discourse and the data being transmitted. This paper describes a light-weight Information Extraction component which uses pattern matching to separate the three aspects: the structure is supplied as a template; domain terms are the metadata of a data source (or their synonyms), and data is extracted as those words matching placeholders in the templates.

## 1 Introduction

In developing a body of information, we typically ask others for information, interpret what they tell us, extract the information we want and store it away. The work described here attempts to build a semi-automated system in which the information is passed by e-mail or SMS text message and is to be stored in a database. In this case, we can exploit two features of the message – it will include terms from the database domain and it is probable that the language structure will be fairly simple. We can also ignore anything in the message which seems to be irrelevant. We cannot, on the other hand, be as confident that syntax and spelling will be accurately used – indeed in the case of text messaging, spelling rules will almost always be transformed dramatically. In fact, a recent experiment soliciting messages of this sort elicited messages which ignored natural language in favour of making up a form structure.

In a factual statement, the function of the words in the sentence naturally falls into three categories. Articles, conjunctives, etc. are there to provide *syntactic structure*. Other words identify information categories from the *domain of discourse*. The remaining words provide *data values* drawn from the information categories. In database terms, the second group are metadata and the third group data. Making this three way distinction explicit permits us to attempt text analysis in a number of ways. Much IE work attempts to learn the structures having been given the terms from the domain of discourse [1]. Other work starts by manually tagging the text [2]. In our work, we can mostly assume we know both terms and structure and are only trying to find the data values. Attempts to discover the structure use two broad approaches: fully parsing the text [3, 4] or matching fragments of the text with structural templates. The second approach seems more promising in contexts such as this one, in which the domain is restricted but the language will be used loosely, and is the approach we will describe here. We will provide sentence pattern templates with placeholders for the domain terms and the data. This is lightweight in the sense used in the work of Kang *et al.* [5], and is also similar to the work of Stratica and Desai [6], both using similar techniques to process natural language queries.

A. Montoyo et al. (Eds.): NLDB 2005, LNCS 3513, pp. 388–391, 2005.
© Springer-Verlag Berlin Heidelberg 2005

Turning to the domain terms, we believe that the IE process needs to be given these as well. To extract information for storage we use the metadata as a basis for our collection of terms, augmented with synonyms to cope with equivalent terms. The terms are combined with the templates to generate patterns for matching. When a match is found, the data is extracted from the parts of the text matching data placeholders and the result is turned into appropriate database updates. The matching process uses a maintained context to deal with anaphoric references and the update generation creates a mixture of entity creation and property update commands.

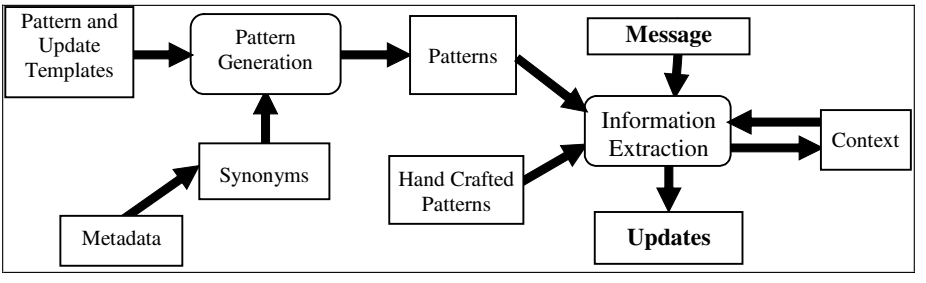



**Fig. 1.** An Architecture for Information Extraction to a Data Source.

## 2 A Pattern Matching Information Extraction System

The system we have created comprises three phases – the generation of a collection of sentence structures patterns; the use of that collection to locate new data; and the use of that data to derive database update statements to store the information found. The architecture supporting these activities is shown as Figure 2.

### 2.1 Setting up the System

The setup process includes patterns generation and consists of the following steps:

1. Create a schema for the data using a special purpose data model which more closely resembles the structures underlying the communication. The model is a standard entity model in which entities have properties which may be base values or other entities, but adds a base type for gender so that every entity can be identified as masculine, feminine or neuter, this being used to interpret pronouns. There are also two notions of keys – ones that a database would use (Dkeys) and ones that a human would use (Hkeys).
2. Generate and edit the synonyms for the metadata using WordNet. Two tasks are required here – noun synonyms which do not change the sentence structure and verb synonyms which create a fresh sentence structure.
3. Input pattern templates.
4. Generate the patterns by combining the templates, the metadata and synonyms.

The pattern templates are text strings distinguishing the three classes of word: structural words, domain terms and data, where the data may be either fresh data that the visitor is communicating or an HKey value which the visitor sending the message expects to find in the database already. The structural words are introduced *verbatim*, the other two classes of word appear as placeholders. The pattern generation mecha-

nism takes the template, leaves the structural words unchanged, replaces the domain term placeholders with meta-data and makes specific the data placeholders.

Here is a simple example of a template which includes a metadata placeholder, which will be replaced by each of the property names and their noun synonyms, and two data placeholders which will be replaced by each of the human key property placeholders and all of the property placeholders respectively.

– "The <PropertyName> of <<HkeyValue>> is <<PropertyValue>>"

The pattern generator takes each entity type in turn and produces patterns for every combination of properties that fit the template, one of which would be for a movie entity type which uses *title* as an Hkey and has another property *year*::

– "The year of <titleValue> is <yearValue >"

from which the information extraction process can recover *yearValue*=1958 from either of: "The year of this film is 1958." or "The year of The Music Room is 1958."

After the IE process, the component will now have values for particular property, in this case the year. It can discover which entity the property is for, either by context in the first instance or by using the Hkey in the second instance.

### 2.2 The Information Extraction Process and the Use of Context

The IE process is passed a message and a starting context (see below), tokenises the message and identifies sentences. Each sentence is then checked against the patterns, ordered so that the most specific structure will be found first. If the sentence does not match anything, it is ignored. Otherwise, data is extracted and update statement(s) are output. After each sentence, the context is updated and the next sentence is checked. Much of this is routine string manipulation, but the complicating factor is the context.

The discussion above assumes that each sentence is complete in itself, but this is rarely the case. Most sentences will have contextual references embedded in them either in the form of pronouns, definites or implicit references. In these cases, our component must discover which entity is referred to before the sentence can be processed. To this end, the component manages an object maintaining contextual information which contains: variables holding references to the most recently mentioned entity type and the most recently mentioned entity of: any type, each gender and each type. When the IE component is passed a message, it will be in response to a request or a question about a specific entity or entity type, in which case it can initialise the context using this. It could also start cold with an empty context.

The process of extracting data is now more complex than just pattern matching and proceeds by identifying explicitly referred to entities first, then uses the gender variable for pronouns, the entity type variables for definites and the most recently used entity for implicit references. When the sentence has been dealt with, it is necessary to update the context in order to prepare the component for the next sentence. Any entity encountered will be used to update the various context variables.

### 2.3 Generating the Updates

The extraction process returns values of one or more properties for one or more entities, identified either by context variable reference or key value, perhaps only a human key. Having located new data values in the message, we proceed as follows.

To update a single property, we will have the property name, the entity key and the value – enough to produce a simple update command. If the property has an entity type, then the Dkey may have to be found from the Hkey to be the updated value.

There are two occasions when we need to add a new entity to the repository – when the sentence explicitly discusses a new entity that the message is informing us about and when the message is informing us about an entity property value that may or may not be in the repository. In either case, we will only have an Hkey and if this is not also a Dkey, we must generate a Dkey. If this fails to find a value, a new entity must be created using an insert command, generating a new Dkey to identify it.

## 3 Conclusions

The system (more fully described in [7,8]) as described handles simple sentences including the use of noun synonyms and context. The design supports the automatic generation of different verb phrases, but that has yet to be fully implemented. However, these can be added by hand if required. There are however, many ways in which the work needs to progress, including the handling of more complex sentence structures, fuzzy word checking, learning of sentence structures [9], extending the context mechanism to hold more of the history, handling negative or conflicting information and the management of synonyms at the data level.

## References

1. R. Gaizauskas and Y. Wilks, *Information Extraction: Beyond Document Retrieva*l, Journal of Documentation, 54(1):70--105, 1998
2. D. Fisher, S.Soderland, J. McCarthy, F. Feng and W. Lehnert, Umass System, MUC-6, 1995
3. C. Cardie, *Empirical Methods in Information Extraction*, AI Magazine, 18:4, 65--79 1997
4. http://gate.ac.uk/
5. I-S Kang, S-H Na, J-H Lee and G. Yang, *Lightweight Natural Language Database Interfaces*, NLDB 2004, LNCS 3136, pp76-88, 2004
6. N. Stratica and B. C. Desai, *Schema-Based Natural Language Semantic Mapping*, NLDB 2004, LNCS 3136, pp103-113, 2004
7. Cooper,R.L. and Ali,S., *Extracting Database Information from E-mail Messages*, 20th British National Conference on Databases, July 2003, pp 271-279, LNCS 2712, Springer
8. Cooper,R.L., Ali,S.and Bi, C.L., *A System for Extracting Information from Short Messages*, Technical Report, University of Glasgow, in press.
9. E. Agichtein and L. Gravano, *Snowball: Extracting Relations from Large Plain-Text Collections*, Proc.5th ACM International Conference on Digital Libraries (DL), 2000

# Automatic Transition of Natural Language Software Requirements Specification into Formal Presentation

M.G. Ilieva and Olga Ormandjieva

Department of Computer Science and Software Engineering
Concordia University
Montreal, Quebec, Canada
ormandj@cse.concordia.ca

**Abstract.** Software requirements specification is a critical activity of the software process, as errors at this stage inevitably lead to problems later on in system design and implementation. The requirements are written in natural language, with the potential for ambiguity, contradiction or misunderstanding, or simply an inability of developers to deal with a large amount of information. This paper proposes a methodology for the natural language processing of textual descriptions of the requirements of an unlimited natural language and their automatic mapping to the object-oriented analysis model.

## 1 Introduction

The modeling of Software Engineering ideas, or, more precisely, the models and the modeling languages used in Object-Oriented (OO) software systems development, exerts a notable influence on the development and application of the natural language processing (NLP). On the one hand, there is OO software modeling which involves the identification of the things, or *concepts*, that are important in the environment where the system will function, i.e. the system's *domain*, and their abstraction as a *domain model* where the relations between the real-world things are abstracted as relationships between conceptual classes. On the other hand, there is the natural language (NL), which is the verbal form of the same mental modeling process of the human, throughout which those real-world things are arranged and connected. Analogies between OO modeling and NL can easily be noted. Naturally, the question arises, isn't the translation of NL into an OO model easy? In practice it is not, because the automated extraction of semantics from NL is difficult. To help with this process, we have the modeling language, which is positioned between the NL description and the OO programming language. It contains, and reflects, the creative process of formalization and modeling. This creative process is the bridge between NL and human thinking on the one hand, and between formal language (FL) and formal thinking on the other. The use of the modeling language no doubt shortens the distance between FL and NL, and is characterized by the following features: i) it breaks the problems down into smaller parts which are differentiated in their functions; ii) it uses formal representations such as schemas, diagrams and drawings, which are close to the thinking of the human, but more precise; iii) it includes ele-

A. Montoyo et al. (Eds.): NLDB 2005, LNCS 3513, pp. 392–397, 2005.
© Springer-Verlag Berlin Heidelberg 2005

ments which have a direct analogy in NL – the actors, for example, constituting the subject of the sentence; iv) it includes NL description even when highly simplified and very formalized. Never before the appearance of unified modeling language (UML) [2] has the distance between NL specifications and FL been so close. The shortened distance becomes a stimulus in the search for new solutions in NLP, which could be used in any position on the way to translation of the software specification into a programming product.

The paper is organized as follows: The main research directions in this area are outlined in section 2. Our approach is presented in section 3, and illustrated in section 4 on a case study of the industrial importance. Finally, the conclusions and future work directions are outlined in section 5.

## 2 Related Work

Different approaches to the solution of similar problems are presented in the literature. Some methodologies [4][7] are primarily concerned with a more simplified NL syntax, since in those cases the extraction of semantics was easier to achieve. A shortcoming of such systems is their limited applicability. Another approach is to process informal NL [1][5][6][8]. The shortcoming of this approach is the considerable effort required for translation of the NL description into a conceptual scheme, which is also a form that the problem-solver must have in mind.

We create a methodology for automated translation of NL specifications into an OO analysis model. Our research concerns the type of knowledge required, and how and to what degree it can be extracted, in order to build an OO analysis model. The novelty of our research consists in proposing a methodology for automatic formalization of software requirements written in unrestricted NL, which avoids the shortcomings described above. The method and its capabilities are introduced in the following section.

## 3 Basic Process Stages

The main objective of the software requirements specification phase is to understand the textual descriptions (requirements) and abstract the software to be built into an OO Analysis Model. Our approach imitates the human analysis process in that it divides the problem into parts to make it more easily understood; then, it collects the parts into a whole (which could be presented in different ways) from which it derives the solution to the problem.

Our method consists of three main processing parts: i) the **Linguistic Component**, in which the sentences in the text are analyzed; ii) the **Semantic Network**, built by the formal NL presentation; and iii) **OO modeling**, the final phase of the formal presentation of the specification, through which the knowledge and information included in the semantic network are transmitted to the OO analysis model's elements.

### 3.1 Linguistic Component

The job of the linguistic analyst is to find the parts of the speech and organize them in groups. A group is defined as a word or related words, which together perform one function. There are three functions (roles) in a sentence: Subject, Predicate and Object. We have created three groups corresponding to these roles – the subject group, the predicate group and the object group. We use the term 'group' rather than the term 'phrase' commonly found in the computer linguistics terminology because we give more functional meaning to the groups. The groups are combined at three levels, as shown in Figure 1. The sentences are rewritten in tabular form, along with important semantic and syntactical information extracted from the requirements during their analysis (see section 4 for an illustration of this). We consider this tabular presentation to be a form of knowledge base, the size and content of which depend on the problem being solved. The original description remains unchanged.

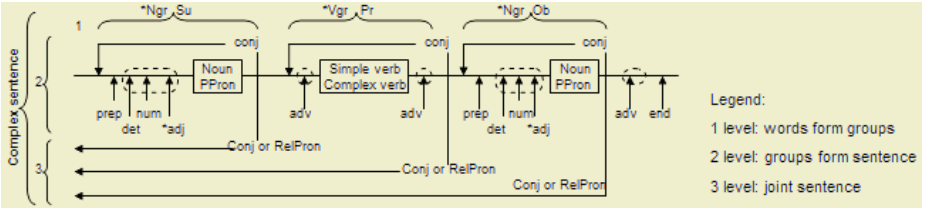



**Fig. 1.** Model of the sentence/text.

### 3.2 Semantic Network

We can use the knowledge extracted from the sentence and presented in tabular form (see section 3.1) to build a semantic network. The network gives us a complete image of the connections between the entities. During the construction of the network graphical presentation, the entities are abstracted as nodes, and the connections with which the entities participate in the entire text are presented as transitions. This construction is harder to build in the memory of the human (analyst) only by reading the text. The graphical elements that we use are shown in Figure 3. The set of graphic elements in the semantic network can be added to, depending on the specifics of a given problem domain and the text being processed.

### 3.3 OO Model

The knowledge from the semantic network can easily be translated to the OO model. What is interesting in this model are the internal knots in the net, which apply to classes, and the connections within them, which apply to their *properties* and *procedures.* In our work, we adapt the various heuristics proposed in [2] [3] [12] for finding the candidate classes and their relationships.

An example of the automatic translation of software specifications into an OO model using the proposed methodology is presented in the following section.

## 4 Case Study on Real-Time Reactive Systems: Robotics

The methodology is intended for unlimited text specification and text-interview (see example in [11]). However, because of the space limitations of this article, we will solve the following short example [9].

> "An assembly unit consists of a user, a belt, a vision system, a robot with two arms, and a tray for assembly. The user puts two kinds of parts, dish and cup, onto the belt. The belt conveys the parts towards the vision system. Whenever a part enters the sensor zone, the vision system senses it and informs the belt to stop immediately. The vision system then recognizes the type of part and informs the robot, so that the robot can pick it up from the belt. The robot picks up the part, and the bell moves again. An assembly is complete when a dish and cup are placed on the tray separately by the arms of the robot."

The text specification in tabular form is introduced in Figure 2. Note how convenient this tabular presentation is, in that it breaks down the description into clear parts, which can be found quickly and manipulated.

| № | Type of sentence | Subject | Predicate | Sequence number | Object | Connection between sentences |
|---|---|---|---|---|---|---|
| 1 | main | An assembly unit | consists of | | a user, a belt, a vision system, a robot with two arms, and a tray for assembly | . |
| 2 | main | The user | puts | 1 | two kinds of parts, dish and cup, onto the belt | . |
| 3 | main | The belt | conveys | 2 | the parts towards the vision system | . |
| 4 | cond (Whenever) | a part | enters | 3 | the sensor zone | , |
| | conj | the vision system | senses | 4 | it | and |
| | conj | | informs | 5<br>6 | the belt<br>to stop immediately | . |
| 5 | main | The vision system | then recognizes | 7 | the type of part | and |
| | conj | | informs | 8 | the robot | , so that |
| | conj | the robot | can pick up | 9 | it from the belt | . |
| 6 | main | The robot | picks up | 10 | the part | , and |
| | conj | the belt | moves again | 11 | | . |
| 7 | main | An assembly | is complete | 12 | | when |
| | cond (when) | a dish and cup | are placed | 13 | on the tray separately by the arms of the robot | . |

**Fig. 2.** Tabular presentation of the text specification.

We build the semantic network according to clearly defined rules using the graphic elements described in Figure 3.

We consider the semantic network carefully. Within it, there are two relations that express structures: an "assembly unit" and a "part". They participate in the relations "consists of" and "is a kind of". We chose these for classes, together with their subclasses. We assign the procedures to classes according to clearly defined rules, and we obtain the class diagram shown in Figure 4.

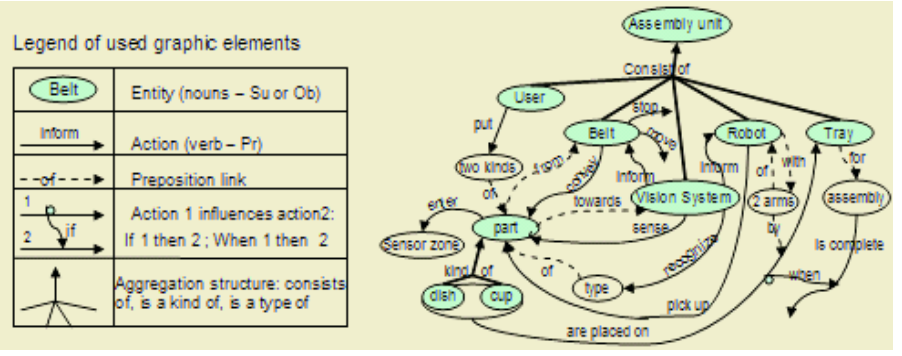



**Fig. 3.** Semantic network of text specification with the use of graphical elements.

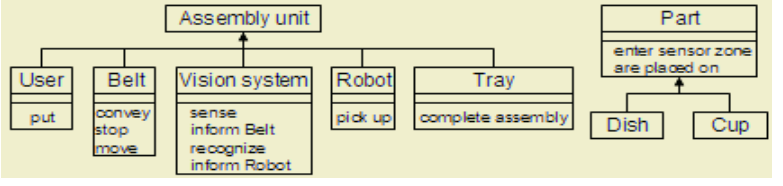



**Fig. 4.** Class Model.

These classes are obtained through automated processing. With clarity, completeness and precision, the schemes give the analyzer enough information for verification and correction, if needed.

## 5 Evaluation of the Results and Future Work

The proposed methodology is tested through various examples taken from publications detailing similar systems [3][4][7][8], which automatically translate an NL requirement into a formal model, most often an OO-class diagram. References [9] and [10] use another type of formalism for OO modeling, but their final results are no different from ours. Moreover, there is no indication of imprecision or incompleteness, which would result from the analysis of the NL description of the specification. The appropriately chosen formal presentation of NL, together with the extra extracted semantic information, offers possibilities for other models, too; for example, the use-case diagram, time-sequence diagrams of real time, activity diagram and state diagram. These models will be included at a later stage.

## References

1. Bryant, B.R.: Object-Oriented Natural Language Requirements Specification. In 23rd Australian Computer Science Conference, 2000 (ACSC 2000).
2. Booch G., Rumbaugh J., Jacobson I.: The Unified Modeling Language User Guide, Addison-Wesley, 19983. van Leeuwen, J. (ed.): Computer Science Today.

3. John Lewis J., Loftus W.: Java Software Solutions – Foundations of Program Design. Published by Addison-Wesley ISBN: 0-321-24583-0. 4th Edition 2004.
4. Kalaivani Subramaniam, Dong Liu, Behrouz H. Far, Armin Eberlein, UCDA: Use Case Driven Development Assistant Tool for Class Model Generation. In Proceeding of Sixteenth International Conference on Software Engineering and Knowledge Engineering (SEKE'04), Banff, Canada, June 2004.
5. Kop, C.; Mayr, H.C.: Mapping Functional Requirements: From Natural Language to Conceptual Schemata. In Proc. 6th IASTED Int. Conference on Software Engineering and Applications (SEA 2002), November 2002, Cambridge, USA.
6. Lee, B.-S., Bryant, B.R.: Automated conversion from requirements documentation to an object-oriented formal specification language. In Proceedings of SAC 2002, March 10-14, 2002, Madrid, Spain, ACM 2002.
7. Mencl, V.: Deriving Behavior Specifications from Textual Use Cases. In Proceedings of Workshop on Intelligent Technologies for Software Engineering (WITSE04, Sep 21, 2004, part of ASE 2004), Lenz, Austria, ISBN 3-85403-180-7, pp. 331-341, Oesterreichische Computer Gesellschaft.
8. Moreno, A. M.: Object-oriented analysis from textual specifications. In Proceedings of Ninth International Conference on Software Engineering and Knowledge Engineering, Madrid, Spain. (1997).
9. V.S. Alagar, O. Ormandjieva, Shi Hui Liu, Jian Shen: Performance Assessment in Real-Time Reactive Systems. In Proceedings of the Seventh IASTED International Conference on Software Engineering and Applications (SEA 2003), November 3-5, 2003, USA, pp.714-722.
10. V. S. Alagar, M. Chen, O. Ormandjieva, M. Zheng: Automated Test Generation from Object-Oriented Specifications of Real-Time Reactive Systems. Tenth Asia-Pacific Software Engineering Conference, December 10-12, 2003, Chiang Mai, Thailand
11. http://odl-skopje.etf.ukim.edu.mk/UML-Help/html/05day2.html – "Discovering Business Processes 1" – The Digital Library – Example.
12. http://odl-skopje.etf.ukim.edu.mk/UML-Help/html/05day4.html – "Domain Analysis" – The object modeler looks for nouns, verbs and verb phrase.

# Automatic Description of Static Images in Natural Language

Azucena Montes Rendón[1], Pablo Sánchez Luna[1], Gerardo Reyes Salgado[1], Juan G. González Serna[1], and Ricardo Fuentes Covarrubias[2]

[1] Centro Nacional de Investigación y Desarrollo Tecnológico
Interior Internado Palmira s/n, col. Palmira.
Cuernavaca, Morelos, México C.P. 62490
{amr,terion,greyes,gabriel}@cenidet.edu.mx
http://www.cenidet.edu.mx
[2] Universidad de Colima
fuentes@ucol.mx

**Abstract.** In this paper, the description of an image in natural language is carried out. The main idea is that from an image, with objects without movement, it is possible to obtain phrases in Spanish describing the position among the objects. In order to put this description into effect, we place ourselves in a theoretical model in which a cognitive-semantic analysis of linguistic units such as the prepositions *sobre* (on), *en* (in), *entre* (between) and the verb *tocar* (to touch) is realized. This analysis will allow to establish rules which will determine the relationship or position among the objects.

## 1 Introduction

In the present paper, the description of an image starts from two processes: the Geometric Description and Extraction of the Characteristics of Objects (GDECO) and the Cognitive-Semantic Analysis (CSA). In the first process the information related among objects, their coordinates, their areas and central points are obtained. In the second process, we place ourselves in a model called Cognitive and Applicative Grammar. In this model, the CSA of some linguistic units such as prepositions and verbs is carried out aiming to extract pertinent information to use with the GDECO in order to establish the rules which will determine the relationship in words among the objects.

## 2 Cognitive-Semantic Analyses

In order to develop the linguistic analysis we place ourselves in a formal model called Cognitive and Applicative Grammar (CAG) [1]. This model manipulates three levels of explicit representations of the language. In the level of cognitive-semantic representations the CSA will be developed for the followings linguistics units. In the case of the prepositions, the analysis will lead us to find a meaning invariant. In the case of the verb, we will analyse the meaning of static use [2].

A. Montoyo et al. (Eds.): NLDB 2005, LNCS 3513, pp. 398–401, 2005.
© Springer-Verlag Berlin Heidelberg 2005

**Preposition *sobre* (on):** According to the analysis accomplished in [3], the preposition *sobre* has the following invariant: location regarding the borderline of a place according to gradient and contact between the two entities. Example of a spatial use: *el libro está sobre la mesa* (the book is on the table). The preposition *sobre* implies considering the borderline of the reference place *mesa*. The object *libro* is located on the surface of this place. This location is related to a vertical axis.

**Preposition *en* (in):** According to our CSA of the spatial uses of the preposition *en*, the following invariant was obtained: location regarding in the closure of a place, and contact between the two entities. Example of spatial use: *el libro está en el escritorio* (the book is on the desk). In this example, the entity *libro* is contained in the area determined by the entity *escritorio*.

**Preposition *entre* (between):** According to our CSA of the spatial uses of the preposition *entre*, the following invariant was obtained: location regarding the outside borderline FRO-ext [4] of two entities having or not having contact, according to a gradient at a horizontal sense. Example of spatial use: *las cajas están entre el escritorio y el librero* (the boxes are between the desk and the bookcase). The entity *cajas* is located at the outside borderline of the entities *escritorio* and *librero*.

**Verb *tocar* (to touch):** We only consider the use of this verb for static situations. In order to carry out the CSA of this verb, let us consider the following expression: *el librero toca las cajas* (the bookcase touches the boxes). The CSS [1] for this expression is: SIT1 [∧(REP FRO (*x,* (loc(*y*))), (REP FRO ( *y,* (loc(*x*))))]; *x* and *y* are variables that indicate the entities *librero* and *cajas* respectively; *x* is located at the borderline of *y*, which is indicated by *x* REP FRO (loc(*y*)), and on the other hand, *y* is located at the borderline of *x*, which is indicated by *y* REP FRO (loc(*x*)). The analysis shows that it is indispensable a contact between both entities.

## 3 Geometric Description and Extraction of the Characteristics of Objects

For the GDECO we have the follow convention:

| Coordinates | Description | Convention |
|---|---|---|
| Point A | Left top corner | pA |
| Point B | Right lower corner | pB |
| Point C | Central point of the object | pC |
| A abcisa | Point A abcisa | pA.x |
| A ordinate | Point A ordinate | pA.y |
| Object *i* abcisa | Point A abcisa of object *i* | $pA.x_i$ |
| Object *i* area | Object *i* area | $A_i$ |

**Rules**

***Sobre* (On):**

Rule 1 (location at the borderline). $pA.x_k <= pA.x_i \wedge pB.x_k >= pB.x_i$

Rule 2 (contact). $pA.y_k <= pB.y_i <= pB.x_k \wedge pA.y_i < pA.y_k$

Rule 3 (difference in size). $A_i <= A_k$

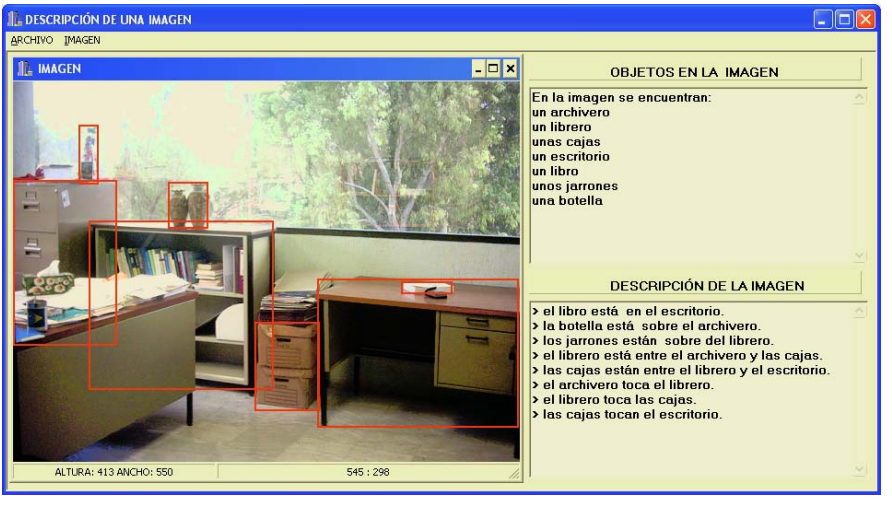
***En* (In):**
Rule 1 (location in the closure and contact). $pA.x_k <= pA.x_i \wedge pB.x_i <= pB.x_k$
Rule 2 (location in the closure and contact). $pA.y_i >= pA.y_k \wedge pB.y_i <= pB.y_k$
Rule 3 (difference of size). $A_i <= A_k$

***Entre* (Between):**
Rule 1 masDer $(pC.x_j) < pC.x_i <$ masIzq$(pC.x_k)$ $j \neq k$
masDer $(pC.x_j)$: represents all the objects that are found to the left of the object *i*, further to the right of these and closer to the object *i*.
masIzq $(pC.x_k)$: represents all of the objects found at the right of the object *i*, the object which is more to the left of these and closer to the object *i*.

***Tocar* (To touch):**
Rule 1 (horizontal location). $pC.x_i < pC.x_k$
Rule 2 (contact). $pB.x_i \pm$ range $>= pA.x_k$, "range" is a minimum distance value.

## 4 Example

This section shows an example of the results obtained from the implemented tool SID (Static Image Descriptor).




In the right frame, the Spanish phrases generated by SID from the objects selected in the image, can be seen.

## 5 Conclusions

The present work was accomplished as how a CSA of linguistic units can take part in the creation of rules which comparing hard data, determine the position of the objects in natural language.

For the description of large variety of images it is necessary to realize a CSA of new linguistic units and to generate their rules. This methodology is applicable to other languages; however the CSA and the rules not necessarily are the same [5].

The type of images with which the tests were accomplished are of bmp extension. The images considered are taken at an angle of 180º, that is to say, facing forward and without considering any objects facing each other, since in this work, the background was not considered during image processing. It is important to point out that in Spanish, the preposition *en* (in) embraces the spatial uses of the preposition *sobre* (on) [3]; therefore, the same situation may be described using any of these two prepositions, but the cognitive representation is different. Therefore, *sobre* and *en* depend on the perspective from which the image is taken.

The algorithm fulfilled provides an acceptation result phrase/image of a 90% from a total of 30 images. The 10 % of failing is due to the form in which the objects of the image are segmented. We have a margin of error in the borders of the objects and this has repercussions on the rules. A simple syntactic structure was in use for constructing the phrases [6].

In [7], the authors use the Description Logics to recognize the "intention" or the "prediction", in English, of a static or in movement scene. In [8], the authors realize a description of an image in movement (Soccer). This description is realized for the Germany language. These works integrate a process of vision and other one of natural language as our work.

## References

1. Desclés, J-P.: Langages applicatifs, langues naturelles et cognition. Hermes París (1990)
2. Montes, R., A., Sánchez L., P.: Análisis semántico-cognitivo de verbos y preposiciones en Español para la descripción de imágenes, submitted and accepted to Asociación Mexicana de Lingüística Aplicada (2005)
3. Montes, R., A., Desclés, J-P.: Representación cognitiva y formalización de la preposición y del prefijo. VII Encuentro Internacional de Lingüística en el Noroeste, Sonora, México (2002)
4. Montes, R., A.: Contribution à un modèle quasi-topologique pour la sémantique des langues prépositions et préverbes, Thèse de doctorat, Université de Paris-Sorbonne, Paris IV, France (2002)
5. Sapir, E.: Le Langage, Petite bibliothèque payot, Paris, (1967)
6. García, A., L.: Gramática del español, Madrid, Arco libros, (1994)
7. High-level Vision: http://kogs.informatik.uni-hamburg.de/~neumann/HBD-WS-2004/HLV-Part1-04.pdf
8. Herzog, G., Wazinski, P.: VIsual TRAnslator: Linking Perceptions and Natural Language Descriptions. In: P. Mc Kevitt, ed., Integration of Natural Language and Vision Processing: Computational Models and Systems, Volume 1, pp. 83-95, Kluwer, Dordrecht (1995)

# On Some Optimization Heuristics for Lesk-Like WSD Algorithms*

Alexander Gelbukh[1], Grigori Sidorov[1], and Sang-Yong Han[2,**]

[1] Natural Language and Text Processing Laboratory,
Center for Computing Research, National Polytechnic Institute, 07738, Mexico
{gelbukh,sidorov}@cic.ipn.mx
www.gelbukh.com
[2] Department of Computer Science and Engineering, Chung-Ang University,
221 Huksuk-Dong, DongJak-Ku, Seoul, 156-756, Korea
hansy@cau.ac.kr

**Abstract.** For most English words, dictionaries give various senses: e.g., "*bank*" can stand for a financial institution, shore, set, etc. Automatic selection of the sense intended in a given text has crucial importance in many applications of text processing, such as information retrieval or machine translation: e.g., "(*my account in the*) *bank*" is to be translated into Spanish as "(*mi cuenta en el*) *banco*" whereas "(*on the*) *bank* (*of the lake*)" as "(*en la*) *orilla* (*del lago*)." To choose the optimal combination of the intended senses of all words, Lesk suggested to consider the global coherence of the text, i.e., which we mean the average relatedness between the chosen senses for all words in the text. Due to high dimensionality of the search space, heuristics are to be used to find a near-optimal configuration. In this paper, we discuss several such heuristics that differ in terms of complexity and quality of the results. In particular, we introduce a dimensionality reduction algorithm that reduces the complexity of computationally expensive approaches such as genetic algorithms.

## 1 Introduction

Most words we use in our everyday communication have several possible interpretations, called senses and listed in dictionaries. E.g., the word *bank* can be interpreted as a financial institution, river shore, stock of some objects, etc. For correct understanding of a text, the reader – be it a human being or a computer program – must be able to determine what sense is intended for each word in the text. Apart from message understanding, there are a number of important applications where automatically determining the correct sense of a word is crucial, e.g., information retrieval.

Given a dictionary and a specific occurrence of a word in a specific text, the problem of the choice, out of the senses listed for this word in the dictionary, of the one

* This research was supported by the MIC (Ministry of Information and Communication), Korea, under the Chung-Ang University HNRC-ITRC (Home Network Research Center) support program supervised by the IITA (Institute of Information Technology Assessment).
** Corresponding author.

A. Montoyo et al. (Eds.): NLDB 2005, LNCS 3513, pp. 402–405, 2005.
© Springer-Verlag Berlin Heidelberg 2005

intended for this occurrence is called the word sense disambiguation (WSD) [2]. One of possible approaches to this problem is global optimization of text coherence, i.e., of average relatedness between the chosen senses for all words in the text [5], see Fig. 1. Due to high computational cost of this approach, evolutionary approaches were used to find a near-optimal solution [1, 4].

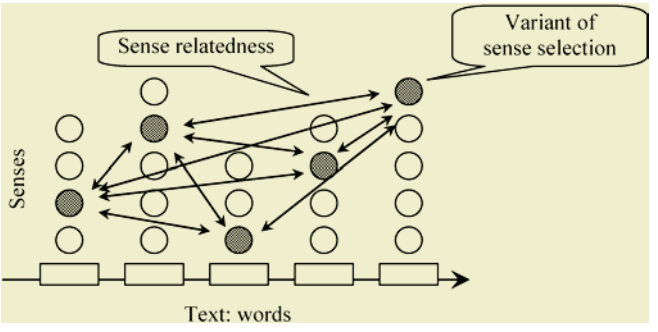



**Fig. 1.** A variant of sense selection and the relatedness measure. The task is to find the variant with the maximum total relatedness.

In this paper, we discuss some heuristics aimed to reduce the complexity of such global optimization. The paper is organized as follows. Section 2 discusses the heuristics for the search algorithm, aimed to fast finding a near-optimal solution. Section 3 introduces a new algorithm for reduction of dimensionality of the search space, useful for computationally expensive approaches. Section 4 presents the experimental results and discussion. Finally, Section 5 concludes the paper.

## 2 Search Heuristics

We denote $N$ the number of words in the text fragment under disambiguation, $n_w$ the number of senses for the word $w$, and $m^{uw}_{s_u s_w}$ the relatedness (a numerical value) between the sense $s_u$ of the word $u$ and $s_w$ of the word $w$. The task in Fig. 1 consists in finding a combination of senses $s_w$ maximizing the value $\sum_{u=1}^{N}\sum_{w=u+1}^{N} m^{uw}_{s_u s_w}$. The complexity of the task is exponential in $N$, namely, $\prod_{w=1}^{N} s_w$.

To select the best combination, exhaustive search [5], simulated annealing [1], or genetic algorithm [4] can be used. A number of heuristics can be tried as well. The heuristics we suggest resemble those used for other combinatorial problems, such as the traveling salesman problem.

- *A greedy approach* that for each word $w$ chooses the sense $s$ with maximum average relatedness to other words:

$$s = \arg\max_s \sum_{u=1,\, u\neq w}^{N} \sum_{i=1}^{n_u} m^{uw}_{is}. \tag{1}$$

The complexity of such an algorithm is only $Nn_w n_{avg}$, where $n_{avg}$ is the average number of senses per word $u \neq w$. This approach is based on the hypothesis that the correct sense of the word $w_i$ is not known and the probability for the sense to be the intended one is distributed uniformly. The average relatedness is that of a given sense to the senses of the other word weighted by the probability of those senses.

- *An even more greedy approach* that for each word $w$ chooses the sense $s$ with maximum relatedness to other words:

$$s = \arg\max_s \{m_{is}^{uw} \mid u \neq w; 1 \leq i \leq n_u\}. \quad (2)$$

The complexity of this algorithm is again $Nn_w n_{avg}$. The latter heuristic seems to be more motivated linguistically than the previous one. Indeed, a word is expected to be immediately related to one (or few) words in the context and not to all surrounding words. However, it is logically inconsistent, since the senses selected for other words can be different from those that influenced the decision for the given one. In this case, the order in which the decisions are made (the words are considered) becomes important. This idea corresponds to the insertion strategies for the traveling salesman problem. Different heuristics can be used to choose the optimal order: (1) *Direct order*: from the first to the last word, (2) *Inverse order*: from the last to the first word, (3) *Greedy order*: at each step, choose the word that gives the best increase in the total coherence value, (4) Another greedy order: choose the word that has a sense with a most clear advantage over other senses of the same word, (5) *A genetic algorithm* can be used to find an optimal order.

## 3 Reduction of Dimensionality

With computationally expensive heuristics discussed above (such as exhaustive search or evolutionary approaches) producing high-quality results, reduction of dimensionality before computation is of great help. For this, some senses can be proven in advance not to give an optimal solution. Namely, a sense $s$ of the word $w$ can be removed from consideration if

$$\max_{s' \neq s} \sum_{u \neq w} \min_{s_u} \left( m_{s_u s'}^{uw} - m_{s_u s}^{uw} \right) \geq 0. \quad (3)$$

Indeed, in this case there exists a sense $s'$ of the same word that gives a better (or at least equal) contribution to the total coherence of the text with any selection of the senses $s_u$ of all other words. In case of equality in the above formula, other criteria can be applied to select one of the two "equal" senses, such as that the senses with smaller numbers in the dictionary are often more frequent and thus more plausible.

The process is repeated iteratively, until no more senses can be excluded. The complexity of this process is polynomial.

## 4 Experimental Results and Discussion

The preliminary results of our experiments can be summarized as follows:

- The algorithms that look for globally optimized solution, such as genetic algorithms, perform (in terms of quality) some 10% better than heuristic approaches.
- Heuristic approaches perform about twice better than the baseline solutions such as random selection.
- Dimensionality reduction algorithm allowed us to reduce the number of senses in a randomly selected sample from 1138 (search space $5.2 \times 10^{95}$) to 433 ($1.4 \times 10^{42}$), which speeds up the genetic algorithm twice. Still, the search space is too huge for exhaustive search. Hence the importance of the heuristic methods.

We also noted that iterating the dimensionality reduction algorithm did not give a considerable gain: most of the removed senses were removed at the first iteration.

## 5 Conclusions

We have suggested some heuristics to improve the speed of global coherence optimization WSD algorithms. In particular, we have described a dimensionality reduction algorithm useful for computationally expensive approaches, such as genetic algorithms. Our experiments show that such reduction can speed up such algorithms approximately twice, though still does not allow for exhaustive search.

In the future, we plan to investigate the effects of linguistically-motivated constraints on sense selection, such as the one-sense-per-discourse heuristic [3].

## References

1. Cowie, J., J. A. Guthrie, L. Guthrie. Lexical disambiguation using simulated annealing. In Proc. of the International Conference on Computational Linguistics, 1992, 359–365.
2. Edmonds, P., and A. Kilgarriff (Eds.), Journal of Natural Language Engineering, Vol. 9 no. 1, 2003. Special issue based on Senseval-2; www.senseval.org.
3. Gale, W., K. Church and D. Yarowsky. *One sense per discourse*. In proc. of the DARPA Speech and Natural Language workshop, Harriman, NY, February 1992.
4. Gelbukh, A., Grigori Sidorov, San-Yong Han. Evolutionary Approach to Natural Language Word Sense Disambiguation through Global Coherence Optimization. *WSEAS Transactions on Communications,* 1(2):11–19, 2003.
5. Lesk, M., Automatic sense disambiguation using machine-readable dictionaries: how to tell a pine cone from an ice cream cone. Proc. of ACM SIGDOC Conference. Toronto, Canada, 1986, p. 24–26.

# Author Index