Zoubida Kedad Nadira Lammari
Elisabeth Métais Farid Meziane
Yacine Rezgui (Eds.)

LNCS 4592

Natural Language Processing and Information Systems

12th International Conference on Applications of Natural Language to Information Systems, NLDB 2007
Paris, France, June 2007, Proceedings

Springer

Zoubida Kedad Nadira Lammari
Elisabeth Métais Farid Meziane
Yacine Rezgui (Eds.)

Natural Language Processing and Information Systems

12th International Conference on Applications
of Natural Language to Information Systems, NLDB 2007
Paris, France, June 27-29, 2007
Proceedings

Springer

Volume Editors

Zoubida Kedad
Laboratoire PRiSM, Université de Versailles, France
E-mail: Zoubida.kedad@prism.uvsq.fr

Nadira Lammari
Elisabeth Métais
Conservatoire National des Arts et Métiers (CNAM)
75141 Paris cedex 3, France
E-mail: {lammari, metais}@cnam.fr

Farid Meziane
University of Salford, Greater Manchester, UK
E-mail: f.meziane@salford.ac.uk

Yacine Rezgui
University of Salford, Informatics Research Institute
Greater Manchester, UK
E-mail: y.rezgui@salford.ac.uk

Library of Congress Control Number: 2007929429

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

LNCS Sublibrary: SL 3 – Information Systems and Application,
incl. Internet/Web and HCI

ISSN 0302-9743
ISBN-10 3-540-73350-7 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-73350-8 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springer.com

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12083701 06/3180 5 4 3 2 1 0

Preface

The 12th International Conference on Applications of Natural Language to Information Systems (NLDB 2007) took place during June 27–29 in Paris (France). Since the first edition in 1995, the NLDB conference has been aiming at bringing together researchers, people working in industry and potential users interested in various applications of natural language in the database and information system areas.

Natural language and databases are core components in the development of information systems. NLP techniques may substantially enhance most phases of the information system lifecycle, starting with requirement analysis, specification and validation, and going up to conflict resolution, result processing and presentation. Furthermore, natural language-based query languages and user interfaces facilitate the access to information for all and allow for new paradigms in the usage of computerized services. Hot topics such as information retrieval and Semantic Web-based applications imply a complete fusion of databases and NLP techniques.

Among an increasing number of submitted papers (110), the Program Committee selected 31 papers as full papers, thus coming up with an acceptance rate of 28%. These proceedings also include 12 short papers that were presented at the conference and two invited talks, one given by Andrew Basden and Heinz Klein and the other given by Max Silberztein.

This conference was possible thanks to the support of three organizing institutions: The University of Versailles Saint-Quentin (Versailles, France), the Conservatoire National des Arts et Métiers (Paris, France) and the University of Salford (Salford, UK). We thank them, and in particular Profs. Akoka (CNAM), Bouzeghoub (UVSQ) and Comyn-Wattiau (CNAM) for their support.

We also wish to thank the entire organizing team including secretaries, researchers and students who put their competence, enthusiasm and kindness into making this conference a real success, and especially Xiaohui Xue, who managed the Web site.

June 2007

Zoubida Kedad
Nadira Lammari
Elisabeth Métais
Farid Meziane
Yacine Rezgui

Conference Organization

Conference Co-chairs

Elisabeth Metais
Jacky Akoka
Mokrane Bouzeghoub
Yacine Rezgui

Program Co-chairs

Zoubida Kedad
Isabelle Comyn-Wattiau
Farid Meziane

Organization Chair

Nadira Lammari

Program Committee

Witold Abramowicz, The Poznañ University of Economics, Poland
Frederic Andres, University of Advanced Studies, Japan
Kenji Araki, Hokkaido University, Japan
Akhilesh Bajaj, University of Tulsa, USA
Maria Bergoltz, Stockholm University, Sweden
Marc El-Beze, CNRS Laboratoire Informatique d'Avignon, France
Béatrice Bouchou, Université François-Rabelais de Tours, France
Mokrane Bouzeghoub, Université de Versailles, France
Andrew Burton-Jones, University of British Columbia, Canada
Hiram Calvo, National Polytechnic Institute, Mexico
Roger Chiang, University of Cincinnati, USA
Gary A Coen, Boeing, USA
Isabelle Comyn-Wattiau, CNAM, France
Cedric Du Mouza, CNAM, France
Antje Düsterhöft, University of Wismar, Germany
Günther Fliedl, Universität Klagenfurt, Austria
Christian Fluhr, CEA, France
Alexander Gelbukh, Instituto Politecnico Nacional, Mexico
Jon Atle Gulla, Norwegian University of Science and Technology, Norway
Udo Hahn, Friedrich-Schiller-Universität Jena, Germany
Karin Harbusch, Universität Koblenz-Landau, Germany

Harmain Harmain, United Arab Emirates University, UAE
Helmut Horacek, Universität des Saarlandes, Germany
Cecil Chua Eng Huang, Nanyang Technological University, Singapore
Paul Johannesson, Stockholm University, Sweden
Epaminondas Kapetanios, University of Westminster, UK
Asanee Kawtrakul, Kasetsart University, Thailand
Zoubida Kedad, Université de Versailles, France
Christian Kop, University of Klagenfurt, Austria
Leila Kosseim, Concordia University, Canada
Nadira Lammari, CNAM, France
Winfried Lenders, Universität Bonn, Germany
Jana Lewerenz, sd&m Düsseldorf, Germany
Deryle Lonsdale, Brigham Young University, USA
Stéphane Lopes, Université de Versailles, France
Robert Luk, Hong Kong Polytechnic University, Hong Kong
Bernardo Magnini, IRST, Italy
Heinrich C. Mayr, University of Klagenfurt, Austria
Paul McFetridge, Simon Fraser University, Canada
Elisabeth Metais, CNAM , France
Farid Meziane, Salford University, UK
Luisa Mich, University of Trento, Italy
Ruslan Mitkov, University of Wolverhampton, UK
Diego Mollá Aliod, Macquarie University, Australia
Andrés Montoyo, Universidad de Alicante, Spain
Ana Maria Moreno, Universidad Politecnica de Madrid, Spain
Rafael Muñoz, Universidad de Alicante, Spain
Samia Nefti-Meziani, Salford University, UK
Günter Neumann, DFKI, Germany
Jian-Yun Nie, Université de Montréal, Canada
Manual Palomar, Universidad de Alicante, Spain
Pit Pichappan, Annamalai University, India
Odile Piton, Université Paris I Panthéon-Sorbonne, France
Violaine Prince, Université Montpellier 2/LIRMM-CNRS, France
Sandeep Purao, Pennsylvania State University, USA
Yacine Rezgui, University of Salford, UK
Reind van de Riet, Vrije Universiteit Amsterdam, The Netherlands
Hae-Chang Rim, Korea University, Korea
Samira si-Said, CNAM, France
Grigori Sidorov, Instituto Politecnico Nacional, Mexico
Max Silberztein, Université de Franche-Comté, France
Veda Storey, Georgia State University, USA
Vijayan Sugumaran, Oakland University Rochester, USA
Lua Kim Teng, National University of Singapore, Singapore
Bernhard Thalheim, Kiel University, Germany
Krishnaprasad Thirunarayan, Wright State University, USA
Juan Carlos Trujillo, Universidad de Alicante, Spain
Luis Alfonso Ureña, Universidad de Jaén, Spain

Panos Vassiliadis, University of Ioannina, Greece
Jürgen Vöhringer, University of Klagenfurt, Austria
Roland Wagner, University of Linz, Austria
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

Jing Bai, Norman Biehl, Terje Brasethevik, Gaël De Chalendar, Hiroshi Echizen-Ya, Óscar Ferrandez, Miguel A.Garcia, Gregory Grefenstette, Trivikram Immaneni, Jon Espen Ingvaldsen, Zornitsa Kozareva, Teresa Martin, Fernando Martinez-Santiago, Arturo Montejo-Raez, Borja Navarro, Octavian Popescu, Rafal Rzepka, Agata Savary, Hideyuki Shibuki, Jonas Sjobergh, Darijus Strasunskas, David Tomas, Stein L.Tomassen

Table of Contents

Invited Paper

An Alternative Approach to Tagging 1
Max Silberztein

Natural Language for Database Query Processing

An Efficient Denotational Semantics for Natural Language Database Queries 12
Richard A. Frost and Randy J. Fortier

Email Management

An Approach to Hierarchical Email Categorization Based on ME 25
Peifeng Li, Jinhui Li, and Qiaoming Zhu

Developing Methods and Heuristics with Low Time Complexities for Filtering Spam Messages 35
Tunga Güngör and Ali Çıltık

Semantic Annotation

Exploit Semantic Information for Category Annotation Recommendation in Wikipedia 48
Yang Wang, Haofen Wang, Haiping Zhu, and Yong Yu

A Lightweight Approach to Semantic Annotation of Research Papers . . . 61
Nicola Zeni, Nadzeya Kiyavitskaya, Luisa Mich, John Mylopoulos, and James R. Cordy

Text Clustering

A New Text Clustering Method Using Hidden Markov Model 73
Yan Fu, Dongqing Yang, Shiwei Tang, Tengjiao Wang, and Aiqiang Gao

Identifying Event Sequences Using Hidden Markov Model 84
Kei Wakabayashi and Takao Miura

The Dictionary-Based Quantified Conceptual Relations for Hard and Soft Chinese Text Clustering 96
Yi Hu, Ruzhan Lu, Yuquan Chen, Hui Liu, and Dongyi Zhang

On-Line Single-Pass Clustering Based on Diffusion Maps 107
Fadoua Ataa Allah, William I. Grosky, and Driss Aboutajdine

Selecting Labels for News Document Clusters 119
Krishnaprasad Thirunarayan, Trivikram Immaneni, and Mastan Vali Shaik

Ontology Engineering

Generating Ontologies Via Language Components and Ontology Reuse 131
Yihong Ding, Deryle Lonsdale, David W. Embley, Martin Hepp, and Li Xu

Experiences Using the ResearchCyc Upper Level Ontology 143
Jordi Conesa, Veda C. Storey, and Vijayan Sugumaran

From OWL Class and Property Labels to Human Understandable Natural Language 156
Günther Fliedl, Christian Kop, and Jürgen Vöhringer

Ontological Text Mining of Software Documents 168
René Witte, Qiangqiang Li, Yonggang Zhang, and Juergen Rilling

Natural Language for Information System Design

Treatment of Passive Voice and Conjunctions in Use Case Documents 181
Leonid Kof

Natural Language Processing and the Conceptual Model Self-organizing Map 193
Ricardas Laukaitis and Algirdas Laukaitis

Automatic Issue Extraction from a Focused Dialogue 204
Koen V. Hindriks, Stijn Hoppenbrouwers, Catholijn M. Jonker, and Dmytro Tykhonov

Information Retrieval Systems

Character N-Grams Translation in Cross-Language Information Retrieval 217
Jesús Vilares, Michael P. Oakes, and Manuel Vilares

Cross-Lingual Information Retrieval by Feature Vectors 229
Jeanine Lilleng and Stein L. Tomassen

Incomplete and Fuzzy Conceptual Graphs to Automatically Index Medical Reports 240
Loic Maisonnasse, Jean Pierre Chevallet, and Catherine Berrut

Combining Vector Space Model and Multi Word Term Extraction for Semantic Query Expansion 252
Eric SanJuan, Fidelia Ibekwe-SanJuan, Juan-Manuel Torres-Moreno, and Patricia Velázquez-Morales

The Bootstrapping Based Recognition of Conceptual Relationship for Text Retrieval 264
Yi Hu, Ruzhan Lu, Yuquan Chen, Xiaoying Chen, and Jianyong Duan

A Framework of NLP Based Information Tracking and Related Knowledge Organizing with Topic Maps 272
Asanee Kawtrakul, Chaiyakorn Yingsaeree, and Frederic Andres

Natural Language Processing Techniques

DLSITE-1: Lexical Analysis for Solving Textual Entailment Recognition 284
Óscar Ferrández, Daniel Micol, Rafael Muñoz, and Manuel Palomar

Text Segmentation Based on Document Understanding for Information Retrieval 295
Violaine Prince and Alexandre Labadié

Named Entity Recognition for Arabic Using Syntactic Grammars 305
Slim Mesfar

Four Methods for Supervised Word Sense Disambiguation 317
Kinga Schumacher

Enhancing Relation Extraction by Eliciting Selectional Constraint Features from Wikipedia 329
Gang Wang, Huajie Zhang, Haofen Wang, and Yong Yu

A Computer Science Electronic Dictionary for NOOJ 341
Farida Aoughlis

Applying Wikipedia's Multilingual Knowledge to Cross–Lingual Question Answering 352
Sergio Ferrández, Antonio Toral, Óscar Ferrández, Antonio Ferrández, and Rafael Muñoz

Zero Anaphora Resolution in Chinese and Its Application in Chinese-EnglishMachine Translation 364
Jing Peng and Kenji Araki

Short Papers

Rule-Based Partial MT Using Enhanced Finite-State Grammars in NooJ 376
Tamás Váradi

Biomedical Named Entity Recognition: A Poor Knowledge HMM-Based Approach 382
Natalia Ponomareva, Ferran Pla, Antonio Molina, and Paolo Rosso

Unsupervised Language Independent Genetic Algorithm Approach to Trivial Dialogue Phrase Generation and Evaluation 388
Calkin S. Montero and Kenji Araki

Large-Scale Knowledge Acquisition from Botanical Texts 395
François Role, Milagros Fernandez Gavilanes, and Éric Villemonte de la Clergerie

Lexical-Based Alignment for Reconstruction of Structure in Parallel Texts 401
Alexander Gelbukh, Grigori Sidorov, and Liliana Chanona-Hernandez

Electronic Dictionaries and Transducers for Automatic Processing of the Albanian Language 407
Odile Piton, Klara Lagji, and Remzi Përnaska

Two Methods of Evaluation of Semantic Similarity of Nouns Based on Their Modifier Sets 414
Igor A. Bolshakov and Alexander Gelbukh

A Service Oriented Architecture for Adaptable Terminology Acquisition 420
Farid Cerbah and Béatrice Daille

Domain Relevance on Term Weighting 427
Marko Brunzel and Myra Spiliopoulou

Flexible and Customizable NL Representation of Requirements for ETL processes 433
Dimitrios Skoutas and Alkis Simitsis

Author Index 441

An Alternative Approach to Tagging

Max Silberztein

LASELDI, Université de Franche-Comté
30 rue Mégevand, 25000 Besançon, France
max.silberztein@univ-fcomte.frUT

Abstract. NooJ is a linguistic development environment that allows users to construct large formalised dictionaries and grammars and use these resources to build robust NLP applications. NooJ's approach to the formalisation of natural languages is bottom-up: linguists start by formalising basic phenomena such as spelling and morphology, and then formalise higher and higher linguistic levels, moving up towards the sentence level. NooJ provides parsers that operate in cascade at each individual level of the formalisation: tokenizers, morphological analysers, simple and compound terms indexers, disambiguation tools, syntactic parsers, named entities annotators and semantic analysers. This architecture requires NooJ's parsers to communicate via a Text Annotation Structure that stores both correct results and erroneous hypotheses (to be deleted later).

Keywords: NooJ. Linguistic Development Environment. Robust NLP applications.

1 Introduction

NooJ is a linguistic development environment that allows users to construct large linguistic resources in the form of electronic dictionaries and grammars and to apply these resources to large texts and build various robust NLP applications.[1]

NooJ's approach to the formalisation of natural languages is bottom-up: linguists start by formalising basic phenomena, such as spelling, morphology and lexicon, and then use these basic levels of description to formalise higher and higher linguistic levels, moving up towards the syntactic and semantic levels. This bottom-up approach is complemented by an accumulative methodology that allows a community of users to share and re-use individual resources, as well as a number of tools (e.g. concordances, contract enforcers and debuggers) that help users maintain the integrity of large resources.

Parallel to these various levels of formalisation, NooJ provides a number of parsers that operate in cascade at each individual level of the formalisation: at the character level (tokenizer and sentence recogniser), morphology (inflectional and derivational analyser), lexical (simple and compound words recognisers), local syntax (disambiguation), structural syntax (frozen and semi-frozen expressions recogniser, syntactic parser) and transformational syntax (semantic analyser).

[1] NooJ is freeware. See: http://www.nooj4nlp.netUT to download the software and its documentation.

Z. Kedad et al.(Eds.): NLDB 2007, LNCS 4592, pp. 1–11, 2007.

2 A Third Type of NLP Tool

Most available NLP tools follow one of two different and incompatible approaches.

On the one hand, some linguistic parsers aim at formalising natural languages, usually at the syntactic and semantic levels. Following Chomsky's discussion of the inadequacies of finite-state machines for NLP [1], researchers have invented and refined several computational devices and their corresponding formalisms capable of representing complex, non finite-state syntactic phenomena, such as unification-based parsers that deal with various types of agreement constraints.

Unfortunately most of these parsers, while powerful enough to compute a variety of complex syntactic analyses, are not adapted to the processing of very simple but cost-intensive phenomena, such as locating multi-word expressions in texts by accessing a dictionary of over 200,000 entries, performing morphological analysis of Hungarian texts, etc. Moreover, they are not capable of parsing large corpora in real-time, and therefore cannot be used as online corpus processing tools, nor can they be used as linguistic engines for "basic" applications such as search engines.

On the other hand, some NLP tools aim at facilitating the implementation of NLP applications such as search engines, automatic construction of abstracts, corpus processors, information extraction, etc. These tools often include very efficient parsers based on finite-state technology, and can indeed be used to parse large quantities of texts. Unfortunately, these tools include at one point or another several algorithms that make them unsuitable to the formalisation of natural languages, such as a statistical tagger that aims at producing "reasonably good" results – which is to say a number of incorrect ones – as well as heuristics to get rid of ambiguities – even when sentences are genuinely ambiguous – etc.

NooJ shares with the above-mentioned linguistic tools the goal of providing linguists with a way to formalise natural languages precisely, and at the same time includes several efficient finite-state tools to parse large texts and process large linguistic resources. This can be done because in NooJ's bottom-up architecture, each level of analysis is processed by a different, specialised (and therefore efficient) computational device. In other words, instead of using one single powerful (and inefficient) computational device to process all kinds of linguistic phenomena, we assume that natural languages are sets of very different phenomena, each of them requiring a specialized mechanism and associated parser; in particular, simple devices such as finite-state machines, which constitute very natural tools to represent a large number of linguistic phenomena, should not be thrown away because they are not adequate to represent other, sometimes even exotic, phenomena.

3 Atomic Linguistic Units

In NooJ, the term **Atomic Linguistic Units** (ALUs) refers to the smallest elements of a given language that are associated with linguistic information. By definition, these ALUs constitute the vocabulary of the language. They can and must be systematically described in extension, because some of, or all their properties cannot be computed from their components.

NooJ's first level of text analysis is at the character level. Characters are classified as **letters** and **delimiters**. **Tokens** are sequences of letters between delimiters. Based on these three definitions, NooJ distinguishes four types of ALUs:

- **Simple Word**: any ALU spelled as a token, e.g. *table*
- **Affix**: any ALU spelled as a subsequence of letters in a token, e.g. *re-, -able*
- **Multi-Word Unit** (MWU): any ALU spelled as a sequence of letters and delimiters, e.g. *as a matter of fact* (the space character is not a letter, hence it is a delimiter)
- **Frozen Expression**: any MWU that accepts insertions, e.g. *take ... into account*

This classification is adequate for any written language, although some languages (e.g. English) require the description of very few affixes whereas others (e.g. Hungarian) require a large morphological system. The vocabulary of Romance languages probably contains five times more MWUs than simple words, whereas Germanic languages have very few MWUs because these units are not spelled with spaces, and hence are processed by NooJ as simple words.[2]

Obviously, it is important for any NLP application to be able to process any type of ALUs, including MWUs. Questions about MWUs – their definition, how to automatically recognise them in texts, how to process them, etc. – have initiated much interest in computational linguistics. Unfortunately, there seems to be confusion about what people mean by MWUs (or compounds, or complex terms, collocations, etc.). For instance, [2] studies "transparent" noun compounds and discusses methods for analysing them. In the NooJ framework, this is contradictory: either these objects are truly MWUs (i.e. **Atomic** Linguistic Units) that must be listed and explicitly described in dictionaries (and therefore don't need to be analysed), or they are transparent and can be analysed automatically (and therefore do not need to be listed explicitly in a dictionary). Note finally that more precise analyses (such as [3]) show that so-called "transparent" MWUs have in fact some degrees of "opacity" that would require robust NLP application to list and describe them explicitly in a dictionary.

Statisticians often equate compounds with collocations, i.e. sequences of tokens that occur together frequently. But trying to characterise MWUs by computing the co-occurrence of their components negates their atomicity.[3] Moreover, MWUs (just like simple words) can be either frequent or rare in texts, and MWUs with a low frequency (say, less than 3 occurrences in a large corpus), are typically left out by statistical methods.

[2] On the other hand, Germanic analysable tokens, such as "*Schiffahrtsgesellschaft*" must be processed as sequences of affixes: "*Schiff fahrt s gesellschaft*".

[3] In the same manner that one should not try to prove the fact that the token "apartment" is an English word from the fact that "apart" and "ment" often occurs together. In fact, tokenizers used by most taggers and statistical parsers naively use the blank character to characterise linguistic units. In NooJ, MWUs are ALUs just like simple words; the fact that they include blanks or other delimiters is not relevant to their status, and does not even complicate their automatic recognition in texts.

4 Processing Multi-word Units

NLP applications would gain significant advantages were they to recognise MWUs in texts just by consulting pre-built dictionaries, or by accessing tagged texts in which these units were fully represented. These include:

(1) Applications that extract and count words in order to characterise parts of a corpus for information retrieval, summarisation, question answering, etc. For instance, indexing tokens such as "plant", "control", "group", "board", etc. produces too many useless hits, whereas indexing MWUs such as "wild plant" or "power plant" produces much more precise results. Moreover, distinguishing terms (e.g. *parking lots*) from non-terms (ex. … *lots of cars…*) will no longer raise complicated issues.

(2) Ranking systems subsequently select among the texts retrieved by the search engine those that will be most useful to the user. Ranking programs are said to be based on *word* counts, although what they really count are *tokens*. One problem is that most terms have abbreviated variants that are often more frequent than their explicit form. For instance, in articles that include the term *school board*, the abbreviated variant "board" will often occur more frequently than the explicit form. Current ranking systems fail to recognise a term when only its abbreviation occurs. The following enhancement could easily be implemented: when a MWU is recognised, all its variants that occur within a certain context should be linked to it. For instance, if an indexer found the term "school board" in a text, the subsequent utterances of the token "board" within the same paragraph could be linked to it. Thus, the term would receive a much higher frequency even though its explicit form might have occurred only once.

(3) In Question Answering applications, one might want to retrieve a text such as:

Last presidential election was won by John Smith ...

from queries such as: 'What was the election about?' or 'Who was elected?'. This is typically done by using a rule such as "WORD election = people elect WORD". But then, we need to inhibit this rule for a number of terms such as:

local election, runoff election, general election, free election...

because people do not elect "locals", "runoffs", "generals" or "freedom." The only proper way[4] to correctly analyse relevant phrases while rejecting all irrelevant ones is to enter explicitly the rules that are associated with each MWU:

presidential election = people elect someone president
party election = a party elects its leader/representative
free election = an election is free = people are free to participate in the election

(4) In multilingual applications, such as the teaching of a second language or Machine Translation (MT), text mining of multilingual corpora, etc. it is crucial to process MWUs. For instance, if the adverb "as a matter of fact" is treated as a sequence of five words, a translator will produce the incorrect French text "comme une matière de fait" rather than the correct term "à vrai-dire." Reciprocally, the

[4] In an equivalent manner, [4] discusses how to link MWUs to their semantic representation, e.g. *prefabricated housing* and *produce (factory, house).*

French MWU “traitement de texte” must be translated as “word processor”, rather than as “treatment of text” or “text treatment.”

MT is one of the most ambitious NLP applications because it requires a precise description of both source and target languages. Moreover, as opposed to syntactic parsers that produce abstract data structures or information retrieval systems that access texts containing billions of words, any native speaker can easily judge the quality of its results. Appendix 1 contains an extract from a French newspaper along with a translation that was produced by current automatic machine translation systems: the reader can judge the quality (or lack thereof) of the result, and evaluate the enormous advantage of having the computer access a dictionary of MWUs.

5 Tagging All ALUs

Tagged texts are mainly used as inputs for syntactic or statistical parsers. Unfortunately, most taggers associate tags to simple words without taking other types of ALUs into account[5] Let us illustrate this by studying the following text sample:[6]

Battle-tested Japanese industrial managers here always buck up nervous newcomers with the tale of **the first of their** countrymen to visit Mexico, a boatload of samurai warriors blown ashore **375 years ago.** From the beginning, it took a man with extraordinary qualities to succeed in Mexico, says Kimihide Takimura, president of Mitsui group's Kensetsu Engineering Inc. unit.

Fig. 1. Multi-word units are underlined in a text

– “Battle-tested”: metaphorical adjective synonymous with “experienced.” One cannot expect a semantic parser to infer the idiosyncratic analysis, or an automatic translator to compute the correct word (e.g. the French adjective “aguerris”) from the two components “Battle” (“Bataille”) and “tested” (“testés”).

– “industrial manager”: must *not* be analysed by the semantic rule:

Industrial N = N is made/created by industrial methods

although that this very productive rule does apply to numerous sequences, such as “industrial diamond”, “industrial food”, “industrial soap”, “industrial chicken”, etc.

– “to buck up”: not to be processed as the free verb *to buck* that would be followed by the locative preposition “up”;

– “a boatload of”: must be tagged as a determiner, and *not* as a noun, because we want the following human noun *warriors* to be the head of the noun phrase;

– “samurai warriors” = explicit variant of the term “samurais”; not to be analysed as “N warriors” such as “Japanese warriors” or “monk warriors” (all samurais *are* warriors, by definition);

[5] Affixes, MWUs and frozen expressions are usually not taken into account by taggers, even when the goal is to locate Noun Phrases, see for instance the NP recogniser described in [5].

[6] The tagging of this text is commented on in [6], but there is no mention of its MWUs.

– “blown ashore”: not to be processed as the free verb *to blow* followed by the locative adverb “ashore”, even though the metaphor is clearer (to a human reader, *not* to an automatic parser)

– “from the beginning”: adverbial complement of date

– “It takes NP to VP”: frozen expression synonymous with “Only NP can VP”

Ignoring any of these MWUs during the tagging process is extremely costly, because it leads to unavoidable mistakes during the subsequent parsing. Note that none of these ALUs are indeed correctly translated by the machine translation systems mentioned in Appendix 1.

A more useful tagger, i.e. one that takes all ALUs (including MWUs and frozen expressions) into account would produce a result similar to the following one:[7]

```
<Battle-tested,A> <Japanese,A> <industrial
manager,N+Hum+p> <here,ADV+Loc> <always,ADV> <buck
up,V+PR> <nervous,A> <newcomer,N+Hum+p> <with,PREP>
<the,DET+s> <tale,N+s> <of,PREP> <the first of
their,DET+p> <countryman,N+Hum+m+p> <to,PREP>
<visit,V+INF> <Mexico,N+Proper+Loc>, <a boatload
of,DET+p> <samurai warrior,N+Hum+p> <blow ashore,V+PP>
<375 years ago,ADV+Date>.
<From the beginning,ADV+Date>, <it,PRO+EXP01+3+s>
<take,V+EXP01+PT> <a,DET+s> <man,N+Hum+m+s> <with,PREP>
<extraordinary,A> <quality,N+p> <to,PREP+EXP01>
<succeed,V+INF> <in,PREP> <Mexico,N+Proper+Loc>,
<say,V+PR+3+s> <Kimihide Takimura,N+Proper+Hum+s>,
<president,N+Hum+s> <of,PREP> <Mitsui,N+Proper+Company>
<group,N+s> <'s,POSS> <Kensetsu Engineering
Inc.,N+Proper+Company> <unit,N+s>.
```

Fig. 2. A Part of Speech tagger that tags all types of ALUs

In this tagged text, all ALUs are represented by one tag, exactly like simple words. The constituents of the frozen expression “it takes NP to VP” are marked with the I.D. number “EXP01” in order to allow subsequent parsers to retrieve the properties of the expressions and link the components together.[8]

In this sample of 59 tokens, 31 tokens are in fact components of MWUs, frozen expressions and semi-frozen expressions (such as “375 years ago”), and thus are usually tagged with misleading tags (such as “boatload=N”), incorrect ones (e.g. “ago=ADV”), redundant ones (“warriors”) or irrelevant ones (“Engineering”) by

[7] In NooJ, tags (between brackets) associate each ALU with its lemma and linguistic information. “A” stands for Adjective, “N” for Noun, “V” for Verb. Morphological, syntactic and distributional features are prefixed by a character “+”, e.g. “+Hum” stands for human; “+t” stands for transitive; +p stands for plural. See [7] for a description of NooJ’s codes.

[8] I assume that there is some provision to process idiomatic and frozen expressions at the syntactic analysis stage.

traditional taggers. The high frequency of MWUs and frozen expressions is far from being exceptional, as [9] showed. Another example is displayed in Appendix 2.

6 An Ambiguous Text Annotation Structure

The best possible outcome would be to produce a tagged text such as the one in figure 2; unfortunately it is not generally possible to remove all ambiguities without performing a complex series of analyses. For instance, in the following context:

There is a round table in this building ...

there *are* two possible analyses: either the MWU "round table" (= a meeting) occurs, or the sequence of two simple words "round" and "table" (= a table with a round shape). It would require a complex discourse analysis of a broader context of this sentence to solve this ambiguity, and any local parser, such as a statistical-based algorithm trained on some other text, or a simple program that either ignores MWUs, or always tags them, is going to produce a significant number of errors. In NooJ's architecture however, no correct hypothesis can be missing (a "no silence" constraint), because in NooJ's entire flow of analyses, later parsers cannot go back to correct their inputs. In other words, if it is possible that the MWU "round table" occurs in the text, it has to be tagged, and, at the same time, if it is possible that the two ALUs "round" and "table" occur in the text, they too have to be tagged. Generally, ambiguities can be solved only at a phase well after the tagging process.

Although annotations or XML tags are widely used to represent linguistic units such as named entities, very few taggers have used them to represent the massively ambiguous results of the lexical analysis of a text.[9]

In order to maintain both hypotheses alive during all the analyses, NooJ's consecutive parsers communicate their results via a Text Annotation Structure which stores both correct results and erroneous hypotheses. Each of NooJ's parsers – at the character, morphological, lexical, syntactic and semantic levels – takes a Text Annotation Structure as its input, and produces a Text Annotation Structure, such as the one shown in Appendix 3.

NooJ's annotation system represents all ALUs in the same exact way, whether they were produced by a morphological parser (affixes), by a lookup of a dictionary (simple words and MWUs) or by local grammars (for discontinuous frozen expressions). Indeed, after NooJ has completed the full lexical analysis of a text, the following syntactic parsers do not even need to know the original type of each ALU that they are processing.

Processing Text Annotation Structures is more complex than processing simple sequences of tags. For instance, processing a simple query such as locating in a text all the word forms "the" followed by a plural Noun, involves using a parser that can read both the text and its (ambiguous) annotations; that is substantially heavier than running a simple "grep" command on traditionally tagged texts.

An important advantage of this architecture is that several applications, such as Named Entity Recognisers, can run on Text Annotation Structures even if these have not been fully disambiguated. For instance, to recognise the sequence "on the 12th of May, 2006", NooJ's TIMEX recogniser does not need to disambiguate the word form

[9] INTEX used finite-state transducers to represent the results of its lexical parser, see [9].

"May" (the verb or the month name): indeed one can argue that the very fact that TIMEX has annotated the previous sequence as an adverb makes the disambiguation of the word form "May" trivial, and even unnecessary.

7 Conclusion

NooJ is currently used as a linguistic tool to formalise more than 20 languages, and NooJ modules are already available for download for Arabic, Armenian, Catalan, Chinese, English, French, Hebrew, Hungarian, Italian, Latin, Portuguese and Spanish. These modules take into account the four types of NooJ ALUs: affixes, simple words, MWUs and frozen expressions.

Because NooJ's architecture guarantees a "zero silence" (though at the cost of some inevitable noise), it becomes possible to build truly robust NLP applications. Several NLP applications already use NooJ as their linguistic engine, including the current French project VODEL from the National Agency for Research (ANR), which aims at parsing automatically medical texts written in French.

References

1. Chomsky, N.: Syntactic Structures. La Haye, Monton (1957)
2. Barker, K., Szpakowicz, S.: Semi-Automatic Recognition of Noun Modifier Relationships. In: Proceedings of COLING-ACL'98. Montréal (1998)
3. Silberztein, M.: Les groupes nominaux productifs et les noms composés lexicalisés. In: Linguisticae Investigationes #17:2., John Benjamins, Amsterdam (1993)
4. Lewis, D., Sparck Jones, K.: Natural Language Processing for Information Retrieval. In: Communications of the ACM #39:1, ACM, New York (1996)
5. Church, K.: A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text. In: Proceedings of Second Conference on Applied Natural Language Processing (ANLP'88), Austin TX (1988)
6. Marcus, M., Santorini, B., Marcinkiewicz, M.A.: Building a large annotated corpus of English: the Penn Treebank. In: Computational Linguistics #19, The MIT Press, Cambridge (1993)
7. Silberztein, M.: NooJ's Manual can be downloaded from (2002), http://www.nooj4nlp.net
8. Silberztein, M.: Dictionnaires électroniques et analyse automatique de textes. Masson. Paris (1993)
9. Silberztein, M.: Dictionnaires électroniques et comptage des mots. In: Proceedings of Troisièmes Journées d'Analyse des Données Textuelles (JADT). Rome (1995)

Appendix 1: Translation of a Sample from *Le Monde*, January 1994

In the following French text, I have underlined MWUs (e.g. vie privée) and verbal expressions (tomber sous le coup de la loi). Semi-frozen expressions are in bold.

"Bonne année". **L'an 1994** aura commencé **samedi à zéro heure**, qui en douterait ? Et pourtant, cette évidence est pour le moins trompeuse. C'est ainsi que, décalage horaire aidant, les Australiens de Sydney fêtent la Saint-Sylvestre **avec neuf heures d'avance sur** nous et les Américains de Los Angeles **avec neuf**

heures de retard. Un détail ? Admettons. Mais que dire, alors, des musulmans pratiquants qui, eux, se considèrent **à la mi-1414**, et célébreront leur "nouvel an" (l'hégire) **le 10 juin** ?

Quant aux juifs, leur **année 5754** a débuté **le 16 septembre**. Incontestablement, le temps est une notion toute relative. Les physiciens savent cela depuis longtemps. Les politiques et les religieux aussi. Mais, grands amateurs de certitudes, ces derniers se sont alliés aux scientifiques pour tenter d'offrir aux hommes un guide chronologique susceptible de régler une vie sociale qu'ils voulaient **aussi** harmonieuse **que possible**. Ce n'est pas un hasard si le calendrier grégorien que nous utilisons actuellement **en Occident** fut imposé (sous peine d'excommunication !) **il y a quatre siècles** par un pape chagriné de voir le jour de Pâques s'éloigner de la période que lui avaient fixée ses prédécesseurs **en l'an 325** !

Following is the best result produced by the French-to-English translators available at: amikai.com, freetranslation.com, officeupdate.lhsl.com, reverso.net, systransoft.com, t-mail.com, tranexp.com, translationwave.com. Mistakes are crossed out:

" ~~Good year~~ ". ~~Will~~ ~~the year 1994~~ have begun Saturday ~~at zero hour~~, ~~which would doubt it~~? And yet, this obviousness is at the very least misleading. Thus, ~~time shift~~ helping, the Australian ~~ones~~ of Sydney celebrate New Year's Eve ~~with nine hours in advance on~~ us and the Americans of Los Angeles ~~with nine hours of delay~~. A detail? Let us admit [it]. But what to say, then, of the ~~Moslems practise~~ who, ~~them~~, ~~are considered with~~ ~~semi-1414~~, and will celebrate their " new year " (the hégire) on June 10? As for the Jews, their year 5754 began on September 16. Incontestably, time is a quite relative concept. The physicists [have known] that for a long time. ~~Policies~~ and ~~monks~~ too. But, ~~large amateurs~~ of certainty, the latter ~~were combined to~~ the scientists to try to offer to ~~the~~ men a chronological guide suitable ~~for~~ regulate a ~~social life~~ which they wanted ~~possible as~~ harmonious ~~as~~. ~~It is not a chance if~~ the Gregorian calendar that we currently use ~~in Occident~~ ~~were~~ imposed (under penalty of excommunicaction!) four centuries ago by a pope ~~grained~~ to see ~~the Easter Day~~ moving away ineluctably from the period that its predecessors ~~in year 325~~ had fixed ~~to him~~!

Note that 18 of the mistakes would have been avoided by a simple lookup of a bilingual dictionary that would include the following MWUs:

Bonne année = Happy new year; qui en douterait ? = who would doubt it?; et pourtant = and yet; pour le moins = at the very least; C'est ainsi que = thus; décalage horaire = time difference; la Saint-Sylvestre = New Year's Eve; musulman pratiquant = practising Muslim; se considérer = to considere oneself to be; nouvel an = new year; depuis longtemps (+Présent) = for a long time (+Present Perfect); grands amateurs = big fans; s'allier à = to ally oneself with; vie sociale = life in society; Ce n'est pas un hasard si = it is not by accident if; calendrier grégorien = Gregorian calendar; sous peine de = under penalty of; jour de Pâques = Easter

Appendix 2: Sample from the Wall Street Journal, January 1996

I have underlined various types of MWUs: nouns (e.g. monetary union), adverbs (in the process), proper names (U.S.), prepositions (according to) and verbal expressions (cast a shadow on). Auxiliary and transparent verbs have been crossed out (e.g. ~~may be~~ heading). In bold are semi-frozen expressions for determiners (**one of the best**), adverbs of date (**in 1995**), and of duration (**in the space of one year**).[10]

Europe ~~may be~~ heading toward economic convergence and monetary union, but its stock markets still set differing courses.

Just as **in 1995**, when some countries produced high double-digit returns while others posted substantial losses, investors ~~can~~ expect a grab bag of returns this year. On average European stocks rose 12.3% **last year**, according to the Eurotop 100 index, which is far below the stratospheric returns of U.S. stocks. But returns in individual European countries ranged **as high as** 20.3% in Zurich, 17.1% in Dublin and 16.9% in London and **as low as** a decline of 8.36% in Helsinki, 7.81% in Milan and 0.49% in Paris. And the performance of specific sectors also varied to a large degree with financial, technology and pharmaceutical stocks producing returns ranging from 21% to 50%. Paper, steel and heavy machinery issues had a rough year, however, with losses ranging from 4% to 26%.

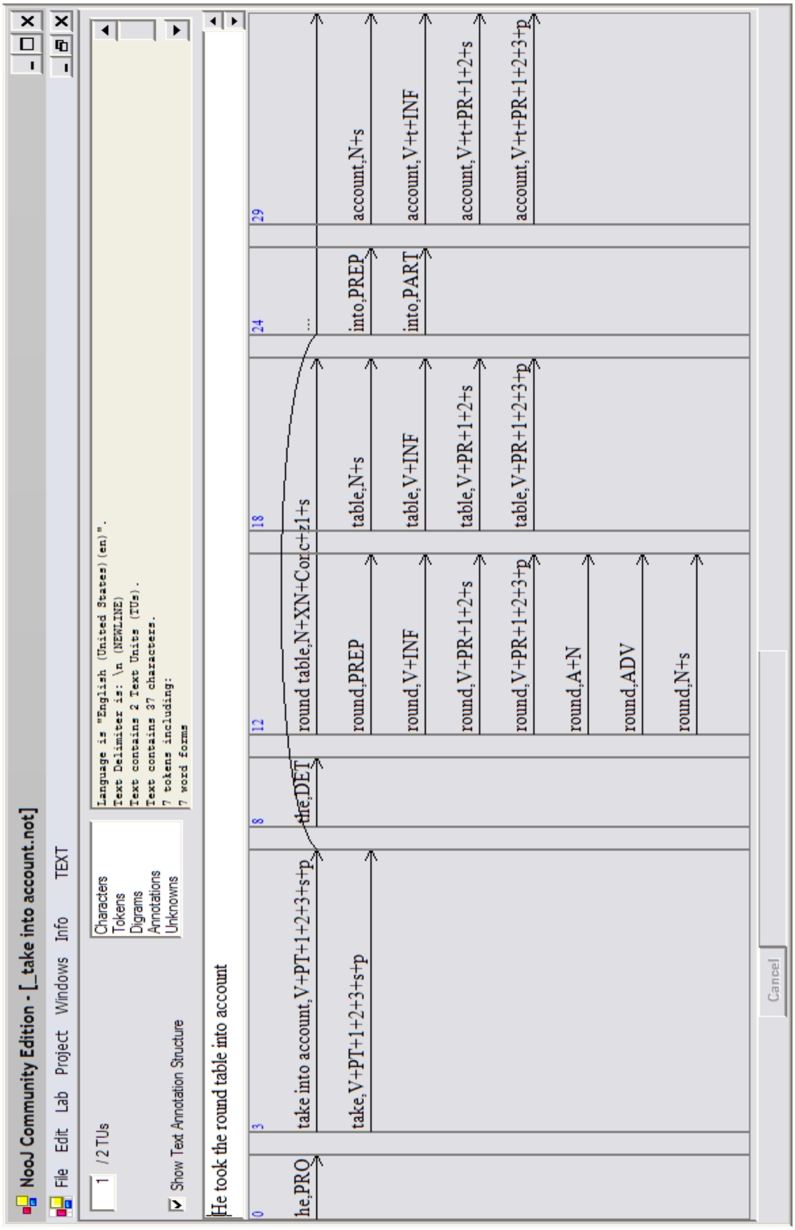
LONDON: Merger activity and low interest rates sent U.K. stocks soaring **last year**. The Financial Times-Stock Exchange 100 Share Index climbed 16.9%, finishing the year at 3689.3. The U.K. market had **one of the best** performances in Europe, with the index gradually rising from a low of 2876.60 **at the beginning of the year**.

A good year for British government bonds, or gilts, also ~~helped~~ propel U.K. stocks upward. The average yield dropped to 7.4% from 8.7% **at the beginning of the year**. Richard Kersley, U.K. market strategist with BZW Ltd. in London, says the change from rising to falling interest rates sparked **much of the** stock market climb **in 1995**. **In mid-December**, the British government cut the base interest rate by a quarter percentage point to 6.50% and analysts say further interest rate cuts ~~are~~ expected **early this year**. FRANKFURT: The strengthening of the mark cast a shadow on Germany's stock prices **throughout last year**, forcing earnings revisions downward **for 1995, 1996 and 1997** in export-oriented companies. The DAX Index, the key gauge of 30 stocks traded at the Frankfurt Stock Exchange, fell to 2253.88 points, chalking up only a 6.54% gain **since the beginning of the year**. **Last year**, it fell to a low of 1893.63 and climbed to a high of 2320.22. But on the positive side, declining German bond yields helped the stock market. **In the space of one year**, the **10-year** benchmark bond yield ~~has~~ fallen from 7.17% to its current 6.02%.

[10] Numerical expressions ("20.3%", "2253.88 points") and intervals ("from 21% to 50%") could also be recognised by a local grammar, and processed as Atomic Linguistic Units.

Appendix 3: A Text Annotation Structure

Right after the lookup of NooJ's dictionaries and morphological grammars, the text annotation structure stores all lexical hypotheses and the corresponding ambiguities, such as the ones between "round" followed by "table" and "round table". Discontinuous linguistic units, such as "to take ... into account", are also represented.



```
Language is "English (United States)(en)".
Text Delimiter is: \n (NEWLINE)
Text contains 2 Text Units (TUs).
Text contains 37 characters.
7 tokens including:
7 word forms
```

He took the round table into account

An Efficient Denotational Semantics for Natural Language Database Queries

Richard A. Frost and Randy J. Fortier

University of Windsor, Canada

Abstract. Early work on natural language database query processing focused on theories of compositional semantics. Recent work concentrates on the translation of NL queries to SQL where semantics is primarily used in an ad hoc manner to guide syntactic translation. Here, we argue that there remains a need for an efficiently-implementable denotational semantics for NL DB queries, and show how this can be achieved by integrating a relatively little-known semantics for transitive verbs with a new efficiently-implementable semantics for negation.

1 Introduction

1.1 Why Do We Need a Denotational Semantics?

Researchers have studied NL DB query processing for over forty years. Androutsopoulos et al (1995) survey early work. Since then, there has been the annual NLDB conference. A review of this and other literature, shows that a) although early work focused on formal compositional semantics, none of the proposed semantics has gained widespread use, and b) much recent work has focused on translating NL queries to SQL, rather than attempting to directly interpret the queries with respect to a formal denotational semantics. Although remarkable achievements have been made in this endeavor, some shortcomings remain:

1) Hsu and Parker (1995) argue that it is not easy to express queries containing generalized quantifiers (GQ) in SQL. In particular it is difficult to create efficient SQL expressions for such queries. An example GQ query is "*Find all patient-disease pairs such that patient* `p` *has [some | all | no | not all] symptoms associated with disease* `d`". They propose an extended SQL, and a translator from their extended SQL to an optimized expression in conventional SQL. As such, they do not claim that SQL is unable to accommodate GQ queries, but that efficient formulations cannot be easily expressed.

2) Rao et al (1996) extend the work of Hsu and Parker and show that even the optimal translation of "*no*" and "*not all*" types of GQ queries to SQL results in "abysmal performance results". They do not criticize Hsu and Parker, but identify the reason for poor performance as resulting from the hidden complementation in such queries. They continue by showing why SQL and conventional database data structures are ill-equipped to deal with such queries.

3) As a corollary of 2) above, we claim that there is no efficient compositional and orthogonal method for translating quantified queries to SQL. A method

Z. Kedad et al.(Eds.): NLDB 2007, LNCS 4592, pp. 12–24, 2007.

is "compositional" if the meaning of a composite query is computed from the meanings of its parts using a small set of rules. A method is "orthogonal" if components have meanings that are independent of context, and a change to a component of a query only affects the meaning of the whole query in a well-defined manner. To illustrate the problem, suppose that we have an `OrbitsRelation` containing the tuples (Phobos, Mars), (Mars, Sol) etc., but not (Sol, anything). Consider the queries q1 = *"List all things that orbit a planet"* and q2 = "*List all things that orbit no planet"*. SQL translations are:

```
q1 = SELECT SubjectName FROM OrbitsRelation
     WHERE ObjectName IN (SELECT Name FROM Planets)
q2 = SELECT SubjectName FROM OrbitsRelation
     WHERE ObjectName NOT IN (SELECT Name FROM Planets)
```

Now consider the queries q3 = "*Does Sol orbit a planet*" and q4 = "*Does Mars orbit no planet"*. A compositional approach would include the relations returned by q1 and q2 respectively, and the resulting answers would be False for q3 and True for q4, which are correct. However, a compositional translation of the query q5 = "*Does Sol orbit no planet*", which should make use of the relation returned by q2, would return the incorrect answer due to the fact that Sol does not appear as subject in the orbits relation. To obtain the correct answer to queries of the form "*Does X REL no Y*", we need an SQL expression which is quite different in structure from the SQL expression for queries "*Does X REL a Y*", and which, according to Rao et al, would be highly inefficient. Because these two types of query have the same syntactic structure, a compositional and orthogonal method should translate both to SQL expressions with similar structure. Such a method would result in inefficient expressions for both types of query. In a sense, there is an impedance mismatch between NL and SQL.

4) Existing NL to SQL translators continue to be error-prone and unable to answer queries that one would reasonably expect. Bhootra (2004) conducted an experiment with Microsoft's English Query (EQ) and Access English Language Front End (ELF) (www.elfsoft.com). A set of questions was created which could reasonably be asked of the Northwind sample database that was shipped with MS SQL. The questions were used to create interfaces to the database using tools provided with EQ and ELF. The interfaces returned 42% (EQ) and 19% (ELF) incorrect answers for the set of questions that were used to generate them. The lack of accuracy of NL to SQL translators has also been identified by Popescu et al (2003) who introduce a new approach, together with a system called PRECISE, for constructing reliable NL DB query interfaces. The method involves matching words in the query to attributes in the database and a subsequent syntactic translation of the NL query to an SQL expression. Although their approach addresses the problem of accuracy, it does not address the problem of generalized quantifiers, nor does it appear to be possible to extend the approach to cover modal or intensional databases (see next).

5) SQL cannot express queries with modal or intensional constructs, such as *"has it always been the case that . . ."* and "*Does Paul believe that ...*". These

limitations are being addressed, by others, who have proposed to extend SQL or to create new SQL-like query languages.

We conclude that there remains a need for an efficiently-implementable denotational semantics for NL DB queries: a) to accommodate queries containing generalized quantification, b) to extend the scope of query processors beyond the first-order expressibility of SQL, and 3) to bridge the gap between Linguists who want to explain natural language, and Computer Scientists who want to build efficient NL interfaces.

In the following, we present an efficient semantics for a subset of NL DB queries. The semantics is a denotational semantics in that all denotations are functions (some of which are constant valued functions) and that the denotations of compound expressions are computed from the denotations of their components using function application and function composition only, according to their syntactic structure.

1.2 An Overview of the Proposed Approach

Work on the semantics of natural language queries falls into three categories:

1) Work, usually carried out by Computer Scientists, who develop new ad hoc but efficient semantics that are not based on established linguistic theories. Examples are described in the survey by Androutsopoulos et al (1995). More recent work includes Owei (2003), Popescu et al (2003), Duesterhoeft and Thalheim (2004), Little et al (2004), Tseng and Fan (2005), and Boonjing and Hsu (2006).

2) Work based on well-defined formal theories, usually carried out by Linguists who are not particularly concerned with efficiency. An example is Nelken and Francez (2002) which contains reference to many other similar papers.

3) Work which involves the modification and/or extension of established linguistic theories for use in database query processing. Most of this is based on Montague-like Semantics (Montague 1974). Examples are: Main and Benson (1983), Clifford and Warren (1983), Clifford (1990), Lapalme and Lavier (1993), Frost and Boulos (2002), Lee and Park (2002), and Cimiano et al (2007).

Our approach falls into category 3). We add a little-known explicit semantics for transitive verbs to Montague Semantics. The effect is similar to that proposed by Main and Benson (1983), and later by Clifford (1990). We convert the extended semantics to a more efficiently-implementable form based on sets and relations rather than characteristic functions. Finally, we add a relatively-new efficient semantics for negation proposed by Frost and Boulos (2002).

Our approach differs from most of the approaches in category 3 (except for Frost and Boulos) - we define denotations directly in terms of database relations, whereas all other researchers have used Montague's intensional logic (IL) as an intermediate representation. None have been able to extend their approach to handle arbitrary negation, and none have been able to directly generate efficiently-implementable denotations of queries without subsequent post-processing of those denotations. Montague stated that IL was dispensable, and we believe that it should dispensed with early in the process in order to obtain an efficient and compositional denotational semantics. We discuss, in section 4, what we have achieved, and what more needs to be done.

2 Montague Semantics (MS) and Its Shortcomings

The following is a brief introduction to some of Montague's ideas. In particular, we will say little here about modal or intensional aspects of natural language as these topics require substantial background discussion which can be found in, for example, (Montague 1974), Partee (1975), and Dowty, Wall and Peters (1981).

Montague claimed that natural language can be described in terms of a formal syntax and an associated compositional semantics. The relationship between syntax and semantics is similar to that in the denotational-semantics approach to the formal specification of programming languages, with the exception that expressions of a natural language have first to be "disambiguated" before interpretation. Such disambiguation involves mapping natural-language expressions to one or more unambiguous expressions in a syntactic algebra. These expressions are then mapped to semantic expressions through a homomorphism.

In MS, each disambiguated syntactic expression of English denotes a function in a function space constructed over a set of entities, the Boolean values `True` and `False`, and a set of states, each of which is a pair consisting of a "possible world" and a point in time. The functions are defined using the notation of lambda calculus. Each syntactic category is associated with a single semantic type. Each syntax rule is associated with a semantic rule which shows how the meanings of composite expressions are computed from the meanings of their constituents. The primary rule for syntactic composition is juxtaposition. The primary rule for semantic composition is function application.

Ignoring intensional aspects, common nouns such as "*planet*" and intransitive verbs such as "*spins*" denote predicates over the set of entities, i.e. characteristic functions of type `entity` → `bool`, where `x` → `y` denotes the type of functions whose input is a value of type `x` and whose output is of type `y`.

One of Montague's insights is that proper nouns do not denote entities directly. Rather, they denote functions defined in terms of entities. For example, the proper noun "*Phobos*" denotes the function λ`p p Phobos` where `Phobos` represents the entity Phobos. (For readers not familiar with the lambda calculus, the expression λ`x e` denotes a function which, when applied to an argument `y`, returns as result the expression `e` with all instances of `x` in it replaced by `y`.)

According to the rules proposed by Montague, the phrase "*Phobos spins*" is interpreted as follows, where `a => b` indicates that `b` is the result of evaluating `a`. Note that in this paper the denotation of a word is indicated by non-italic monospaced font. For example, `spins` is shorthand for the denotation of the word "spins", which according to Montague is a unary predicate:

```
(λp p Phobos) spins => spins Phobos
```

Quantifiers such as "*every*", and "*a*" denote higher-order functions of type `(entity→bool)→((entity→bool)→bool)`, e.g. the quantifier "*every*" denotes the function $\lambda p \lambda q\ \forall x\ (p\ x) \rightarrow (q\ x)$, where $\rightarrow$ is overloaded to denote logical implication here. Accordingly, the phrase "*every planet spins*" is interpreted as:

```
(λpλq ∀x p(x) → q(x)) planet spins
      => (λq ∀x planet(x) → q(x)) spins
      => ∀x planet(x) → spins(x)
```

Constructs of the same syntactic category denote functions of the same semantic type, e.g. the phrases "*Mars*" and "*every planet*" are both deemed to denote functions of type $(\texttt{entity} \rightarrow \texttt{bool}) \rightarrow \texttt{bool}$.

The resulting approach is highly orthogonal: many words that appear in differing syntactic contexts denote a single polymorphic function thereby avoiding the need to assign different meanings in these different contexts. For example, the word "*and*", which can be used to conjoin nouns, verbs, term-phrases, etc., denotes the polymorphic function $\lambda g \lambda f \lambda x\ (g\ x)\ \&\ (f\ x)$. For example, the phrase "*Phobos and Deimos spin"* is interpreted as shown below (where identifiers representing entities begin with a capital letter):

```
=> ((λgλfλx (g x) & (f x))(λp p Phobos) (λp p Deimos)) spin
=> (λx ((λp p Phobos) x) & ((λp p Deimos) x) ) spin
=> ((λp p Phobos) spin) & ((λp p Deimos) spin)
=> (spin Phobos) & (spin Deimos)
```

Montague Semantics has a number of shortcomings when used as a basis for the interpretation of NL DB queries: a) it is not fully compositional as it does not provide a direct denotation for transitive verbs. Instead, MS uses a convoluted syntactic process involving "relational notation" and a "delta *" operator (see page 216 in Dowty et al 1981, for details), b) MS cannot accommodate queries such as "*Does Phobos orbit and Luna orbit Mars.*" due to the fact that the phrases "*Phobos orbit*" and "*Luna orbit*" cannot be given straightforward denotations using function application, as the input types of the denotations of "*Phobos*" and "*Luna*" are incompatible with the type of the denotation of "*orbit*", c) direct implementation of MS is computationally intractable. Phrases such as "*every planet spins*" require all entities in the universe of discourse to be examined (see the interpretation of this phrase given earlier), and d) Montague gave no details of how negation should be accommodated w.r.t. the closed world assumption. We address these issues in the next section.

3 The Proposed Approach

3.1 A Little-Known Semantics for Transitive Verbs

It is possible to give a direct denotation for transitive verbs thereby avoiding a convoluted manipulation of an intermediate representation. We begin by noting that although Montague defined the denotation of proper nouns as, for example, $\lambda p\ p\ \texttt{Phobos}$, he viewed such denotations as being of type $(\texttt{entity} \rightarrow \texttt{bool})$

→ bool. This creates a difficulty when, attempting to define a denotation for transitive verbs, e.g. the denotation of "*discover*" would have to be of type:

((entity → bool) → bool) → (entity → bool) so that the denotation of "*discovered Phobos*" would be of the correct type for input to the denotation of "*Hall*". This does not appear to be possible.

The solution follows from the fact that the type of denotations such as λp p Phobos is more polymorphic than Montague stated. It is of type (entity → *) → * where * can be any type. This allows us to define the denotations of transitive verbs directly as follows : discover = λz z(λxλy disc_pred(y,x))

This is similar to, but not the same as, that proposed by Main and Benson (1983) and Clifford (1990). The following uses this denotation. The polymorphic type of λq q Phobos allows the lambda conversion at step 3:

```
         Hall              ( discovered            Phobos )
(λp p Hall) ((λz z(λxλy disc_pred(y,x))) (λq q Phobos))
=> (λp p Hall) ((λq q Phobos)(λxλy disc_pred(y,x)))
=> (λp p Hall) ((λxλy disc_pred(y,x)) Phobos)
=> (λp p Hall) (λy disc_pred(y,Phobos))
=> (λy disc_pred(y,Phobos)) Hall
=> disc_pred(Hall,Phobos)
```

Barbara Partee (personal communication) has pointed out that the above approach is not standard in linguistics, but that Dr. Kratzer, at the University of Massachusetts Amherst, has done something similar (Kratzer 2003), and Hendricks (1993) has proposed type-lifting to achieve a similar result in a more powerful semantic theory. Discussion of polymorphic types for transitive verbs and termphrases occurred over twenty years ago (e.g. Partee and Rooth 1983). Also, it is most likely that Montague was aware of the polymorphic type of termphrases, but chose to fix the type for linguistic reasons as it can be argued that the simpler type is more linguistically plausible. In addition, similar formalizations have been proposed in the context of logic programming, e.g. Blackburn and Bos (2005) who attribute it to Robin Cooper at Goteborg University.

The second problem of not being able to provide a compositional semantics for phrases such as "*Phobos orbits*" can be easily solved by extending MS to allow denotations to be created through function composition as well as function application. For example, the denotation of "*Phobos orbits*" is phobos . orbit. Thus, queries such as "*Does Phobos orbit and Deimos orbit Mars?*" are now interpreted as shown below using the denotation of "*and*" given earlier:

```
((phobos.orbits) and (luna.orbits)) mars
=>((phobos.orbits)mars) & ((luna.orbits)mars)
=>(phobos(orbits mars)) & (luna(orbits mars))
```

This approach accommodates a wide range of queries such as "*Did Hall discover and is Mars orbited by Phobos*", etc.

3.2 An Efficient Set-Theoretic Version of Montague Semantics

The computational intractability of MS results from denotations such as ∀x planet(x) → spins(x) which require that characteristic functions of sets (the denotations of common nouns and intransitive verbs) be applied to all entities in the universe of discourse. Other researchers, referred to earlier, who have based their semantics on Montague's approach, have not addressed this problem. Our solution is to use the sets themselves as denotations, rather than their characteristic functions, and convert all other denotations appropriately, e.g.

```
spins  = {Mars, Jupiter, Phobos, ..     moon = {Phobos, Deimos, ..
planet = {Mars, Jupiter, Mercury, ..    antibiotic = [Penicillin, ..
person = {Hall, Kuiper, Kowal, Fleming ..
```

The denotations of proper nouns are defined in terms of set membership, e.g.:

```
phobos = λp Phobos ∈ p
```

Quantifiers are defined in terms of set operators, e.g.:

```
every = λsλt s ⊆ t        a = λsλt s ∩ t  = {}
```

Conversion of the denotation of "*discover*" to a set-theoretic version gives:

```
discover = λq {x |(x,image_x) ∈ collect disc_rel & q image_x }
     where disc_rel = {(Hall,Phobos), (Hall,Deimos),
                       (Kuiper,Nereid),(Fleming, Penicillin), etc.}
```

In the above, the collect function is defined such that it returns a new binary-relation containing one binary tuple (x, image_x) for each member of the projection of the left-hand-column of disc_rel, where image_x is the image of x under the relation disc_rel, e.g. collect disc_rel => {(Hall, {Phobos,Deimos}),etc. An example application of discover is:

```
  discover phobos =>
λq {x |(x,image_x) ∈ collect disc_rel & q image_x }(λp Phobos ∈ p)
     => {x |(x,image_x) ∈ collect disc_rel & (λp Phobos ∈ p) image_x }
     => {x |(x,image_x) ∈ collect disc_rel & (Phobos ∈ image_x)}
     => {Hall}
```

One disadvantage of converting MS to a set-theoretic form is some loss of orthogonality. For example, the word "*and*" now needs more than one denotation:

```
term_and    = λpλqλs (p s) & (q s)        noun_and = λsλt s ∩ t
transvb_and = λpλqλr (p r) ∩ (q r)
```

Even with this slight loss of orthogonality, the mini-semantics is highly compositional: a) denotations are created using function application and function composition according to the syntactic structure of the query. For example, the query "*Did Kuiper discover and Hall discover a moon?*" is evaluated as:

```
(term_and (kuiper . discover) (hall . discover))(a moon)
```

and, b) words and phrases of the same syntactic category (e.g. "*a moon*" and "*Phobos*") denote semantic values of the same type as required by Montague.

It should be noted that syntactic ambiguity is accommodated by having the parser generate more than one disambiguated form. Semantic ambiguity is resolved, in a linguistically simple, yet efficient way, by assuming a right–to–left distribution. For example, according to our semantics, the query *"Did Kuiper discover and Hall discover a moon"* would be rewritten to:

```
(Kuiper (discover (a moon))) & (hall (discover (a moon)))
```

To obtain the answer to the other reading of this query, it would have to be restated as "*Was a moon discovered by Kuiper and Hall*". (The denotations of passive forms of verbs are defined using the inverse of the associated relations).

3.3 Accommodating Negation

Linguists have studied negation extensively (e.g. Iwanska 1992). However, no efficiently-implementable compositional semantics for accommodating arbitrary negation in NL DB queries exists. The problem can be illustrated by considering the following queries with respect to the `disc_rel` relation given earlier: "*Did Fleming discover no moon?*" and "*Did Lewis discover no moon?*". Most compositional semantic theories will return the correct answer for the first query but the wrong answer to the second query (with respect to the closed-world assumption which is appropriate for many database applications). This is because `disc_rel` does not contain `Lewis` in its left-hand column owing to the fact that Lewis did not discover anything according to this database. One solution to this problem is to extend the discover relation to include `(x, ''nothing'')` for all entities `x` in the domain of discourse which do not already occur in the left-hand column. This is clearly impractical for all but very small databases, and is useless for databases with infinite domains. A more practical solution, proposed by Frost and Boulos (2002), is based on the notion that a set can be represented in two ways: explicitly by enumerating its members, or implicitly by enumerating the members of its complement. When a set is computed as the denotation of a phrase that involves negation, it is represented as a complement. The set operators are redefined to take complements as operands.

We now show how our approach to transitive verbs can be integrated with the above approach to negation. We begin by introducing two "constructors" `SET` and `COMP` to distinguish between sets defined in the usual way, and those which are defined by enumerating the elements of their complement, e.g.

```
 SET  {Phobos,Deimos,etc.} denotes the moons
COMP {Phobos,Deimos,etc.} denotes the non moons
```

Operations on sets and complements are defined as follows:

```
c_member e  (SET  s) = e ∈ s
c_member e  (COMP s) = not (e ∈ s)
c_union     (SET  s) (SET  t) = SET  (s ∪ t)
c_union     (SET  s) (COMP t) = COMP (t -- s)
c_union     (COMP s) (SET  t) = COMP (s -- t)
c_union     (COMP s) (COMP t) = COMP (s ∩ t)
c_intersect (SET  s) (SET  t) = SET  (s ∩ t)
c_intersect (SET  s) (COMP t) = SET  (s -- t)
c_intersect (COMP s) (SET  t) = SET  (t -- s)
c_intersect (COMP s) (COMP t) = COMP (s ∪ t)
c_subset    (SET  s) (SET  t) = s ⊆ t
c_subset    (SET  s) (COMP t) = (t -- s) = t
c_subset    (COMP s) (SET  t) = (all_entities -- s) ⊆ t
c_subset    (COMP s) (COMP t) = t ⊆ s
```

where -- is set difference, `all_entities` denotes the set of all entities in the universe of discourse, and the definition of the set-cardinality operator # is extended as follows:

```
#(SET s) = #s      and #(COMP s)= #all_entities -- #(SET s)
```

In only one line, in the definition of `c_subset`, is it necessary to refer to the set of all entities. This is where we need to determine if a set represented by an enumeration of the members of its complement is a subset of a set that is represented by an explicit enumeration of it members (this computation occurs in the evaluation of the denotation of queries such as "*Does every thing that is orbited by no moon spin?*"). Fortunately, this part of the definition can be replaced by the following which refers only to the size of the set of all entities and not to the entities themselves:

```
c_subset (COMP s)(SET t) = ( #(s ∪ t) = #all_entities)
```

Redefinition of the denotations of most words is straightforward:

```
moon       = SET {Deimos, Phobos, etc.}    planet = SET {Jupiter,Mars}
spins      = SET {Jupiter,Mars,Phobos,etc} thing  = COMP {}
antibiotic = SET {Penicillin, etc.}
deimos     = λs c_member Deimos s           mars = λs c_member Mars s
a          = λsλt #(c_intersect s t) > 0   every = λsλt c_subset s t
no         = λsλt #(c_intersect s t) = 0    non  = λs COMP s
not        = s COMP s etc.
```

Evaluation of the denotation of the phrase "*non moon that spins*" would result in the following (assuming that the denotation of *that* is set to `noun_and`):

```
c_intersect (COMP{Phobos,Deimos, etc.})(SET {Jupiter,Mars,Phobos,etc})
     =>SET {Mars, Jupiter, etc.}
```

The problem with negation in queries such as *"Did Lewis discover no moon?"* is now solved by redefining the denotation of each transitive verb so that the function begins by applying the predicate given as argument to `SET{}` (representing the empty image of all entities that do not appear on the left-hand side of the associated relation), otherwise the result is returned in the form of a complement. If the predicate fails, the result returned is the same as that returned by the original denotation of the verb.

According to this approach, the new denotation of " `discover`" is:

```
discover = λq COMP({a |(a,b) ∈ disc_rel} -- result),if q(SET{}) = True
        SET result, otherwise
        where result = {x |(x,image_x) ∈ collect disc_rel & q image_x }
```

Now, the interpretation of "*discovered no moon*" returns the correct answer w.r.t. the `disc_rel` and *"Did Lewis discover no moon?"* is interpreted as:

```
=> (λp c_member Lewis p) (COMP ({a|(a,b) ∈ disc_rel} -- result))
        where
        result = {x|(x,image_x) ∈ collect disc_rel & (no moon) image_x}
=> (λp c_member Lewis p) (COMP ({Hall, Kuiper, Kowal, Fleming, etc}
                              -- {Fleming,etc})
=> (λp c_member Lewis p) (COMP {Hall, Kuiper, Kowal, etc.})
=> c_member Lewis (COMP {Hall, Kuiper, Kowal, etc.})
=> True
```

3.4 A Note on Compositionality, Efficiency, and Orthogonality

The semantics is highly compositional in that each word (after syntactic disambiguation) has a single denotation (meaning), and the meaning of composite queries is computed by function application and function composition only, in an order that is determined by the syntactic structure of the query. In addition, syntactic subcomponets of a query have meanings that can be computed independently of the whole query. Compositionality can be proven formally by induction on the length of expressions that can be derived from the context-free grammar (CFG) of the query language. The base case states that basic terms (words) of the query language have denotations of the required semantic type for their syntactic category (this follows directly from the semantic definitions that we have presented). The inductive step shows that the denotations of compound expressions that are created though use of any CFG rule are of the correct semantic type for the syntactic category, under the assumption that this is true for their components (this can be shown by considering each rule separately, and showing that for each alternative the computed denotation has the correct type. This also follows directly from our semantic definitions). In addition, the semantics overcomes the problem of "hidden complementation" discussed in section 1.1 item 2. The quantifier "*no*" can be treated in the same way semantically as "*a*" and "*every*" without incurring the inefficiency (and in some cases intractability) that would occur if complements of unary relations were enumerated explicitly.

As a consequence of the compositionality and the efficient treatment of negation, our semantics is highly orthogonal: a) the meaning of words and phrases within a query is, in most cases (with the exception of "*and*", independent of their context, b) if the meaning of a word is changed, the effect is propagated through the evaluation process in a well-defined way, and c) words and phrases of the same syntactic category, such as "*a*" and "*no*" can be interchanged without affecting the efficiency of the semantic evaluation process.

3.5 An Implementation and Evaluation

We have implemented the semantics directly in Miranda, a higher-order functional programming language. Example results are:

```
every (thing $that (orbits (no moon))) (orbits (no planet))    => False
a (non moon) (orbits sol)                                      => True
every moon (orbits (no moon))                                  => True
sol (orbits (a (non moon)))                                    => False
not (every moon) (is_orbited_by phobos)                        => True
a (moon $that (was_discovered_by hall))(does(not(orbit earth)))=> True
moon $that (was_discovered_by hall) => SET  [Phobos, Deimos]
orbits (no planet)              => COMP [Phobos, Deimos, Nereid, etc.]
non moon                        => COMP [Phobos, Deimos, Nereid, etc.]
discovered(a(moon $that (does(not(orbit (mars $term_or uranus)))))
                                => SET [Kowal, etc.]
```

To determine the viability of the approach, we have integrated the semantics with a functional parser using techniques described in Frost (2006), and have deployed the resulting application in a Public-Domain SpeechWeb (Frost 2005) which provides a speech interface. Details of how to access the speech interface from a PC through the Opera web browser are described in (Frost and Fortier 2007). A video demonstration is available at:

http://www.cs.uwindsor.ca/˜speechweb/movie.mov

4 Concluding Comments

We have identified a need for an efficiently-implementable denotational semantics for NL DB queries. We have integrated an explicit little-known denotation for transitive verbs with Montague Semantics. Then, unlike others, we have transformed the extended semantics to a computationally-tractable form by replacing characteristic functions of sets by the sets themselves and modifying all denotations accordingly. We then added a relatively-new semantics for negation. The resulting approach has three advantages compared to others: a) all forms of generalized quantification are accommodated efficiently in a compositional and orthogonal way, b) the explicit denotation for transitive verbs improves compositionality, and c) by defining the denotations of words, phrases, and queries directly in terms of database relations, we avoid the impedance mismatch that occurs between NL and SQL as discussed in section 1.1.

The approach accommodates queries containing common and proper nouns, transitive and intransitive verbs, conjunction, disjunction, and arbitrarily-nested quantification and negation. However, this is insufficient for many applications. We are currently extending the semantics to accommodate prepositional phrases as in *"Did Galileo discover Europa with a telescope?"*, indirect objects as in *"Did Galileo give Europa its name?"*, and aggregates as in *"What is the average mass* *of the planets?."* We will then begin the more-ambitious task of accommodating modal and intensional constructs. At that point we will compare our approach with that of others who are developing extended forms of SQL.

References

1. Androutsopoulos, I., Ritchie, G.D., Thanisch, P.: Natural Language Interfaces to Databases: An Introduction. J. of Lang. Engineering 1(1), 29–81 (1995)
2. Blackburn, P., Bos, J.: Representation and Inference for Natural Language. A First Course in Computational Semantics. CSLI Publications, Stanford, CA (2005)
3. Bhootra, R.: Natural Language Interfaces: Comparing English Language Front End and English Query. Master's Thesis, Virginia Commonwealth Univ (2004)
4. Boonjing, V., Hsu, C.: A New Feasible Natural Language Database Query Method. International Journal on Artificial Intelligence Tools 15(2), 323–330 (2006)
5. Cimiano, P., Haase, P., Heizemann, J.: Porting Natural Language Interfaces Between Domains — An Experimental User Study with the ORAKEL System. In: Proc. 12th Int. Conf. Intelligent User Interfaces IUI'07, pp. 180–189 (2007)
6. Clifford, J.: Formal Semantics and Pragmatics for Natural Language Querying. In: van Rijsbergen, C.J. (ed.) Cambridge Tracts in Theoretical Computer Science 8, Cambridge University Press, Cambridge (1990)
7. Clifford, J., Warren, D.S.: Formal Semantics for Time in Databases. ACM Trans. on Database Systems 8(2), 215–254 (1983)
8. Dowty, D.R., Wall, R.E., Peters, S.: Introduction to Montague Semantics. D. Reidel Publishing Company, Dordrecht, Boston, Lancaster, Tokyo (1981)
9. Duesterhoeft, A., Thalheim, B.: Linguistic based search facilities in snowflake-like database schemes. Data and Knowledge Engineering 48, 177–198 (2004)
10. Frost, R.A.: Realization of Natural-Language Interfaces using Lazy Functional Programming. ACM Comput. Surv. 38(4) Article No. 11 (2006)
11. Frost, R.A.: Call for a Public-Domain SpeechWeb. Commun. ACM 48(11), 45–49 (2005)
12. Frost, R.A., Boulos, P.: An efficient compositional semantics for natural-language database queries with arbitrarily-nested quantification and negation. In: Cohen, R., Spencer, B. (eds.) Advances in Artificial Intelligence. LNCS (LNAI), vol. 2338, pp. 252–267. Springer, Heidelberg (2002)
13. Frost, R.A., Ma, X., Shi, Y.: A Browser for a Public-Domain SpeechWeb. Accepted for WWW 2007 (2007)
14. Hendriks, H.: Studied flexibility: categories and types in syntax and semantics. Doctoral Thesis, Universiteit van Amsterdam (1993)
15. Hsu, P-Y., Parker, D.S.: Improving SQL with Generalized Quantifiers. In: Proc. 11th Int. Conf. on Data Engineering, pp. 298–305 (1995)
16. Iwanska, L.: A General Semantic Model of Negation in Natural Language: Representation and Inference. Doctoral Thesis, Computer Science, University of Illinois at Urbana-Champaign (1992)
17. Kratzer, A.: The event argument and the semantics of verbs, (2003), http://semanticsarchive.net/GU1NWM4Z
18. Lapalme, G., Lavier, F.: Using a functional language for parsing and semantic processing. Computational Intelligence 9, 111–131 (1993)
19. Lee, H., Park, J.C.: Interpretation of Natural Language Queries for Relational Databases with Combinatory categorial Grammar. Int. J. Comput. Proc. Oriental Lang 15(3), 281–303 (2002)

20. Little, J., de Ga, M., Ozyer, T., Alhajj, R.: Query Builder: A Natural Language Interface for Structured Databases. In: Aykanat, C., Dayar, T., Körpeoğlu, İ. (eds.) ISCIS 2004. LNCS, vol. 3280, pp. 470–479. Springer, Heidelberg (2004)
21. Main, M.G., Benson, D.B.: Denotational semantics for natural language question answering programs. American J. of Comput. Ling. 9(1), 11–21 (1983)
22. Montague, R.: In Formal Philosophy: Selected Papers of Richard Montague. Thomason, R.H. (ed.) Yale University Press, New Haven CT (1974)
23. Nelken, R., Francez, N.: Bilattices and the Semantics of Natural Language Questions. Linguistics and Philosophy 25(1), 37–64 (2002)
24. Owei, P.: Development of a Conceptual Query Language: Adopting the User-Centered Methodology. The. Computer Journal 46(6), 602–624 (2003)
25. Partee, B., Rooth, M.: Generalized conjunction and type ambiguity. In: Bauerle, R., Schwarze, C., von Stechow, A. (eds.) Meaning, Use and Interpretation of Language, pp. 361–383. Mouton de Gruyter, Berlin (1983)
26. Partee, B.H.: Montague Grammar and Transformational Grammar. Linguistic Inquiry 6(2), 203–300 (1975)
27. Popescu, A.M., Etzioni, O., Kautz, H.: Towards a Theory of Natural Language Interfaces to Databases, IUI'03 Miami, Florida, pp. 149–157 (2003)
28. Rao, S., Badia, A., Van Gucht, D.: Processing queries containing generalized quantifiers. Tech. Report TR 428. Computer Science, Indiana Univ. (1996)
29. Tseng, F.S.C., Fan, T.K.: Extending the Concepts of Object Role Modeling to Capture Natural Language Semantics for Database Access. In: Proc. IASTED Conference on Databases and Applications (2005)

An Approach to Hierarchical Email Categorization Based on ME

Peifeng Li, Jinhui Li, and Qiaoming Zhu

School of Computer Science & Technology, Soochow University, Suzhou, China, 215006
{pfli,jhli,qmzhu}@suda.edu.cn

Abstract. This paper proposes a hierarchical approach for categorizing emails with the ME (Maximum Entropy) model based on its contents and attributes. That approach categorizes emails in a two-phase way. First, it divides emails into two sets: legitimate set and Spam set; then it categorizes them in two different sets with different feature selection methods respectively. In addition, the pre-processing, the construction of features and the ME model suitable for the email categorization are also described in building the categorizer. Experimental results testify that our hierarchical approach is more efficient than existing approaches and the feature selection is an important factor that affects the precision of email categorization.

Keywords: Hierarchical categorization; ME model; Feature selection.

1 Introduction

Nowadays the email has become one of the most popular methods for people to communicate each other. Though the email gave us such timely convenience, it also caused the trouble of processing omnifarious emails. Classifying those emails into categories is a convenient and efficient way for people to read them.

A variety of approaches towards email categorization have been put forward in the past few years. Popular approaches to email categorization include RIPPER [1, 2], Rough Set [3], Rocchio [4, 5], Naïve Bayes [4, 6] SVM [6], Winnow [7], Neural network [8], etc. Those work proposed some useful approaches to email categorization. Nevertheless, most of above approaches were oriented from text categorization, so they classify emails using the plain text categorization approach, regardless of the differences between text and email. However, an email is a semi-structure text which includes a structure in the email head and it redounds to email categorization. Besides, the most popular approach used in email categorization is Bayes. That approach is poor on the precision though it is suitable for the requirement of the rapidity and dynamics in email categorization. Otherwise, mostly the SVM approach can get the highest precision in text categorization, but it doesn't satisfy the requirement of rapidity and dynamics because training the categorization model is expensive on time cost.

Z. Kedad et al.(Eds.): NLDB 2007, LNCS 4592, pp. 25–34, 2007.

Therefore, this paper introduces the ME (Maximum Entropy) model [9] into email categorization and proposes a hierarchical approach which categorizes the email based on its contents and attributes. This paper also discusses other techniques to improve the performance of categorizer, such as email pre-processing, features selection, iteration, etc.

2 To Pre-process the Email

After having analyzed the structure of an email, we divide it into two parts: contents and attributes. Contents include the email body and the subject which constitute the main part of an email. Attributes include those fields such as “From”, “Cc”, “To”, “Date”, “SMTP server”, “Attached files”, etc. The content part is mostly like a plain text while the attribute part is the characteristic of the email.

2.1 To Pre-process the Email Contents

The purpose of pre-processing email contents is to delete unused texts and to standardize them. The email format is different from plain text, and the additional pre-processing for email categorization is described as below:

(1) To filter the HTML tag in the email body

Generally the email body has two styles: plain text and html. If the email is in the html format, it should be converted to plain text because most html markups would confuse the categorizer except “<a href>” and “<img>”.

In this step, we firstly record the links (<a href>) and images (<img>) which is embedded in the html markups, and then delete all html markups and convert the html file to a plain text file.

(2) To filter the non-character symbol

In many emails, especially in spam emails, there are many non-character symbols, such as “☺”, “►”, “♫”, etc. Those symbols themselves are useless for categorizer so it’s necessary to delete them. In other way, non-character symbols usually are the characteristic of spam emails, so the number of non-character symbols also should be recorded as an important feature of the email for the categorizer.

(3) To unify the format of digitals

There are many digital formats, such as currency, date, time, phone number, etc. each one also has many different styles. For example, the telephone number “812-2345678” has many other styles, such as “(812) 234 5678”, “812-234-5678”, “812 2345678”, “812 234 5678”, etc. Therefore, we have defined a unified style for each digital format and all other format digitals should be transferred to that unified style.

(4) To revert the intersected words

To prevent spam flittering tools from labeling them as spams, many spam emails are inserted some marks between words or characters. Those marks must be deleted before starting the categorization. For example, the string “M*A*I*L” must be reverted to “MAIL”.

2.2 To Pre-process the Email Header

The email header consists of some important information, such as the name of sender/receiver, the email server, the IP address, etc. Those data are valuable for email categorization, especially for email filtering, so the fields of “From”, “To”, “Cc”, “Date”, “Content-Type”, etc should be extracted as features for the categorizer. For example, the content of the “From” is “Jason Lee <yoyo@aclweb.org>”, then the features could be extracted as follows: (in xml style)

<Sender>
<SenderName>Jason Lee</SenderName>
<SenderID>yoyo</SenderID>
< DomainName>aclweb.org</DomainName >
</Sender>

3 An Email Categorization Model

3.1 The ME Model

The ME model is one mature statistics model and it is suitable for email categorization. The basic theory of ME is that we should prefer to uniform models that also satisfy any given constraints, which are mined from the known event collection.

As for the ME model applied to text or email categorization, Zhang [10] provided an approach of filtering spam emails based on a ME model and Li [11] applied the ME model to classify the text. And there are no researches in public which are concerned with how to apply the ME model to categorize emails.

In email categorization, each email is deemed as an event. For example, there is an event collection which is presented as $\{(e_1,c_1),(e_2,c_2),(e_3,c_3),....,(e_N,c_N)\}$, where $e_i(1\le i\le N)$ denotes an email and $c_i(1\le i\le N)$ is the category of document e_i. To obtain the constraints from the event set, a feature function was introduced into ME model. The feature function in email categorization could be built on features and categories of emails. For the feature w and the category c', its feature function is:

$$f_{w,c'}(e,c)=\begin{cases}1 & c=c' \ \&\ e\,\text{contains}\,w\\ 0 & \text{otherwise}\end{cases} \tag{1}$$

The goal of email categorization is to obtain the best probability restricted by features, and the probability is defined as:

$$p_\lambda(c\,|\,e)=\tfrac{1}{Z_\lambda(e)}\exp(\sum_i\lambda_i f_i(e,c)) \tag{2}$$

where

$$Z_\lambda(e)=\sum_c\exp(\sum_i\lambda_i f_i(e,c)) \tag{3}$$

is simply a normalizing factor determined by the requirement of that $\sum_{c} p_{\lambda}(c \mid e) = 1$ for each document e, f_i is the feature function, λ_i is the weight assigned to feature f_i. Two algorithms specifically tailored to calculate the parameters of a ME classifier are Generalized Iterative Scaling Algorithm (GIS) and Improved Iterative Scaling Algorithm (IIS).

From our experiments, we found out that the performance of only using binary valued feature as formula (1) is inferior. So we optimize it and use word-frequency and word-position weight as feature's value. The new feature is defined as:

$$f_{w,c'}(e,c) = \begin{cases} \sum_{i=1}^{3} fq_i(w) * \lambda_i & c = c' \ \& \ e \ \text{contains} w \\ 0 & otherwise \end{cases} \tag{4}$$

where $fq_i(w)$ $(1 \le i \le 3)$ is the word-frequency weight and its value is related to the frequency of word *w* in body, subject and header of email *e*. λ_i $(1 \le i \le 3)$ is word-position weight and its value is defined as *1, 1.5* and *2* on our experiments.

3.2 To Extract the Features from the Emails

How to extract features from the email is very important for the categorizer. After the pre-processing, an email is expressed as a set of features actually. A categorizer should learn those features from the training set and then form the feature set for each category.

(1) In the training set, there are *k* pre-defined categories, noted as c_i $(1 \le i \le k)$ and in each category c_i there are a set of emails as $\{e_{i,1}, e_{i,2}, e_{i,3}, \ldots, e_{i,m}\}$ while each email $e_{i,j}$ consists of a set of features:

$$fe_{i,j} = \{f_{i,j,1}, f_{i,j,2}, f_{i,j,3}, \ldots, f_{i,j,n}\}$$

So the feature set of category c_i is defined as**:**

$$cfe_i = fe_{i,1} \cup fe_{i,2} \cup fe_{i,3} \cup \ldots \cup fe_{i,m} = \bigcup_{j=1}^{m} fe_{i,j} \tag{5}$$

(2) Then for each cfe_i, to delete the features which occurred in other $cfe_j\,(i<>j)$:

$$\begin{aligned} cfe_i &= cfe_i - cfe_i \cap (cfe_1 \cup cfe_2 \cup \ldots \cup cfe_{i-1} \cup cfe_{i+1} \cup \ldots \cup cfe_k) \\ &= cfe_i - cfe_i \cap (\bigcup_{1 \le j \le k, j \ne i} cfe_j) \end{aligned} \tag{6}$$

(3) The experimental results indicated that chi-square statistics was the best approach to extract feature in email categorization. So for each feature $f_{i,j}$ in category c_i, to calculate the chi-square statistic value between $f_{i,j}$ and c_i:

$$chi(f_{i,j}, c_i) = \frac{N[P(f_{i,j}, c_i) * P(\bar{f}_{i,j}, \bar{c}_i) - P(f_{i,j}, \bar{c}_i) * P(\bar{f}_{i,j}, c_i)]^2}{P(f_{i,j}) * P(\bar{f}_{i,j}) * P(c_i) * P(\bar{c}_i)} \tag{7}$$

where

$P(f_{i,j},c_i)$ is the probability of $f_{i,j}$ and c_i co-occur, $P(\bar{f}_{i,j},\bar{c}_i)$ is the probability of neither $f_{i,j}$ or c_i occur, $P(f_{i,j},\bar{c}_i)$ is the probability of $f_{i,j}$ occurs without c_i, $P(\bar{f}_{i,j},c_i)$ is the probability of c_i occurs without $f_{i,j}$.

(4) To sort all features on the chi-square statistic values, the result of category c_i is as follow:

$$fe_i=\{f_{i,1}, f_{i,2}, f_{i,3},\dots,f_{i,l}\} \tag{8}$$

while

$$chi(f_{i,j},c_i) \geq chi(f_{i,j+1},c_i) \qquad (1 \leq j \leq l-1)$$

Then delete all $fe_{i,j}$ while j>2000. This means that each category only reserves 2000 features.

4 Experiments and Analysis

4.1 The Email Corpus

Currently there are some public email corpora, such as Ling-spam, PU1, PU123A, Enron Corpus [12], etc. But all of above corpora couldn't be used to test our approach, because most of them are mainly used to filter spam emails and only have two categories: legitimate emails and spam emails. Besides, Enron Corpus has many categories, but it just categorizes the email by users, not by contents. Therefore, to test our approach, we have to build an email corpus which includes 11907 category-defined emails with 7 categories. And the categories are {Work & Study, Auto-reply, Private Contents, Advertisements, Invoice and Tax, Porn and Adult, Train and Lecture}. The first 3 categories are legitimate emails sets (Legi) while the others are spam emails sets (Spam). Otherwise, all emails in our corpus must satisfy one restriction: the number of words in the email body must be greater than 10. We choose 1/3 emails randomly from each category as the test set, and the rest regards as the training set. Table 1 shows the test set and we provide four combinations to extract the features from the email. Those combinations are as follows:

SB: Subject + Body	B: Body
HS: Header + Subject	HSB: Header + Subject + Body

Table 1. The test set for email categorization

Categories	Num	Categories	Num
work & study	2217	advertisements	848
auto-reply	82	invoice and tax	316
private contents	217	Porn and adult	69
		train and lecture	220

In our experiments we report Recall (R), Precision (P) and F-measure (F) for each category, and Micro-average P (or simply Micro-P) for each text categorization approach. These measures are defined as follows:

$$R(c) = \frac{the\ \ number\ \ of\ \ emails\ \ correctly\ \ assigned\ \ to\ \ class\ \ c}{the\ \ number\ \ of\ \ emails\ \ contained\ \ in\ \ class\ \ c} \tag{9}$$

$$P(c) = \frac{the\ \ number\ \ of\ \ emails\ \ correctly\ \ assigned\ \ to\ \ class\ \ c}{the\ \ number\ \ of\ \ emails\ \ assigned\ \ to\ \ class\ \ c} \tag{10}$$

$$F = \frac{R \times P \times (1+\beta^2)}{R + P \times \beta^2} \tag{11}$$

$$Micro\text{-}P = \frac{the\ \ number\ of\ \ correctly\ assigned\ test\ \ emails}{the\ \ number\ of\ \ test\ \ emails} \tag{12}$$

where β is the relative weight of the precision and the recall. We assign 1 to β, namely F-1 value.

4.2 Experiments on the Iteration and Feature Number

Firstly, we experiment our approach on the test set with different iterations from 50 to 550. Figure 1 shows the results.

From figure 1, we found out that every Micro-P increases rapidly as the increasing of the number of iterations from 50 to 250. But when the iteration number is greater than 250, each line goes steady and becomes a horizontal line. The greater the number of iteration is, the higher the time cost is. So we choose 250 as the iteration in our approach.

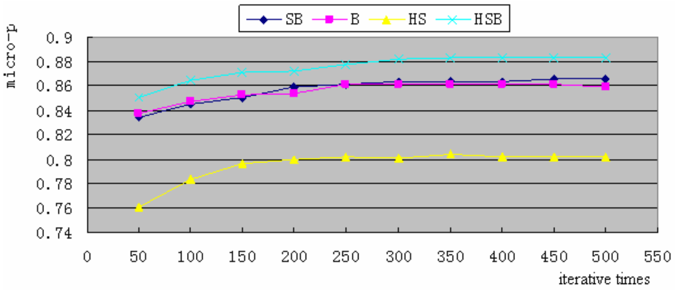


Fig. 1. The relation between the iteration and Micro-P (feature number: 2000)

We also tested our approach with different numbers of features from 500 to 5500. Figure 2 shows the results.

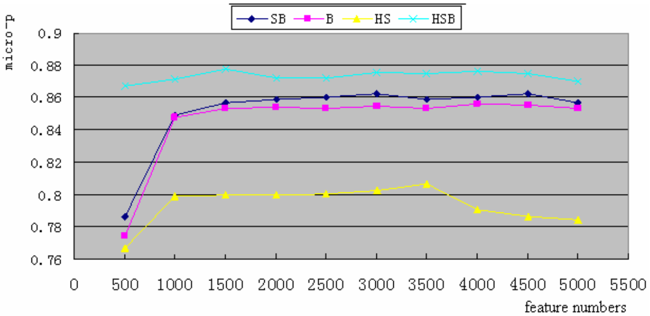


Fig. 2. The relation between the feature number and Micro-P

In figure 2, we found out that the Micro-P increases rapidly as the increasing of the feature number from 50 to 1500. And when the feature number increased from 3500 to 5500, the Micro-P didn't increase actually. On the contrary, it decreased with the increasing of the feature number sometime. So we choose 2000 as the feature number at last considering both the performance and time cost.

4.3 Experiment on Different Feature Combinations

We also tested our categorizer on four different feature combinations mentioned above. Table 2 shows the results.

In table 2, we found out that HSB was the best of all four combinations for it only had 286 emails incorrectly categorized. But for legitimate set, HS was the better one than the others and only had 111 emails incorrectly categorized.

From that result, we proposed a two-phrase categorization. This approach firstly divides emails into two sets: Legi and Spam while using HSB as features, and then categorizes emails in two different sets respectively on different feature combinations while using HS in Legi and HSB in Spam. We named this approach as hierarchical categorization approach while the original approach was called as direct categorization.

Table 2. The number of emails incorrectly categorized with four feature combinations

Combinations	HS	B	SB	HSB
Number of emails incorrectly categorized in Legi	**111**	214	142	145
Number of emails correctly categorized in Spam	327	189	171	**141**
Total number of emails incorrectly categorized	438	403	310	**286**
Micro-P	0.8896	0.8985	0.9219	**0.9279**

4.4 Experiments on Hierarchical Categorization

We tested the hierarchical categorization approach on our test set, and the results showed in table 3 and 4.

Table 3. The recall, precision, F-measure and Micro-P after the first categorization

Categories	R	P	F	Micro-P
Legi	0.9759	0.9807	0.9783	0.9725
Spam	0.9665	0.9585	0.9625	

Table 4. The recall, precision, F-measure and Micro-P after the second categorization in Legi set and spam set respectively

Categories	R	P	F
work & study	0.9838	0.9528	0.9681
auto-reply	0.8095	0.8947	0.8500
private contents	0.4561	0.9630	0.619
advertisements	0.9220	0.8935	0.9075
invoice and tax	0.9905	0.9811	0.9858
Porn and adult	0.7826	0.7826	0.7826
train and lecture	0.8631	0.8872	0.8750
Micro-P	0.9346		---

In table 3, it lists the recall, precision, F-measure and Micro-P after the first categorization. In this step, the categorizer divides the test set into two sets: Legi and Spam, so it mostly liked a spam email filter, but a categorizer. The Micro-P of the first categorization is 97.25%. This result also testified ME model utilizing word- frequency and word-position weight is an efficient way to filter spam emails.

In table 4, it lists the recall, precision and F-measure of all categories and the Micro-P after the second categorization in Legi set and spam set respectively. Hereinto, the categorizer classifies the Legi set based on HS while it classifies the spam set based on HSB. After two phrases, the final Micro-P is 93.46% in all.

From table 2, 3, and 4, we find out that hierarchical categorization is better than direct categorization. It achieved an improvement of Micro-P by 0.67% over the direct categorization. But hierarchical categorization is more complex than direct categorization, because it must categorize the test set twice while direct categorization only need once.

4.5 To Compare with Other Approaches

To compare our approach with other popular approaches, we also tested the Bayes, SVM, KNN and Winnow approaches on our email corpus. In those experiments, we extracted features from email body and subject by utilizing chi-square statistics (those experiments also indicated that chi-square statistics is better than Information Gain, Mutual Information, etc in email categorization) and the feature number was also selected as 2000. Figure 3 shows the result.

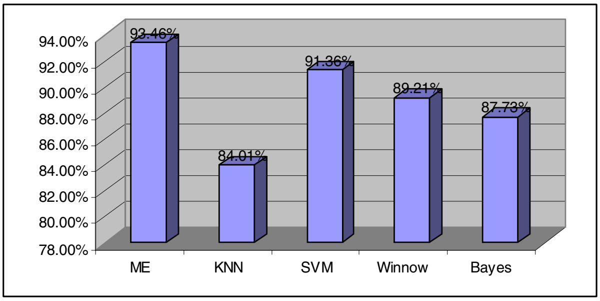


Fig. 3. The Micro-P of ME, KNN, SVM, Winnow and Bayes

In common, SVM is the best approach to categorize the text, but it needs too much time to train the model and is not suitable for email categorization because it is time costly sometimes. But in our experiment, the results testify that ME approach is the best one to classify emails, and it achieves an improvement of Micro-P by 2.1% over the SVM approach. We believe that the pre-processing, appropriate feature selection method and the hierarchical categorization approach are the main factors for ME to beat the SVM in our experiments.

4.6 Experiments Analysis

Based on above experiments, we also can find out that:

(1) To extract features from all fields of the email is the best way. Except the body, other fields also can provide useful information for categorizer to improve the performance.
(2) The hierarchical categorization is better than direct categorization, but it also is more complex and time-consuming than that one.
(3) Usually, features extracted from HSB are the best combination for email categorization, but for legitimate email, the HS is the best choice in our experiments. The reason is that: the subject of a legitimate email often can abstract the content while the spam email always give a fake subject in order to cheat the filtering program.
(4) In our email corpus, emails in the category “Work & Study” and “Private Contents” are easy to confuse because of their content also can’t be distinguished by people. So the recall of category “Private Contents” is very low in table 4 and many emails in such category are assigned to category “Work & Study” by our categorizer.

5 Conclusion

This paper proposes a hierarchical email categorization approach based on ME model and also discusses the pre-process, the feature selection and the iteration. We have implemented a categorizer which based on such a model and that categorizer is a plug-in component for the Microsoft Outlook. It is used by many users and the survey result indicates that it works well. Our future work mainly focuses on optimizing the ME model and adding machine learning algorithm to learn users' actions and then to adjust the model. Otherwise, our approach is really poor to classify that email when its body only has few words, especially it's empty. So we also plan to research on those emails and try to find a method to classify them correctly.

Acknowledgments. The authors would like to thank three anonymous reviewers for their comments on this paper. This research was supported by the National Natural Science Foundation of China under Grant No.60673041, Nation 863 Project of China under Grant No. 2006AA01Z147 and the High Technology Plan of Jiangsu Province, China under Grant No.2005020**.**

References

1. Cohen, W.: Learning rules that classify e-mail. In: Proc. of AAAI Spring Symposium on Machine Learning and Information Retrieval, pp. 18–25 (1996)
2. Provost, J.: Naïve-bayes vs. rule-learning in classification of email. Technical Report AITR-99-284, University of Texas at Austin, Artificial Intelligence Lab (1999)
3. Li, Z., Wang, G., Wu, Y.: An E-mail classification system based on Rough Set. Computer Science 31(3), 58–60 (2004)
4. Yang, J., Chalasani, V., Park, S.: Intelligent email categorization based on textual information and metadata. IEICE Transactions on Information and Systems, pp. 1280–1288 (2003)
5. Yang, J., Park, S.: Email categorization using fast machine learning algorithms. In: Proc. of the 5th Int. Conf. on Discovery Science, pp. 316–323 (2002)
6. Bekkerman, R., McCallum, A., Huang, G.: Automatic categorization of email into folders: benchmark experiments on Enron and SRI corpora. CIIR Technical Report IR418 (2004)
7. Zhu, Q., Zhou, Z., Li, P.: Design of the Chinese mail classifier based on Winnow. Acta Electronica Sinca 33(12A), 2481–2482 (2005)
8. Clark, J., Koprinska, I., Poon, J.: LINGER – a smart personal assistant for e-mail classification. In: Proc. Of the 13th Int. Conf. on Artificial Neural Networks, pp. 274–277 (2003)
9. Berger, A., Pietra, S., Pietra, V.: A maximum entropy approach to natural language processing. Computational Linguistics 22(1), 38–73 (1996)
10. Zhang, L., Yao, T.: Filtering junk mail with a maximum entropy model. In: Proc. of 20th Int. Conf. on Computer Processing of Oriental Languages, pp. 446–453 (2003)
11. Li, R., Wang, J., Chen, X., et al.: Using Maximum Entropy model for Chinese text categorization. Journal of Computer Research and Development 42(1), 94–101 (2005)
12. Klimt, B., Yang, Y.: The Enron Corpus: A new dataset for email classification research. In: Proc. of ECML'04, 15th European Conf. on Machine Learning, pp. 217–226 (2004)

Developing Methods and Heuristics with Low Time Complexities for Filtering Spam Messages

Tunga Güngör and Ali Çıltık

Boğaziçi University, Computer Engineering Department, Bebek,
34342 İstanbul, Turkey
gungort@boun.edu.tr, ali@ciltik.com

Abstract. In this paper, we propose methods and heuristics having high accuracies and low time complexities for filtering spam e-mails. The methods are based on the n-gram approach and a heuristics which is referred to as the first n-words heuristics is devised. Though the main concern of the research is studying the applicability of these methods on Turkish e-mails, they were also applied to English e-mails. A data set for both languages was compiled. Extensive tests were performed with different parameters. Success rates of about 97% for Turkish e-mails and above 98% for English e-mails were obtained. In addition, it has been shown that the time complexities can be reduced significantly without sacrificing from success.

Keywords: Spam e-mails, N-gram methods, Heuristics, Turkish.

1 Introduction

In parallel to the development of the Internet technology, the role of e-mail messages as a written communication medium is increasing from day to day. However, besides the increase in the number of legitimate (normal) e-mails, the number of spam e-mails also increases. Spam e-mails are those that are sent without the permission or interest of the recipients. According to a recent research, it was estimated that about 55 billion spam messages are sent each day [1]. This huge amount of e-mails cause waste of time and resources for the users and the systems, and have the potential of damaging the computer systems. Thus it is crucial to fight with spam messages.

The simplest solution to preventing spam e-mails is blocking messages that originate from sites known or likely to send spam. For this purpose, blacklists, whitelists, and throttling methods are implemented at the Internet Service Provider (ISP) level. Although these methods have the advantage of being economical in terms of bandwidth, they are static methods and cannot adapt themselves easily to new strategies of spammers. More sophisticated approaches rely on analyzing the content of e-mail messages and are usually based on machine learning techniques. Naïve Bayes Network algorithms were used frequently and they have shown a considerable success in filtering spam e-mails [2,3,4]. The effects of several factors on Bayesian filtering such as the size of the training corpus, lemmatization, and stop words have been investigated. Success rates around 96-98% were obtained for English e-mails.

Z. Kedad et al.(Eds.): NLDB 2007, LNCS 4592, pp. 35–47, 2007.

In [5], a rule-based approach was used against spam messages. Two methods for learning text classifiers were compared: a traditional information retrieval method and a method for learning sets of keyword-spotting rules. It was found that rule-based methods obtain significant generalizations from a small number of examples.

Since spam filtering can be considered as a text categorization task, support vector machines (SVMs) were also employed recently for predicting the classes spam and normal [6,7]. It was argued that SVMs outperform other learning methods under some conditions. In addition, using all the features (words) in the messages rather than restricting to a subset was found to result in better performance. This is due to the difficulty of determining an optimum set of features. In [8], case-based reasoning which is a lazy machine learning technique was applied to spam filtering. Different types of spam e-mail incorporate different concepts. Case-based classification works well for disjoint concepts whereas other techniques like Naïve Bayes tries to learn a unified concept description. Memory-based learning [9] and maximum entropy models [10] are among the other learning paradigms used in spam filtering.

Besides trying to apply machine learning methods to the spam problem, the research has also progressed in other directions. The solutions based on some protocols and standards form a different point of view to the problem. Authenticated SMTP (Simple Mail Transfer Protocol) and SPF (Sender Policy Framework) have been developed as tools that restrict the spammers dramatically. SPF has also increased the popularity of Authenticated SMTP [11,12]. Another solution proposed recently is using cost-based systems. Since spammers send huge amount of e-mails, requiring senders to pay some cost for each e-mail will make it prohibitively expensive for spammers. However, this idea is not mature yet and some issues like what to do when an infected computer of a user originates the spam messages need to be solved before putting it into practice.

In this paper, we propose an approach for spam filtering that yields high accuracy with low time complexities. The research in this paper is two-fold. First, we develop methods that work in much less time than the traditional methods in the literature. For this purpose, two novel methods are presented and some variations of each are considered. We show that, despite the simplicity of these methods, the success rates lie within an acceptable range. Second, in relation with the first goal, we develop a heuristics based on an observation about human behavior for spam filtering. It is obvious that humans do not read an incoming e-mail till the end of it in order to understand whether it is spam or not. Based on this fact, we form a heuristics, named as *first n-words heuristics*, which takes only the initial n words in the e-mail into account and discards the rest. The plausibility of the heuristics is tested with different n values. We find that similar performance can be achieved with small n values in addition to a significant decrease in time.

Though the approach proposed and the methods developed in this paper are general and can be applied to any language, our main concern is testing their effectiveness on Turkish language. To the best of our knowledge, the sole research for filtering Turkish spam e-mails is given in [13]. Two special features found in Turkish e-mails were handled in that research: complex morphological analysis of words and replacement of English characters that appear in messages with the corresponding correct Turkish characters. By using artificial neural networks (ANNs) and Naïve Bayes, a success rate of about 90% was achieved.

In the current research, we follow the same line of processing of Turkish e-mail messages and solve the problems that arise from the agglutinative nature of the language in a similar manner. Then by applying the aforementioned methods and the heuristics, we obtain a success rate of about 97% (and a lower time complexity), which indicates a substantial increase compared to [13]. In addition to Turkish messages, in order to be able to compare the results of the proposed approach with the results in the literature, we tested on English e-mails. The results reveal that up to 98.5% success rate is possible without the use of the heuristics and 97% success can be obtained when the heuristics is used. We thus conclude that great time savings are possible without decreasing the performance below an acceptable level.

2 Data Set

Since there is no data available for Turkish messages, a new data set has been compiled from the personal messages of one of the authors. English messages were collected in the same way. The initial size of the data set was about 8000 messages, of which 24% were spam. The data set was then refined by eliminating repeating messages, messages with empty contents (i.e. having subject only), and 'mixed-language' messages (i.e. Turkish messages including a substantial amount of English words/phrases and English messages including a substantial amount of Turkish words/phrases). Note that not taking repeating messages into account is a factor that affects the performance of the filter negatively, since discovering repeating patterns is an important discriminative clue for such algorithms. It is a common style of writing for Turkish people including both Turkish and English words in a message. An extreme example may be a message with the same content (e.g. an announcement) in both languages. Since the goal of this research is spam filtering for individual languages, such mixed-language messages were eliminated from the data set.

In order not to bias the performance ratios of algorithms in favor of spam or normal messages, a balanced data set was formed. To this effect, the number of spam and normal messages was kept the same by eliminating randomly some of the normal messages. Following this step, 640 messages were obtained for each of the four categories: Turkish spam messages, Turkish normal messages, English spam messages, and English normal messages.

In addition to studying the effects of spam filtering methods and heuristics, the effect of morphological analysis (MA) was also tested for Turkish e-mails (see Section 4). For this purpose, Turkish data set was processed by a morphological analyzer and the root forms of words were extracted. Thus three data sets were obtained, namely English data set (1280 English e-mails), Turkish data set without MA (1280 Turkish e-mails with surface forms of the words), and Turkish data set with MA (1280 Turkish e-mails with root forms of the words). Finally, from each of the three data sets, eight different data set sizes were formed: 160, 320, 480, 640, 800, 960, 1120, and 1280 e-mails, where each contains the same number of spam and normal e-mails (e.g. 80 spam and 80 normal e-mails in the set having 160 e-mails). This grouping was later used to observe the success rates with different sample sizes.

3 Methods and Heuristics

We aim at devising methods with low time complexities, without sacrificing from performance. The first attempt in this direction is forming simple and effective methods. Most of the techniques like Bayesian networks and ANNs work on a word basis. For instance, spam filters using Naïve Bayesian approach assume that the words are independent; they do not take the sequence and dependency of words into account. Assuming that w_i and w_j are two tokens in the lexicon, and w_i and w_j occur separately in spam e-mails, but occur together in normal e-mails, the string w_iw_j may lead to misclassification in the case of Bayesian approach. In this paper, on the other hand, the proposed classification methods involve dependency of the words as well.

The second attempt in this direction is exploiting the human behavior in spam perception. Whenever a new e-mail is received, we just read the initial parts of the message and then decide whether the incoming e-mail is spam or not. Especially in the spam case, nobody needs to read the e-mail till the end to conclude that it is spam; just a quick glance might be sufficient for our decision. This human behavior will form the base of the filtering approach presented in this paper. We simulate this human behavior by means of a heuristics, which is referred to as the *first n-words heuristics*. According to this heuristics, considering the first n words of an incoming e-mail and discarding the rest can yield the correct class.

3.1 Parsing Phase

In this phase, Turkish e-mails were processed in order to convert them into a suitable form. Then, the words were analyzed by morphological module, which extracted the roots. The root and surface forms were used separately by the methods.

One of the conversions employed was replacing all numeric tokens with a special symbol ("num"). This has the effect of reducing the dimensionality and mapping the objects belonging to the same class to the representative instance of that class. The tests have shown an increase in the success rates under this conversion. Another issue that must be dealt with arises from the differences between Turkish and English alphabets. Turkish alphabet contains special letters ('ç','ğ','ı','ö','ş','ü'). In Turkish e-mails, people frequently use 'English versions' of these letters ('c','g','i','o','s','u') to avoid from character mismatches between protocols. During preprocessing, these English letters were replaced with the corresponding Turkish letters. This is necessary to arrive at the correct Turkish word. This process has an ambiguity, since each of such English letters either may be the correct one or may need to be replaced. All possible combinations in each word were examined to determine the Turkish word.

We have used the PC-KIMMO tool in order to extract the root forms of the words, which is a morphological analyzer based on the two-level morphology paradigm and is suitable for agglutinative languages [14]. One point is worth mentioning. Given an input word, PC-KIMMO outputs all possible parses. Obviously, the correct parse can only be identified by a syntactic (and possibly semantic) analysis. In this research, the first output was simply accepted as the correct one and used in the algorithms. It is possible to choose the wrong root in this manner. Whenever the tool could not parse the input word (e.g. a misspelled word), the word itself was accepted as the root.

3.2 Perception Using N-Gram Methods

The goal of the perception phase is, given an incoming e-mail, to calculate the probability of being spam and the probability of being normal, namely P(spam|e-mail) and P(normal|e-mail). Let an e-mail be represented as a sequence of words in the form E=$w_1w_2\ldots w_n$. According to Bayes rule

$$P(\text{spam} \mid E) = \frac{P(E \mid \text{spam})\,P(\text{spam})}{P(E)} \tag{1}$$

and, similarly for P(normal|E). Assuming that P(spam)=P(normal) (which is the case here due to the same number of spam and normal e-mails), the problem reduces to the following two-class classification problem:

$$\text{Decide}\begin{cases}\text{spam} & \text{, if } P(E \mid \text{spam}) > P(E \mid \text{normal}) \\ \text{normal} & \text{, otherwise}\end{cases} . \tag{2}$$

One of the least sophisticated but most durable of the statistical models of any natural language is the n-gram model. This model makes the drastic assumption that only the previous n-1 words have an effect on the probability of the next word. While this is clearly false, as a simplifying assumption it often does a serviceable job. A common n is three (hence the term trigrams) [15]. This means that:

$$P(w_n \mid w_1, \ldots, w_{n\text{-}1}) = P(w_n \mid w_{n\text{-}2}, w_{n\text{-}1}) \; . \tag{3}$$

So the statistical language model becomes as follows (the right-hand side equality follows by assuming two hypothetical starting words used to simplify the equation):

$$P(w_{1,n}) = P(w_1)\,P(w_2 \mid w_1)\prod_{i=3}^{n} P(w_i \mid w_{i\text{-}2}, w_{i\text{-}1}) = \prod_{i=1}^{n} P(w_i \mid w_{i\text{-}2}, w_{i\text{-}1}) \; \cdot \tag{4}$$

Bayes formula enables us to compute the probabilities of word sequences ($w_1\ldots w_n$) given that the perception is spam or normal. In addition, n-gram model enables us to compute the probability of a word given previous words. Combining these and taking into account n-grams for which n≤3, we can arrive at the following equations (where C denotes the class spam or normal):

$$P(w_i \mid C) = \frac{\text{number of occurrences of } w_i \text{ in class C}}{\text{number of words in class C}} \tag{5}$$

$$P(w_i \mid w_{i\text{-}1}, C) = \frac{\text{number of occurrences of } w_{i\text{-}1}w_i \text{ in class C}}{\text{number of occurrences of } w_{i\text{-}1} \text{ in class C}} \tag{6}$$

$$P(w_i \mid w_{i\text{-}2}, w_{i\text{-}1}, C) = \frac{\text{number of occurrences of } w_{i\text{-}2}w_{i\text{-}1}w_i \text{ in class C}}{\text{number of occurrences of } w_{i\text{-}2}w_{i\text{-}1} \text{ in class C}} \; . \tag{7}$$

A common problem faced by statistical language models is the sparse data problem. To alleviate this problem, several smoothing techniques have been used in the literature [15,16]. In this paper, we form methods by taking the sparse data problem into account. To this effect, two methods based on equations (5)-(7) are proposed. The first one uses the following formulation:

$$P(C \mid E) = \prod_{i=1}^{n} \left[\mathrm{P}(\mathrm{w_i} \mid \mathrm{C}) + \mathrm{P}(\mathrm{w_i} \mid \mathrm{w_{i-1}}, \mathrm{C}) + \mathrm{P}(\mathrm{w_i} \mid \mathrm{w_{i-2}}, \mathrm{w_{i-1}}, \mathrm{C}) \right] \cdot \tag{8}$$

The unigram, bigram, and trigram probabilities are totaled for each word in the e-mail. In fact, this formula has a similar shape to the classical formula used in HMM-based spam filters. In the latter case, each n-gram on the right-hand side is multiplied by a factor λ_i, $1 \leq i \leq 3$, such that $\lambda_1+\lambda_2+\lambda_3=1$. Rather than assuming the factors as predefined, HMM is trained in order to obtain the values that maximize the likelihood of the training set. Training a HMM is a time consuming and resource intensive process in the case of high dimensionality (i.e. with large number of features (words), which is the case here). In spam filtering task, however, time is a critical factor and processing should be in real time. Thus we prefer a simpler model by giving equal weight to each factor.

The second model is based on the intuition that n-gram models perform better as n increases. In this way, more dependencies between words will be considered; a situation which is likely to increase the performance. The formula used is as follows:

$$P(C \mid E) = \prod_{i=1}^{n} (\eta_{\mathrm{i}}) \tag{9}$$

where

$$\eta_i = \begin{cases} \mathrm{P}(\mathrm{w_i} \mid \mathrm{w_{i-2}}, \mathrm{w_{i-1}}, \mathrm{C}), \text{if } \mathrm{P}(\mathrm{w_i} \mid \mathrm{w_{i-2}}, \mathrm{w_{i-1}}, \mathrm{C}) \neq 0 \\ \mathrm{P}(\mathrm{w_i} \mid \mathrm{w_{i-1}}, \mathrm{C}) \quad , \text{ if } \mathrm{P}(\mathrm{w_i} \mid \mathrm{w_{i-1}}, \mathrm{C}) \neq 0 \text{ and } \mathrm{P}(\mathrm{w_i} \mid \mathrm{w_{i-2}}, \mathrm{w_{i-1}}, \mathrm{C}) = 0 \\ \mathrm{P}(\mathrm{w_i} \mid \mathrm{C}) \quad , \text{otherwise} \end{cases} \tag{10}$$

As can be seen, trigram probabilities are favored when there is sufficient data in the training set. If this is not the case, bigram probabilities are used, and unigram probabilities are used only when no trigram and bigram can be found.

It is still possible that the unigram probabilities may evaluate to zero for some words in the test data, which has the undesirable effect of making the probabilities in (8) and (9) zero. The usual solution is ignoring such words. Besides this strategy, we also considered another one, which minimizes the effect of those words rather than ignoring them. This is achieved by replacing the zero unigram value with a very low value. Both of the methods mentioned above were applied with each of these variations (referred to as (a) and (b)), yielding a total of four different models.

3.3 Combining Class Specific and E-Mail Specific Perception

An analysis of the preliminary results obtained using the methods explained in Section 3.2 has revealed an interesting situation. Some messages in the test set that have been misclassified and whose spam and normal probabilities are very close to each other highly resemble to some of the messages of the correct class in the training set. For instance, a spam message is more similar to normal messages than the spam messages on the average (i.e. when the whole data set is considered) and thus is classified as normal, but in fact it is quite similar to a few of the spam messages. In such a case, if we took these specific messages into account rather than all the messages, it would be classified correctly.

Based on this fact, we propose a method that combines the class specific perception methods explained previously with an e-mail specific method. We divide the data set into training, validation, and test sets. The method is formed of two steps. In the first step, we use the methods of Section 3.2. However, only those messages for which the ratio of spam and normal probabilities exceeds a threshold are classified. The threshold values are determined using the validation set (VS) as follows:

$$\begin{aligned} f_{UB} &= \max\{\max\{f(E){:}E \in VS \textit{ and } E \textit{ is spam,}\},\ 1\} \\ f_{LB} &= \min\{\min\{f(E){:}E \in VS \textit{ and } E \textit{ is normal,}\},\ 1\} \end{aligned} \tag{11}$$

where f(E) gives the ratio of spam and normal probabilities for e-mail E:

$$f(E) = \frac{\mathrm{P(normal\,|\,E)}}{\mathrm{P(spam\,|\,E)}} \quad . \tag{12}$$

f_{UB} and f_{LB} stand for the upper bound and the lower bound, respectively, of the region containing the e-mails that could not be classified in the first step. We refer to this region as the uncertain region. f_{UB} corresponds to the ratio for the spam e-mail which seems “most normal”, i.e. the spam e-mail for which the method errs most. If f_{UB} is 1, there is no uncertainty about spam messages and all have been identified correctly. Similarly, f_{LB} corresponds to the ratio for the normal e-mail which seems “most spam”. If f_{LB} is 1, there is no uncertainty about normal messages.

In the second step, the messages within the uncertain region are classified. For this purpose, we use the same methods with a basic difference: each e-mail in the training set is considered as a separate class instead of having just two classes. In this way, the similarity of an incoming e-mail to each individual e-mail is measured. More formally, let C_k denote the class (e-mail) k, where k ranges over the e-mails in the training set. Then the equations for calculating the probability under the two methods that the e-mail E belongs to any C_k will be the same as equations (5) through (10), except that C is replaced with C_k. However in this case we have more than two classes and we cannot arrive at a decision by simply comparing their probabilities. Instead, we make the decision by taking the highest 10 scores and using a voting scheme:

$$\text{Decide}\begin{cases} \text{spam} & \text{, if} \sum_{i=1}^{10} coef_{\mathrm{max(i)}} \cdot P(C_{\mathrm{max(i)}} \mid E) > 0 \\ \text{normal} & \text{, otherwise} \end{cases} . \tag{13}$$

where max(i), 1≤i≤10, corresponds to k for which $P(C_k|E)$ is the largest i'th probability, and $coef_{max(i)}$ is 1 if $C_{max(i)}$ is spam and -1 otherwise. In short, among the 10 classes (e-mails) having the highest scores, equation (13) sums up the scores of spam classes and scores of normal classes, and decides according to which is larger.

4 Test Results

As stated in Section 2, three data sets have been built, each consisting of 1280 e-mails: data set for English e-mails, data set for Turkish e-mails with MA, and data set for Turkish e-mails without MA. In addition, from each data set, eight different data sets were formed: 160, 320, 480, 640, 800, 960, 1120, and 1280 messages. The messages in each of these eight data sets were selected randomly from the corresponding data set containing 1280 messages. Also the equality of the number of spam and normal e-mails was preserved. These data sets ranging in size from 160 to all messages were employed in order to observe the effect of the sample size on performance. Finally, in each execution, the effect of the first n-words heuristics was tested for eight different n values: 1, 5, 10, 25, 50, 100, 200, and all.

In each execution, the success rate was calculated using cross validation. The previously shuffled data set was divided in such a way that 7/8 of the e-mails were used for training (6/8 for training and 1/8 for validation, for combined method) and 1/8 for testing, where the success ratios were generated using eight-fold cross validation. Experiments were repeated with all methods and variations explained in Section 3. In this section, we give the success rates and time complexities. Due to the large number of experiments and lack of space, we present only some of the results.

4.1 Experiments and Success Rates

In the first experiment, we aim at observing the success rates of the two methods relative to each other and also understanding the effect of the first n-words heuristics. The experiment was performed on the English data set by using all the e-mails in the set. The result is shown in Figure 1. We see that the methods show similar performances; while the second method is better for classifying spam e-mails, the first method slightly outperforms in the case of normal e-mails. Among the two variations (a) and (b) of the methods for the sparse data problem, the latter gives more successful results and thus we use this variation in the figures. Considering the effect of the first n-words heuristics, we observe that the success is maximized when the heuristics is not used (all-words case). However, beyond the limit of 50 words, the performance (average performance of spam and normal e-mails) lies above 96%. We can thus conclude that the heuristics has an important effect: the success rate drops by only 1-2 percent with great savings in time (see Figure 5).

Following the comparison of the methods and observing the effect of the heuristics, in the next experiment, we applied the filtering algorithms to the Turkish data set. In this experiment, the first method is used and the data set not subjected to morphological analysis is considered. Figure 2 shows the result of the analysis. The maximum success rate obtained is around 95%, which is obtained by considering all the messages and all the words. This signals a significant improvement over the previous results for Turkish e-mails. The success in Turkish is a little bit lower than that in English. This is an expected result due to the morphological complexity of the language and also the fact that Turkish e-mails include a significant amount of English words. Both of these have the effect of increasing the dimensionality of the word space and thus preventing capturing the regularities in the data.

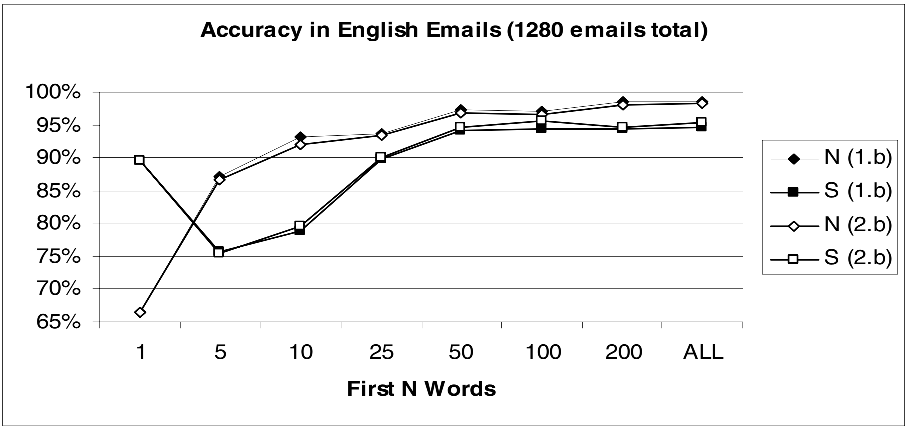


Fig. 1. Success rates of the methods for English data set

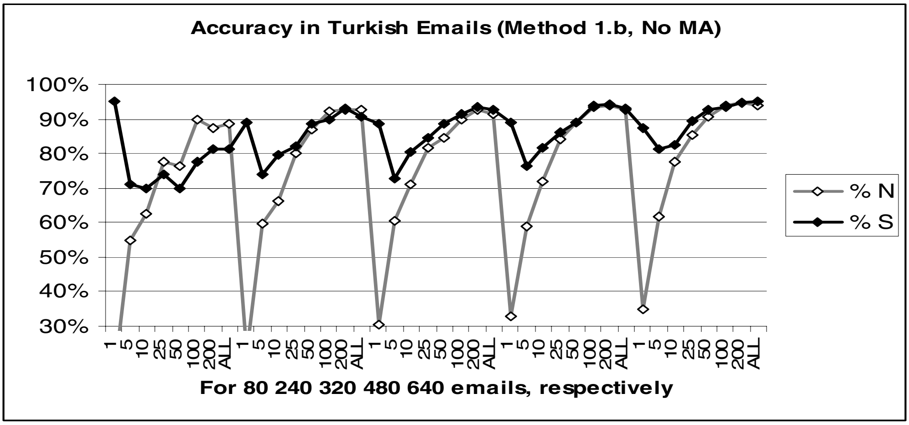


Fig. 2. Success rates in Turkish e-mails for different sample sizes

We observe a rapid learning rate. For instance, with 480 messages (240 normal and 240 spam), the performance goes up to 93%. Also, the usefulness of first n-words heuristics shows itself after about 50 words. 92% success is possible with that number of words (for 1280 e-mails). An interesting point in the figure that should be noted is the decline of success after some point. The maximum success in these experiments occur using 200 words. Thus, beyond a point an increase in the number of initial words does not help the filter.

The next experiment tests the effect of morphological analysis on spam filtering. The algorithms were executed on Turkish data sets containing root forms and surface forms. The results are shown in Figure 3. There does not exist a significant difference between the two approaches. This is in contrary to the conclusion drawn in [13]. The difference between the two works probably comes from the difference between the word sets used. Though a small subset of the words (a feature set) was used in the mentioned work, in this research we use all the words. This effect is also reflected in the figure: morphological analysis is not effective when all the words are used, whereas it increases the performance when fewer words are used (i.e. our first n-words heuristics roughly corresponds to the feature set concept in [13]). The fact that morphological analysis does not cause a considerable increase in performance may originate from two factors. First, it is likely that using only the root and discarding the affixes may cause a loss of information. This may be an important type of information since different surface forms of the same root may be used in different types of e-mail. Second, the algorithms choose randomly one of the roots among all possible roots of a word. Choosing the wrong root may have a negative effect on the success.

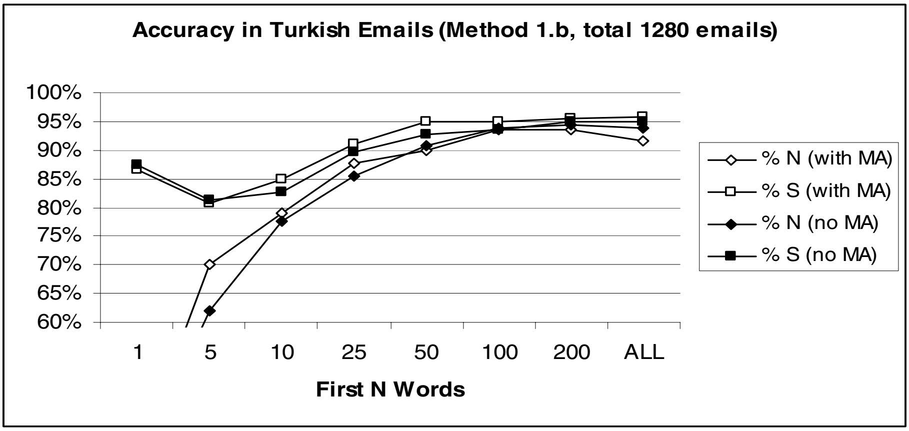


Fig. 3. Success rates in Turkish e-mails with MA and without MA

In the last experiment, we test the combined method explained in Section 3.3. Figure 4 shows the success rates (average of normal and spam) under this method and compares it with one of the previous methods. The figure indicates a definite increase in performance for Turkish data set with morphological analysis and the same situation occurs with the other two data sets as well. We have observed a significant error reduction of about 40-50% with the combined method for each data set size and first n-words. For instance, when all the messages in the data set are used with first 100-words, the success increases from 94.7% to 97.5%, which indicates about 47% improvement in error. Also the success rates reach their maximum values under this model: 98.5% for English and 97.5% for Turkish. So we conclude that the combined perception model achieves a quite high success rate with a low time complexity.

The time for training and testing is a function of the number of e-mails and the initial number of words. The execution times according to these two criteria for Turkish e-mails are shown in Figure 5. There is an exponential increase in time as the number of initial words increases. This effect reveals itself more clearly for larger sample sets. The positive effect of the first n-words heuristics becomes explicit. Although using all the words in the e-mails usually leads to the best success performance, restricting the algorithms to some initial number of words decreases the running time significantly. For instance, using the first 50 words instead of all the words reduces the time about 40 times. Finally, incorporating e-mail specific perception into the methods increases the execution time just by 20%.

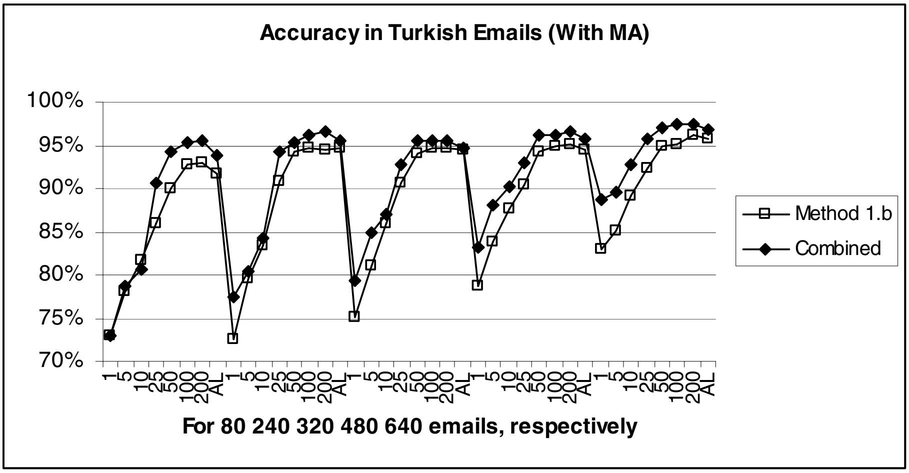


Fig. 4. Improvement in success rates with the combined method

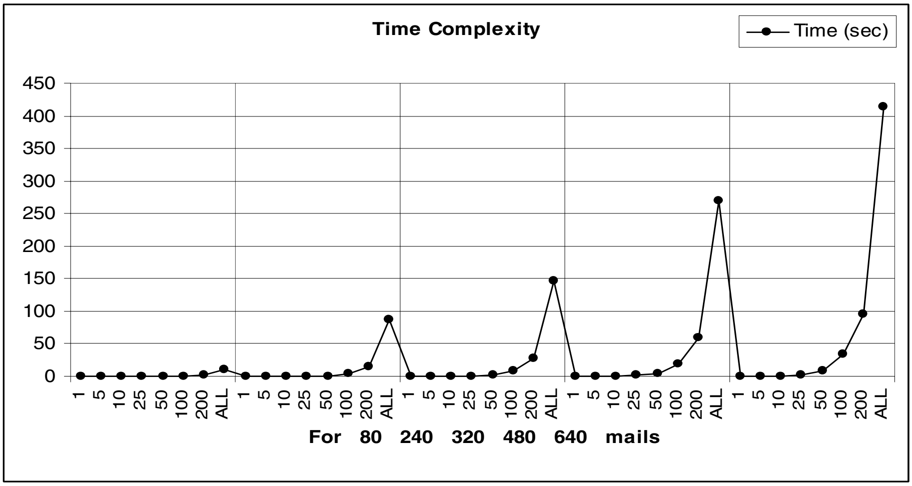


Fig. 5. Average execution times

5 Conclusions

In this paper, some simple but effective techniques have been proposed for spam filtering. The techniques achieved high success rates (97.5% for Turkish and 98.5%

for English) and caused execution time to decrease substantially. We performed extensive tests with varying numbers of data sizes and initial words. We observed the effects of these parameters on success rates and time complexities. The success rates reach their maximum using all the e-mails and all the words. However, training using 300-400 e-mails and 50 words results in an acceptable accuracy in much less time.

As a future work, we may use the affixes that contain additional information. Another extension is considering false positives and false negatives separately. In this respect, receiver operating characteristics (ROC) analysis can be combined with the technique here. This is a subject for future work involving cost-sensitive solutions. Some collaborative methods such as Safe Sender Listing may also be used [17].

Acknowledgements

This work was supported by Boğaziçi University Research Fund, Grant no. 04A101.

References

1. Burns, E.: New Image-Based Spam: No Two Alike (2006), http://www.clickz.com/ showPage.html ?page=3616946
2. Androutsopoulos, I., Koutsias, J., Chandrinos, K., Paliouras, G., Spyropoulos, C.: An Evaluation of Naive Bayesian Anti-Spam Filtering. In: Machine Learning in the New Information Age. Barcelona, pp. 9–17 (2000)
3. Sahami, M., Dumais, S., Heckerman, D., Horvitz, E.: A Bayesian Approach to Filtering Junk E-Mail. In: AAAI Workshop on Learning for Text Categorization. Madison, pp. 55–62 (1998)
4. Schneider, K.M.: A Comparison of Event Models for Naïve Bayes Anti-Spam E-Mail Filtering. In: Conference of the European Chapter of ACL. Budapest, pp. 307–314 (2003)
5. Cohen, W.: Learning Rules That Classify E-mail. In: AAAI Spring Symposium on Machine Learning in Information Access. Stanford California, pp. 18–25 (1996)
6. Drucker, H., Wu, D., Vapnik, V.N.: Support Vector Machines for Spam Categorization. IEEE Transactions on Neural Networks 10(5), 1048–1054 (1999)
7. Kolcz, A., Alspector, J.: SVM-Based Filtering of E-Mail Spam with Content-Specific Misclassification Costs. In: TextDM Workshop on Text Mining (2001)
8. Delany, S.J., Cunningham, P., Tsymbal, A., Coyle, L.: A Case-Based Technique for Tracking Concept Drift in Spam Filtering. Knowledge-Based Systems 18, 187–195 (2005)
9. Androutsopoulos, I., Paliouras, G., Karkaletsis, V., Sakkis, G., Spyropoulos, C., Stamatopoulos, P.: Learning to Filter Spam E-Mail: A Comparison of a Naïve Bayesian and a Memory-Based Approach. In: Workshop on Machine Learning and Textual Information Access. Lyon, pp. 1–13 (2000)
10. Zhang, L., Yao, T.: Filtering Junk Mail with a Maximum Entropy Model. In: International Conference on Computer Processing of Oriental Languages, pp. 446–453 (2003)
11. http://www.faqs.org/rfcs/rfc2554.html/
12. http://www.openspf.org/
13. Özgür, L., Güngör, T., Gürgen, F.: Adaptive Anti-Spam Filtering for Agglutinative Languages:A Special Case for Turkish. Pattern Recognition Letters 25(16), 1819–1831 (2004)

14. Oflazer, K.: Two-Level Description of Turkish Morphology. Literary and Linguistic Computing 9(2), 137–148 (1994)
15. Charniak, E.: Statistical Language Learning. MIT, Cambridge, MA (1997)
16. Manning, C.D., Schütze, H.: Foundations of Statistical Natural Language Processing. MIT, Cambridge, MA (2000)
17. Zdziarski, J.: Ending Spam: Bayesian Content Filtering and the Art of Statistical Language Classification. No Starch Press (2005)

Exploit Semantic Information for Category Annotation Recommendation in Wikipedia

Yang Wang, Haofen Wang, Haiping Zhu, and Yong Yu

APEX Data and Knowledge Management Lab,
Department of Computer Science and Engineering,
Shanghai JiaoTong University, Shanghai, 200240, P.R. China
{wwwy,whfcarter,zhu,yyu}@apex.sjtu.edu.cn

Abstract. Compared with plain-text resources, the ones in "semi-semantic" web sites, such as Wikipedia, contain high-level semantic information which will benefit various automatically annotating tasks on themself. In this paper, we propose a "collaborative annotating" approach to automatically recommend categories for a Wikipedia article by reusing category annotations from its most similar articles and ranking these annotations by their confidence. In this approach, four typical semantic features in Wikipedia, namely incoming link, outgoing link, section heading and template item, are investigated and exploited as the representation of articles to feed the similarity calculation. The experiment results have not only proven that these semantic features improve the performance of category annotating, with comparison to the plain text feature; but also demonstrated the strength of our approach in discovering missing annotations and proper level ones for Wikipedia articles.

Keywords: Collaborative Annotating, Semantic Features, Vector Space Model, Wikipedia Category.

1 Introduction

Nowadays, collaborative annotating has become more and more popular in various web scales. Some web sites directly supply an environment to share information and knowledge cooperatively contributed by millions of users. As a representative, Wikipedia[1] has become the largest free online encyclopedia and one of the top 20 most popular web sites on earth[2]. The English version of Wikipedia owns a prodigious number of more than 1.5 million articles[3].

More attractively, Wikipedia has many characteristics which are useful for building an automatic annotation system. It contains not only plain text, but also structural information[1] as source of semantic features including abundant inter-links, categories, sections and templates. Plain text content and semantic information coexist, which makes Wikipedia semi-semantic. Besides, we have

[1] http://en.wikipedia.org

[2] http://www.alexa.com

[3] http://en.wikipedia.org/wiki/Special:Statistics

Z. Kedad et al.(Eds.): NLDB 2007, LNCS 4592, pp. 48–60, 2007.

much confidence of using above characteristics which are refined by collective intelligence in the cooperative way[2]. Thus, semi-semantic sites can be looked to be promising bridges between the traditional web and the semantic web of which the advantages been widely acknowledged as convenient management, share and reuse of knowledge[3]. For more detail about semantic features, Section 3 makes comparison between Wikipedia and Ontology. Especially, category system in Wikipedia is a hierarchical taxonomy system for all articles in Wikipedia, which indicates generic information about articles and assists users to conveniently find target articles by navigationally focusing the scale of searching from more general categories to more specific ones. Hence, the category annotating is the foundational one among typical annotation tasks in Wikipedia.

In the manual editing way, most Wikipedia contributors may feel confused when designating categories to some edited article due to lack of global understanding of numerous existing category annotations, known as *information overload* problem[4]. For example, after we have edited an article about Bill Gates, we do not know which categories are suitable to assign to it though there are several proper categories existing in Wikipedia such as "Category:Billionaires". Then we are likely to assign it one arbitrary category "Category:Rich people" which can not describe Bill Gates well owing to missing other more exact annotations such as "Category:Forbes world's richest people". What is worse, improper categories are annotated. On the other hand, the category annotations need to update when the article is edited again. Therefore, the categories manually annotated need to check and refine again manually with huge labor and long time.

In terms of the problems and the idea of reusing for annotating, we propose a "collaborative annotating" approach to automatically recommend proper categories to Wikipedia articles. The approach mainly contains two steps. Given a target article to be annotated, the first one is to find its most similar evidences which will provide the candidate categories. The target article and evidences are represented by chosen four typical semantic features which widely exist in Wikipedia. In the second step, our system adopts a ranking algorithm to sort all candidate categories offered by the evidences. Finally those top ranked categories are returned to contributors as the recommended categories. Compared with heavyweight techniques, such as IE, NLP, etc, for knowledge acquisition from plain text[5], the real-time lightweight annotation recommendation we exploited does not need a large number of annotated data and long time for training.

The remainder of the paper is organized as follows. The next section dwells on the related work. Section 3 will introduces semantic features in Wikipedia. Section 4 elaborates on the method of category recommendation. Section 5 presents experiments for proving the validity and efficiency of the approach and some important findings are discussed. Finally, we address conclusions and future works.

2 Related Work

There are several work related to the approach we propose in this paper. Semantic annotation is a hot research direction in semantic web community.

[6] gives a full survey including notions, frameworks and tools of semantic annotation. Many tasks[7][8][9] adopt techniques, such as IE, NLP, etc, and lexical resource such as WordNet to extract annotations for traditional web pages, whose methodologies and objects are different from ones in our approach. Although the system PANKOW[10][11] aims to annotate public web resources with category annotations as well, their pattern-based method has low recall compared with our approach. [12] proposes to similarly reuse previous annotations from other resources but it adopts different methods and features to annotate images.

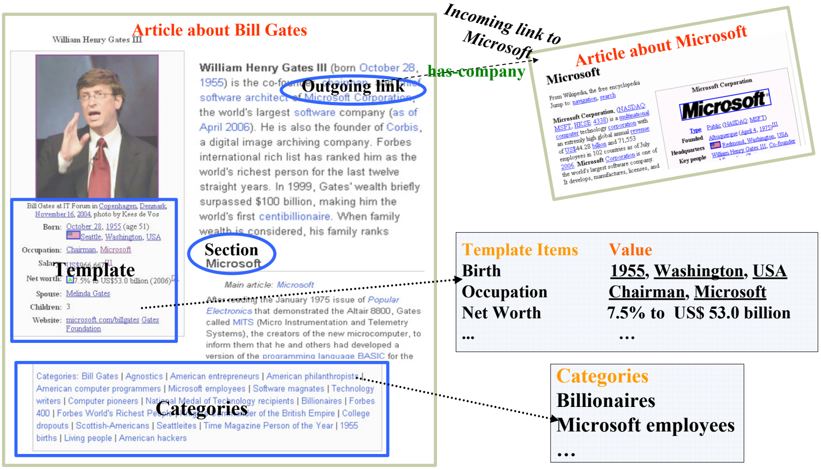
Many researchers make use of Wikipedia as a background to implement various applications. [13] learns lexical patterns for identifying relations between concepts in WordNet using Wikipedia as corpus. [14] proposes an algorithm to find similar articles for the target article. It only makes use of incoming link features and it recommends the link annotations with comparison to our category annotation recommendation. Annotating relations is another important topic in Wikipedia. [15] takes advantage of link information for mining relationship between categories. [2] adopts an automatic tagging procedure to learn semantic relation by generating candidates for link types.

Several recommendation tasks are exploiting the Collaborative Filtering as the state of the art in recent several years. As mentioned in several works[16][4], the k-NN method is the foundation to recommendation. [16] proposes a clustering algorithm to improve the efficiency of k-NN calculation. [4] combines CF and content information for boosting the performance of recommendation. The differences between our work and above tasks lie in that instead of personalized web recommendation, the semantic annotation is the focus of our task and the semantic feature representation for articles is employed in our approach. These features outperforms plain text for calculating the similarity. The conclusion is also acknowledged by other researchers[17][18][19]. Because the semantic features provide a high-level article presentation which avoids disadvantages of plain text, such as synonymy, polysemy, etc. Our appoach is also closely related to Instance-Based Learning[20], a Machine Learning algorithm.

3 Exploit Semantic Information in Wikipedia

If we treat categories as classes, articles as class instances and relation between supercategory and subcategory as super-sub class relation, it is natural to look on Wikipedia as an inherent ontology taxonomic system. In addition, the titles of articles and the names of categories are exclusive identifiers in Wikipedia which can be regarded as URI(Uniform Resource Identifier)[21]. For example, There is an article about Bill Gates with title “Bill Gates” which has categories named “Category:Billionaires”, “Category:Microsoft employees”, etc. Fig.1 gives an illustration. Usage of these title URIs prevents synonymy and polysemy problems instead of using IE and NLP techniques for NER and WSD.

In addition to the taxonomic semantics, there is a lot of other semantic information. Inter-links in Wikipedia aritlces build up an implicit semantic network, thus by clicking the hyperlinks user is shifted to other articles with same or



Template Items	Value
Birth	1955, Washington, USA
Occupation	Chairman, Microsoft
Net Worth	7.5% to US$ 53.0 billion
...	...

Categories
Billionaires
Microsoft employees
...

Fig. 1. A Wikipedia Article Sample Annotated with some Semantic Features

close topic[14]. The phenomenon indicates inter-links imply the semantic relations, related-to[22], between two article instances. In Fig.1, the link with the highlighting anchor text "Microsoft Corporation" denotes a outgoing link from "Bill Gates" and an incoming link to another article "Microsoft" at the same time, which implicitly denotes the relation "has-company" between them.

Sections structurally denote sub-topics which depict some attributes of current article. Especially, the section headings are the most representative identifiers of these sub-topics. A section can be mapped to an attribute with content in the section as value and its section heading as the attribute label. "Bill Gates" has a section "Microsoft" which describes the company owned by "Bill Gates".

Templates list many items which represent the properties of the instance described by the current article in a normative way. The items have values in plain text form or hyperlink form so each item in a template directly implies one attribute or relation. The template item name labels the attribute or relation. In Fig.1, there is a template listing items "born", "occupation", etc. The "occupation" item can be taken as a relation which associates "Bill Gates" with the article "chairman" and the item value is rendered with the underline to denote hyperlink form. Another item "Net Worth" with plain text value "7.5% to US$ 53.0 billion" can be regarded as an attribute of Bill Gates.

Given to above elaborate description, we find Wikipedia can be taken as a huge semantic repository. We employ these semantic elements as features to represent articles for assisting to recommend categories.

4 Category Recommendation Method

It is recommended to reuse the existing categories to annotate an article, from both Wikipedia and the semantic web perspective. We employ the "k-NN" thought that categories in similar articles would suggest the usage of such categories in the target article. Besides, we have an assumption that if two articles have more commonalities on the four types of semantic features mentioned above, they also have more commonalities on category annotations. Thus, our method for category recommendation consists of two main steps:

Evidence Generation. It is believed that a few closely similar articles, referred as evidences, may suffice to identify the most important categories. In this step, we use *Information Retrieval* based method and document representation in Vector Space Model(VSM) owing to its efficiency and effectiveness. We choose four main semantic features as mentioned above from Wikipedia for article representation and similarity measuring:

1. Incoming links. Given a Wikipedia article d, collect all titles of the articles that link to d. Let d_I be the resulting bag of terms $\{t_{I_1}, \cdots, t_{I_{|d_I|}}\}$. We call this the *incoming link representation* of d. The similarity measure is that two articles are similar if they are co-cited by a third.
2. Outgoing links. Given a Wikipedia article d, collect all titles of the articles that are linked by d . Let d_O be the resulting bag of terms$\{t_{O_1}, \cdots, t_{O_{|d_O|}}\}$. We call this the *outgoing link representation* of d. The similarity measure is that two articles are similar if they both refer to a third.
3. Section headings. Given a Wikipedia article d, collect all section heading names in d. Let d_S be the resulting bag of terms$\{t_{S_1}, \cdots, t_{S_{|d_S|}}\}$. We call this the *section heading representation* of d. The similarity measure is that two articles are similar if they share their section heading names a lot.
4. Template items. Given a Wikipedia article d, collect all template item names in d. Let d_T be the resulting bag of terms$\{t_{T_1}, \cdots, t_{T_{|d_T|}}\}$. We call this the *template representation* of d. The similarity measure is that two articles are similar if they share their template item names a lot.

Finally, we construct four fields d_I, d_O, d_S and d_T in a virtual document vd which represents d. We assume that the common feature counts in the four fields correlate with the strength of similarity. Thus, we adopt a standard TF-IDF model to measure similarity between target article d and an article d_i, namely $sim(d_i, d)$. For each article d users are editing, we use its virtual documents representation as query representation in Boolean form as well to retrieve a ranked list, and collect Top-n similar articles $d_1, d_2, \cdots, d_n$ as evidences collection $D(d)$ which underlies the next procedure.

Another feature used usually is plain text. However, we eventually abandon using it because its little performance improving and expensive computation cost shown in our experiments. Besides, There are several parameters to be identified such as the evidence amount n and the different weight w_j for each feature.

Candidates Ranking. The second step involves ranking of the candidate categories collected from $D(d)$ for recommendation. Let f be a ranking function that determines the preference relation between categories. For a candidate category c, all evidence articles are annotated by c compose a set, namely EA_c

$$f_c = \sum_{d_i \ \in \ EA_c} plus_{c,d_i} \tag{1}$$

where the $plus_{c,d_i}$ denotes the contribution to category c preference from article d_i. The f_c function is the sum of all contributions from articles in $D(d)$.

We design five ranking functions by defining the $plus_{c,d_i}$:

1.*DIRECT COUNT abbr.* DC. We regard all evidence articles as of the same importance and simply make use of the "voting" idea to rank a category in the light of the count of these evidence articles annotated by it. It indicates that the more popular one category is in $D(d)$, the higher rank it should have:

$$plus_{c,d_i} = 1. \tag{2}$$

2. *WEIGHTED COUNT abbr.* WC. Based on the DC mechanism, we take the similarity degree together with the targeted article into account. It means the category provided by more similar evidence articles will be more important:

$$plus_{c,d_i} = sim(d_i, d). \tag{3}$$

3. *BOOSTED WEIGHTED COUNT abbr.* BWC. We modify the importance factor by adding a boost coefficient $|D|/rank_{d_i}$, where the $|D|$ denotes the number of articles as evidences i.e. n value in Top-n and the $rank_{d_i}$ denotes the position of article d_i in the ranking list.

$$plus_{c,d_i} = sim(d_i, d) \cdot |D|/rank_{d_i}. \tag{4}$$

It gives different weighting for each article, besides their similarity degree to the targeted article, the most similar article is amplified by $(N - 1)$ times, i.e. the boost is $|D|$, while the least similar one keeps as-is.

4.*GLOBAL POPULARITY REWARDED BOOSTED COUNT abbr.* GPR. Furthermore, we consider the global popularity into rank value of each candidate category. The modification is inspired by the vision that the more popular one category is in the whole corpus, the more suitable it is to annotate articles.

$$plus_{c,d_i} = (sim(d_i, d) \cdot |D|/rank_{d_i}) \cdot popularity(c). \tag{5}$$

where the $popularity(c)$ denotes the amount of articles under the category c.

5.*GLOBAL POPULARITY PUNISHED BOOSTED COUNT abbr.* GPP. On the contrary, we consider the global popularity into rank value of each category compared to previous ranking function. The modification is inspired by the vision that the more popular one category is in the whole corpus, the lower importance one article in the category has in the recommendation.

$$plus_{c,d_i} = (sim(d_i, d) \cdot |D|/rank_{d_i})/popularity(c). \tag{6}$$

5 Experiment and Analysis

We evaluate our approach in terms of several aspects detailedly including identifying different importance of each semantic feature, combining them with different weights and selecting the best ranking function. The number of article evidences and the preferred number of recommended categories are investigated on how they impact the recommendation performance. The experiments have also demonstrated the strength of our approach in discovering missing categories for target article from existing evidence articles.

5.1 Experiment Setting and Metrics

We collected data from the WikipediaXML Corpus [23] to build the testbed. WikipediaXML is an XML-version snapshot of Wikipedia in early 2006. This corpus contains 59,388 articles, 113,483 categories, 20,021,059 links, 1,086,757 section headings and 970,134 template items. We built and stored the indices of these data in the IBM DB2 database system.

Lucene is employed as the backend search engine for the whole system. Each article consists of four feature fields, namely incoming link, outgoing link, section heading, template item as mentioned in Section 4. The similar article collection will be returned by querying the index.

In the following evaluations, we adopt the original categories assigned to target article as the *Golden Standard* and compare them with the *recommended* categories to assess the performance. The standard Information Retrieval criteria are employed: Precision, Recall and $F1$.

5.2 Evaluation on Different Features

First of all, the different importance of each feature is evaluated. Because various features has various expressiveness in representing articles, the investigation of their relative expressiveness underlies selecting features and assigning the respective combination weights.

Here, we choose the plain text feature as the baseline to compare with the semantic features. The plain text representation employs the words which have moderate *Document Frequency* values because words with high *Document Frequency* values have weak expressiveness and will decrease the speed of queries, while words with low *Document Frequency* value are too insufficient to collect enough similar articles by queries owing to the sparsity problem. To resolve the tradeoff between the two impacts, we choose 300 words with the most large *Document Frequency* values but not more than 10,000 *Document Frequency* value. Besides, the stopwords were ignored and words were stemmed. We also indexed plain text representation of articles with Lucene. Then, 300 articles from 6 different domains, 50 articles each domain, were sampled as test data. We investigated different amounts of articles returned as evidences by queries, i.e. k value of k-NN. All the categories from evidences are regarded as the recommended categories without exploiting any ranking algorithm at this section. The recall values for different features are depicted in Fig.2.

From the left part of the figure, it reveals that the incoming link feature has the most remarkable advantage in representing articles. Besides, the outgoing link feature has the comparable effect. The plain text feature also has good effectiveness compared with other two features section heading and template item

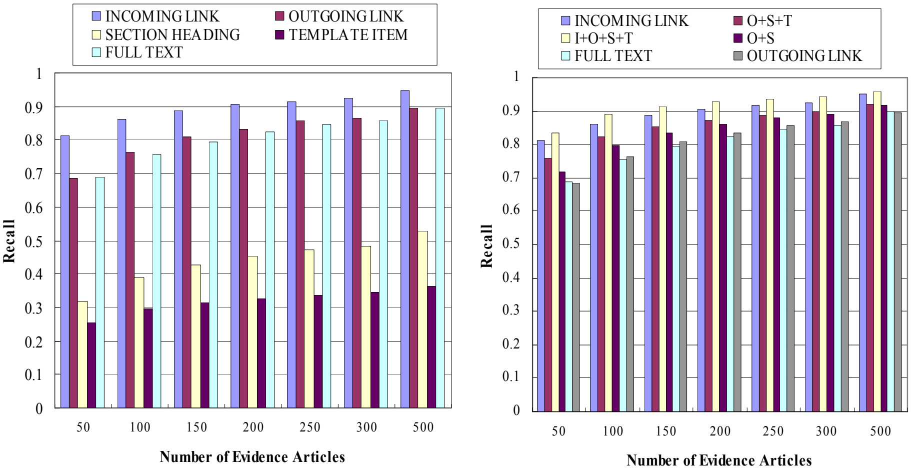


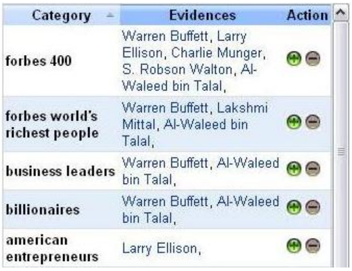
Fig. 2. Comparison between various features and combination

which poorly perform due to their sparseness. The reason is that to recommend all categories from evidence articles without ranking them is relatively looser evaluation; therefore, we add the precision and $F1$ for the further evaluation in next section. Besides, a serious disadvantage of plain text feature exists: The query response time is almost intolerable as the long returned results list which makes it unsuitable for real-time applications, which is average 20,441 ms for each sample target article (on a 2.4GHz CPU, 1GB RAM workstation and to limit the returned article number not more than 10,000). However, incoming link, outgoing link, section heading and template items features respectively need shorter response time : 469ms, 730ms, 138ms, 31ms.

According to the different performance of each feature, we give them different weights respectively. Weights $w_I = 0.4, w_O = 0.3, w_S = 0.2, w_T = 0.1$ are assigned for incoming link, outgoing link, section heading, template item, which can also be easily trained by regression technique. The performance of several combinations is shown in the right part of Fig.2. The combinations includes:

- Outgoing link + Section *abbr.* O+S
- Outgoing link + Section heading + Template item *abbr.* O+S+T
- Incoming link + Outgoing link + Section heading + Template item *abbr.* I+O+S+T

The combination I+O+S+T outperforms others because the features perform well on a reciprocal basis. Thereby, it is selected as the default article representation. For newly created articles, it is difficult to acquire incoming links. Luckily, the O+S and O+S+T perform beyond outgoing link and plain text and has the comparable effect with incoming link. In terms of efficiency, all combinations perform as fast as incoming link alone and need only 3%-5% time of plain text feature, less than 1 second per query. In addition, the number of evidence



Category	Evidences	Action
forbes 400	Warren Buffett, Larry Ellison, Charlie Munger, S. Robson Walton, Al-Waleed bin Talal,	⊕⊖
forbes world's richest people	Warren Buffett, Lakshmi Mittal, Al-Waleed bin Talal,	⊕⊖
business leaders	Warren Buffett, Al-Waleed bin Talal,	⊕⊖
billionaires	Warren Buffett, Al-Waleed bin Talal,	⊕⊖
american entrepreneurs	Larry Ellison,	⊕⊖

Fig. 3. Top 5 Recommended Categories and Corresponding Evidence Articles for "Bill Gates"

articles impacts recall value. However, it has a bit influence on the recall when it exceeds 200. The larger number will depress the efficiency of ranking algorithm. Considering a tradeoff between performance and efficiency, we chose 200 article evidences, i.e. 200-NN, in the further experiments.

5.3 Evaluation on Different Ranking Functions

In this section, we put main attention on the investigation of the performance of the five ranking functions mentioned in Section 4. It indicates that the number of recommended categories will impact both precision and recall because the ranking algorithm which perform well will rank proper categories at top of the returned list. In Table 1, The yellow shade(light) denotes the best performing ranking function using certain feature(s), reversely, the gray shade(deep) denotes the worst one. It shows that the *Boosted Weighted Count*(BWC) ranking function performs best no matter how the number of recommended categories(denoted by CAT.x in the table) and the semantic features change. The performance of *Weighted Count*(WC) is very close to BWC which considers article ranking position as a boost factor. The DC, WC and GPR are sensitive to different features. Neither GPR nor GPP can improve the performance in all situations. Thus, BWC is selected as default ranking function in the further experiment.

Additionally, the various features and their combinations are also investigated by comparison. In Table 1 the plain text feature performs poorly compared to incoming link, outgoing link and section. Given poor performance and low efficiency mentioned above, we ultimately abandon it in feature combinations. Other combinations also improve the performance of using respectively features through experiments. They are not shown in Table 1 for the sake of space. In the further experiments, I+O+S+T combination will be finally set as the default article representation for its best performance.

In Fig.3, we give a example of recommending categories for article "Bill Gates" using our original system, using the default I+O+S+T feature combination, BWC ranking function and 200 articles as evidences. The "Forbes world's

Table 1. Ranking Algorithm Comparison Using Different Features. (precision / recall)

FEATURE	RANK. FUNC.	CAT.1	CAT.3	CAT.5	CAT.7	CAT.9
I+O+S+T	***BWC***	**1.00 / 0.25**	**0.93 / 0.60**	**0.81 / 0.80**	**0.67 / 0.87**	**0.55 / 0.90**
	DC	0.59 / 0.12	0.44 / 0.26	0.36 / 0.36	0.32 / 0.44	0.29 / 0.49
	WC	0.98 / 0.24	0.91 / 0.59	0.78 / 0.78	0.65 / 0.85	0.54 / 0.89
	GPR	0.98 / 0.24	0.92 / 0.59	0.79 / 0.78	0.65 / 0.85	0.54 / 0.88
	GPP	0.98 / 0.24	0.90 / 0.58	0.77 / 0.76	0.64 / 0.85	0.54 / 0.88
Incoming Link	***BWC***	**1.00 / 0.25**	**0.92 / 0.59**	**0.80 / 0.78**	**0.66 / 0.86**	**0.54 / 0.88**
	DC	0.55 / 0.12	0.44 / 0.26	0.36 / 0.35	0.31 / 0.42	0.28 / 0.48
	WC	0.97 / 0.24	0.90 / 0.58	0.78 / 0.77	0.65 / 0.85	0.53 / 0.87
	GPR	0.98 / 0.24	0.90 / 0.58	0.77 / 0.76	0.63 / 0.83	0.53 / 0.87
	GPP	0.98 / 0.24	0.88 / 0.56	0.76 / 0.75	0.63 / 0.83	0.52 / 0.86
Outgoing Link	***BWC***	**0.99 / 0.24**	**0.90 / 0.57**	**0.76 / 0.74**	**0.62 / 0.81**	**0.51 / 0.83**
	DC	0.42 / 0.09	0.32 / 0.20	0.28 / 0.28	0.25 / 0.35	0.22 / 0.39
	WC	0.97 / 0.24	0.88 / 0.56	0.75 / 0.73	0.61 / 0.80	0.50 / 0.82
	GPR	0.96 / 0.22	0.87 / 0.55	0.74 / 0.72	0.61 / 0.79	0.50 / 0.82
	GPP	0.97 / 0.24	0.85 / 0.55	0.72 / 0.71	0.60 / 0.78	0.50 / 0.81
Section Heading	***BWC***	**0.73 / 0.16**	**0.61 / 0.34**	**0.49 / 0.42**	**0.38 / 0.45**	**0.31 / 0.46**
	DC	0.14 / 0.02	0.13 / 0.07	0.12 / 0.11	0.11 / 0.14	0.10 / 0.17
	WC	0.57 / 0.12	0.47 / 0.26	0.40 / 0.35	0.32 / 0.38	0.27 / 0.40
	GPR	0.71 / 0.15	0.57 / 0.32	0.45 / 0.40	0.36 / 0.43	0.30 / 0.45
	GPP	0.54 / 0.11	0.46 / 0.26	0.38 / 0.35	0.32 / 0.38	0.28 / 0.42
Template Items	***BWC***	**0.16 / 0.03**	**0.14 / 0.07**	**0.11 / 0.09**	**0.09 / 0.11**	**0.08 / 0.11**
	DC	0.13 / 0.02	0.09 / 0.05	0.08 / 0.07	0.07 / 0.09	0.06 / 0.10
	WC	0.16 / 0.03	0.12 / 0.07	0.10 / 0.09	0.09 / 0.11	0.08 / 0.13
	GPR	0.16 / 0.03	0.14 / 0.07	0.11 / 0.10	0.10 / 0.11	0.09 / 0.13
	GPP	0.05 / 0.01	0.05 / 0.02	0.05 / 0.04	0.04 / 0.05	0.04 / 0.06
Plain Text	***BWC***	**0.46 / 0.10**	**0.36 / 0.21**	**0.29 / 0.29**	**0.24 / 0.34**	**0.20 / 0.36**
	DC	0.36 / 0.08	0.21 / 0.13	0.19 / 0.19	0.17 / 0.24	0.15 / 0.27
	WC	0.46 / 0.09	0.33 / 0.19	0.29 / 0.27	0.24 / 0.32	0.22 / 0.37
	GPR	0.34 / 0.07	0.28 / 0.17	0.24 / 0.22	0.21 / 0.29	0.19 / 0.33
	GPP	0.11 / 0.02	0.10 / 0.06	0.09 / 0.08	0.11 / 0.15	0.12 / 0.22

richest people" is recommended for "Bill Gates" by evidences such as a billionaire "Warren Buffett", etc.

Best Recommendation Number. Furthermore, the Table 1 indicates that when we only recommend the several categories on the top of ranking list to users, the precision values are very high. For example, the precision reaches 81% when we use BWC and return top 5 categories, which means that 4 out of 5 categories are proper(Especially, 100% for top 1 category). However, when we recommend 9 categories by BWC, the precision value is only 55%. It means that 5 out of 9 categories is proper. The situation broadly exist. It implies that our approach tends to recommend the proper categories in most front, which fits the users' habits of liking to get right answers in the most front of the returned list. Besides, we find that the recall value will be lifted by increasing the number of recommended categories from Table 1. Therefore, we should take $F1$ into consideration for the sake of tradeoff of precision and recall.

In Fig.4, the broad-brush curve in black color indicates the average $F1$ measure for all sampled articles without considering domain and suggests that the preferred number of recommended categories is about 5. Meanwhile, it is proven that our approach is domain-independent by comparing $F1$ values in representational domains: "people", "company", "location", "film", "sports" and "jargon".

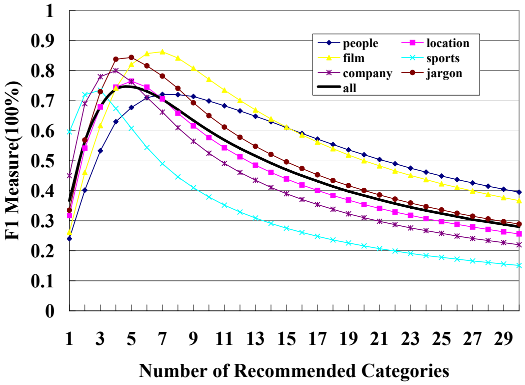


Fig. 4. F1 Measure for Different Domains and All Samples

5.4 Other Important Findings

In the period of experiments, some important and interesting phenomena are observed. We summarize them in follows:

1. Our approach is able to suggest missing categories. For example, in the recommendation list for “United Kingdom” there exists a category named “Category:Island nations”, whose evidences consist of “Republic of Ireland”, “Australia”, etc. The category does not exist in the article for “United Kingdom” in wikipediaXML, but is later appended to its updated version (with a synonymous category name “Category:Island countries”) in the current Wikipedia. We gather 606 such missing categories for sampled 300 articles(Dec. 10, 2006). In the experiment, 47 missing categories are recommended by our algorithm(10 recommended categories per article). In addition, we find the main content in almost all these articles only received minor editing by observing the sampled editing logs. Therefore, the articles should be annotated by the missing categories.

2. Our approach can categorize an article to the proper level of abstraction. Take the article for “Support Vector Machine” as an example, our method tends to rank “Category: Machine Learning” on a higher position of recommendation list than ”Category: Artificial Intelligence”. Additionally, by consulting the category hierarchy we find that it subsumes the category “Category: Machine Learning” which is already associated with the article and is a sub-category of ”Category: Artificial Intelligence”. Though the latter is actually a relevant category, our approach does not advise to assign it to “Support Vector Machine” if the more specific category ”Category: Machine Learning” is already used.

6 Conclusion

This paper proposes a new approach to automatically annotating category by consulting the similar articles represented by semantic features in wikipedia. To reuse the collaborative annotations, which include not only category annotations, but also semantic features, is the characteristic of our approach. The approach is proven efficient and can be used for other similar applications in other similar environments which provide similar features.

The future researches contain several aspects: (1) refine the feature representation for articles to improve the performance, such as adding the template name to improve the accuracy, (2) adopt a specific method to assess the weights for different semantic features, such as EM, (3) to use the approach to other environments and evaluate the results, The famous site *http://del.icio.us* is a suitable choice owing to the rich tagging information.

References

1. Voss, J.: Collaborative thesaurus tagging the Wikipedia way. Wikimetrics (2006)
2. Ruiz-Casado, M.: From Wikipedia to Semantic Relationships: a semi-automated Annotation Approach. SemWiki (2006)
3. Lee, T.B., Hardler, J., Lassila, O.: The Semantic Web. Scientific American Magazine (2001)
4. Melville, P., Mooney, R.J., Nagarajan, R.: Content-Boosted Collaborative Filtering for Improved Recommendations. AAAI (2002)
5. Buitelaar, P.: Ontology Learning from Text. Tutorial at ECML/PKDD (2005)
6. Uren, V., Cimiano, P., Iria, J., Handschuh, S., Ciravegna, F.: Semantic annotation for knowledge management: Requirements and a survey of the state of the art. Journal of Web Semantics (2005)
7. Mukherjee, S., Yang, G., Ramakrishnan, I.V.: Automatic Annotation of Content-Rich HTML Documents:Sementic Analysis. ISWC'03
8. Erdmann, M., Maedche, A.: From Manual to Semi-automatic Semantic Annotation: About Ontology-based Text Annotation Tools. Semantic Annotation (2000)
9. Kiryakov, A., Popov, B., Terziev, I.: Semantic Annotation, Indexing, and Retrieval. Journal of Web Semantics (2004)
10. Cimiano, P., Handschuh, S., Staab, S.: Towards the SelfAnnotating Web. WWW'04
11. Cimiano, P., Handschuh, S., Staab, S.: Gimme' the context: context-driven automatic semantic annotation with C-PANKOW. WWW (2005)
12. Marques, O., Barman, N.: Semi-Automatic Semantic Annotation of Images Using Machine Learning Techniques. In: Fensel, D., Sycara, K.P., Mylopoulos, J. (eds.) ISWC 2003. LNCS, vol. 2870, Springer, Heidelberg (2003)
13. Ruiz-Casado, M.: Automatic Extraction of Semantic Relationships forWordNet by Means of Pattern Learning fromWikipedia. In: Montoyo, A., Muńoz, R., Métais, E. (eds.) NLDB 2005. LNCS, vol. 3513, Springer, Heidelberg (2005)
14. Adafre, S.F., de Rijke, M.: Discovering Missing Links in Wikipedia. LinkKDD (2005)
15. Chernov, S., Iofciu, T.: Extracting Semantic Relationships between Wikipedia Categories. SemWiki (2006)
16. Mobasher, B., Dai, H., Luo, T., Nakagawa, M.: Improving the Effectiveness of Collaborative Filtering on Anonymous Web Usage Data. IJCAI'01
17. Fan, W., Gordon, M.D.: Ranking Function Optimization For Effective Web Search By Genetic Programming: An Empirical Study. HICSS 2003 (2003)
18. Bloehdorn, S., Hotho, A.: Boosting for Text Classification with Semantic Features. In: Proceeding of Text Information Rerieval 2004 (2004)

19. Liddy, E.D., Paik, W., Yu, E.S.: Text Categorization for Multiple Users Based on Semantic Features from a Machine-Readable Dictionary. ACM TransactIons on Information Systems 12(3), 278–295 (1994)
20. Aha, D.W., Kibler, D., Albert, M.K.: Instance-Based Learning Algorithms. Journal of Machine Learning 6, 37–66 (1991)
21. Hepp, M.: Harvesting Wiki Consensus - Using Wikipedia Entries as Ontology Elements. In: Processing of ESWC workshop, SemWiki 2006 (2006)
22. Vöel, M., Krözsch, M., Vrandecic, D., Haller, H., Studer, R.: Semantic Wikipedia. WWW 2006 (2006)
23. Denoyer, L.: The Wikipedia XML Corpus. SIGIR Forum 2006 (2006)

A Lightweight Approach to Semantic Annotation of Research Papers

Nicola Zeni[1], Nadzeya Kiyavitskaya[1], Luisa Mich[2], John Mylopoulos[1],
and James R. Cordy[3]

[1] Dept. of Information and Communication Technology,
University of Trento, Italy
{nadzeya,nzeni,john.mylopoulos}@dit.unitn.it
[2] Dept. of Computer and Management Sciences,
University of Trento, Italy
luisa.mich@unitn.it
[3] School of Computing, Queens University, Kingston, Canada
cordy@cs.queensu.ca

Abstract. This paper presents a novel application of a semantic annotation system, named Cerno, to analyze research publications in electronic format. Specifically, we address the problem of providing automatic support for authors who need to deal with large volumes of research documents. To this end, we have developed Biblio, a user-friendly tool based on Cerno. The tool directs the user's attention to the most important elements of the papers and provides assistance by generating automatically a list of references and an annotated bibliography given a collection of published research articles. The tool performance has been evaluated on a set of papers and preliminary evaluation results are promising. The backend of Biblio uses a standard relational database to store the results.

Keywords: bibliography generation, semantic annotation.

1 Introduction and Motivation

This paper presents a novel application of the lightweight semantic annotation method founded on the Cerno system [1,2] to the task of research publication analysis. At present, information mining for collections of published research papers is supported by search engines such as Google Scholar [3], the DBLP repository [4], and electronic libraries such as Citeseer [5], the ACM Portal [6], and the IEEE Digital Library [7]. These services provide access to large knowledge bases of research publications, allowing a user to search for a paper and retrieve the details of its publication. In digital libraries, it is also possible to see the abstract and citations present in the paper.

However, organization of a personal collection of electronic publications remains a manual task for authors. The authors of a new research paper are often forced to spend time analyzing the content of accumulated related work to find information relevant to the results being published. Such effort is required for any kind of scientific publication (e.g., dissertation, technical report, book, or journal article) in

Z. Kedad et al.(Eds.): NLDB 2007, LNCS 4592, pp. 61–72, 2007.

order to complete a review of the state-of-the-art and provide an appropriate bibliography of related work. Obviously, authors could benefit from assistance in collecting, managing and analyzing relevant literature. Such assistance can come from tools that perform the analysis of relevant literature with "good enough" results. Such an analysis could also be used to provide other helpful services, such as – for example – the automatic construction of a bibliography for selected articles.

The difficulties posed by a research publications analysis task may not be obvious to the reader. Here are some of the issues that need to be addressed:

- *Variable document formats*. Document source files may be stored as PDF, MS Word, LaTeX, PostScript, HTML and other electronic formats;
- *Page layout.* Depending on the requirements of the publisher, the layout of a document varies; for example, pages may be organized in a one or two-column fashion; for papers published in journals, the header may be present on some pages; footnotes may be allowed or not, and so on;
- *Document structure*. At first sight, one may assume that scientific documents are semi-structured; this means that the arrangement of document elements can be predicted with some certainty: for instance, first there is a title, then a list of authors with affiliations, abstract, introduction and other sections; unfortunately, such structure is not universally adopted;
- *Semantic analysis*. The key elements that a reader (for example, a reviewer) looks for are the problem considered and the main contributions. The reader may also be interested in finding background knowledge on which the work is based, particularly references to related work;
- *Bibliography generation*. Normally, in order to generate each item of the bibliography, the writer has to manually find all bibliographical details of a selected document and fill the required template with this data.

In this work, we address all these issues and provide semantic and structural annotation of document sections. This paper presents Biblio, a tool that is based on Cerno and is intended to support the analysis of research articles. Cerno is a semantic annotation system based on software code analysis techniques and tools proven effective in the software analysis domain for processing billions of lines of legacy software source code [1]. Cerno has been already applied in several case studies involving the analysis of different kinds of documents from the tourism sector [2,3]. The system has demonstrated good performance and scalability while yielding good quality results. In this work we also present preliminary experimental results of the technique on a set of published papers. Apart from semantic markup, Biblio also supports the generation of a bibliography in different formats.

The rest of the paper is structured as follows: Section 2 reviews the semantic annotation method of Cerno. Section 3 explains how this method can be adapted for the domain of electronic literature analysis annotation, providing some insights on the implementation details. Section 4 illustrates the document analysis process on a specific example. Section 5 provides results of a limited evaluation of Cerno-based annotation for a set of research papers, while section 6 surveys related work, and conclusions are drawn in Section 7.

2 Text Analysis with Cerno

Our document analysis is based on Cerno [1, 2], a semantic annotation system that utilizes highly efficient methods of software (code) analysis and reengineering for text processing. Cerno is based on TXL [8], a generalized parsing and structural transformation system. TXL is especially designed to allow rapid prototyping and processing of text patterns. The formal semantics and implementation of TXL are based on formal tree rewriting, but the trees are largely hidden from the user due to the by-example style of rule specification. The system accepts as input a grammar (set of text patterns) and a document, generates a parse tree for the input document, and applies transformation rules to generate output in a target format. TXL uses full backtracking with ordered alternatives and heuristic resolution, which allows efficient and flexible parsing.

The architecture of Cerno consists of a series of components that perform sequential transformations on an input document:

1. *Parse*. First, the system breaks down raw input text into its constituents, producing a parse tree of an input document. The peculiarity of parsing textual documents with Cerno, compared to design recovery of source code, is that in this case, the parse tree is composed of such structures as *document, paragraph, phrase* and *word,* rather than *program, function, expression*, and so on. In this stage, annotation fragments are delimited according to a user-defined grammar. At the same time, complex word-equivalent objects, such as phone numbers, e-mail and web addresses, and so on, are recognized using predefined structural patterns of object grammars. All grammars are described in BNF (Backus Naur Form)-like form using the TXL language notation.
2. *Markup*. The next stage uses an annotation schema to infer annotations of document fragments. This schema contains a list of concept names drawn from the domain-dependent model and a vocabulary consisting of indicators related to each concept. Cerno assumes that this domain input is constructed beforehand either automatically using some learning methods or manually in collaboration with domain experts. Indicators can be literal words, phrases or names of parsed entities. They also can be *positive*, i.e., pointing to the presence of the given concept, or *negative*, i.e., pointing to the absence of this concept. If a positive indicator is present in a text fragment, Cerno annotates the fragment with a corresponding tag, unless a negative indicator is present.
3. *Mapping*. In the last stage, annotated fragments are selected from all annotations according to a database schema template schema, and copied to an external database. The database schema is manually derived from the domain-dependent semantic model and represents a set of fields of a target database.

In Biblio application we partially adapt Cerno to perform semantic annotation using structural, syntactic and semantic information.

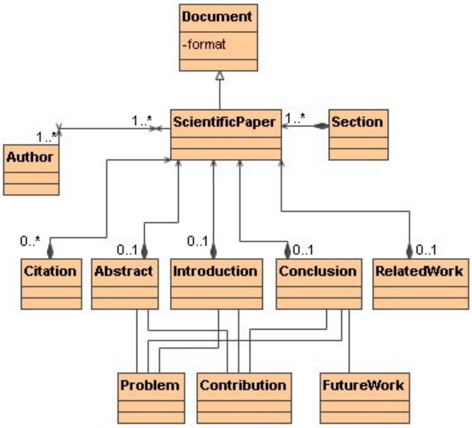
In the following section, we demonstrate our application of Cerno to the new domain of research publications analysis.

3 The Biblio Tool for Analyzing Research Papers

The design of Biblio was based on a set of initial requirements:

- *Plain text extraction.* As the input for the Cerno system must be provided in plain text format, a preprocessing stage is required in order to convert files in various formats to plain text. The main formats that must be accepted are PDF and MS Word, as they are wide-spread in scientific publishing.
- *Analysis of document structure.* The tool should be able to identify the main structural elements of the document and, in particular, to identify for the user such sections as title and author information, abstract, introduction, conclusions, and references.
- *Semantic annotation.* The tool should provide semantic recognition and extraction of important aspects of the document such as the problem addressed and the claimed contributions. Most often, phrases that briefly and explicitly describe the main achievements of a work are located in the abstract, introduction and conclusion sections. Based on this premise, we will look for this information only in these sections and ignore the remaining content of the file.
- *Other functionalities.* Apart from the functions listed above, the tool should provide the user a possibility to view all the fields, to fill or edit their contents and to execute various queries over the uploaded bibliographical data. Finally, by request, the tool must produce a citation line for a specified document in the necessary format and visualize in the interface the complete bibliography.

In order to use Cerno, we first had to elicit domain-dependent knowledge. For this purpose, we designed a conceptual model representing scientific document content, as shown on the left in Fig. 1. On the basis of this model, the annotation schema and database schema template was derived, as seen on the right in Fig. 1. The underlying database schema includes the tables *Document, Document_author, Document_citation, Document_contribution, Document_problem, Document_Future_Work, Author, Citation* and others, as shown in Fig. 2.



```
<Document>
   <Title></Title>
   <Authors></Authors>
   <Email></Email>
   <Abstract>
      <Problem></Problem>
      <Contribution></Contribution>
   </Abstract>
   <Introduction>
      <Problem></Problem>
      <Contribution></Contribution>
   </Introduction>
   <Conclusion>
      <Problem></Problem>
      <Contribution></Contribution>
      <FutureWork></FutureWork>
   </Conclusion>
   <Citation>
      <CitationTitle></CitationTitle>
      <CitationYear></CitationYear>
   </Citation>
</Document>
```

Fig. 1. Conceptual model of the domain and annotation schema

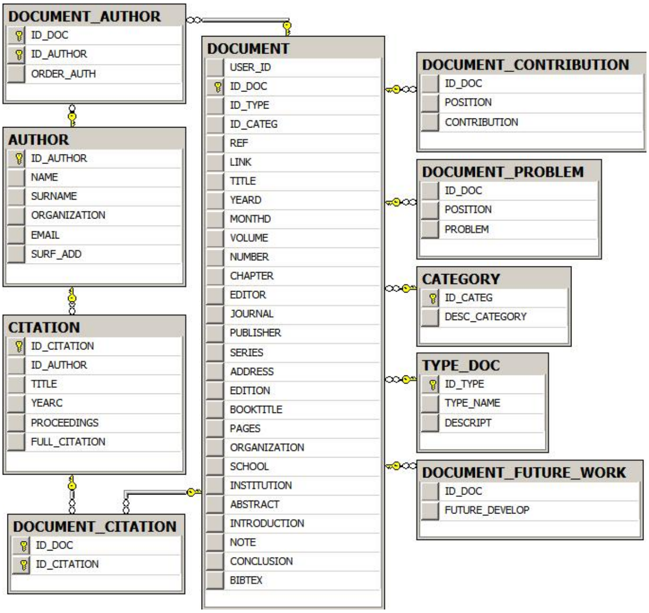


Fig. 2. Database schema

If input documents are PDF files we use a PDF text extractor tool to convert the documents into plain text files. For MS Word documents, we first convert input files into RTF format and then extract plain text from the converted files.

To address the problem of varied page layouts, we adapted Cerno, with a preprocessing module which flattens paragraphs and sentences in such way as to obtain one sentence per line.

Recognition of document structure is another challenge that we accommodated. In order to identify structural elements, we used a set of patterns combining position information and keywords. For instance, a pattern for identifying authors information takes into account both position within the document and syntactic representation of a text line, i.e., a line containing authors information normally is the sequence of names separated by commas and followed by email addresses. Note, that the pattern for each structural element is a generalization of several commonly used formats.

As for semantic elements, in Cerno they are extracted on the basis of the category wordlists yielded manually using entities of the conceptual model and context driven indicators. For example, the contribution information is certainly present in sentences having one of the following structures: “this work proposed…”, “the present paper analyzes…” and other similar phrases.

Finally, the issue of generating citation lines is also realized by reusing the information recognized in the documents of the user’s collection. This information is then used by the query engine when retrieving data from the database.

According to the Cerno’s method, document processing is then organized into three phases:

1. *Parse*: Lightweight parse of the document structure;
2. *Markup*:
    a. Structural markup of the main sections;
    b. Semantic markup of relevant facts in specific sections;
3. *Mapping*: Annotations are copied into the database.

The result of the first stage is a parse tree of the input text. A fragment of the grammar imposed on input documents is provided in Fig. 3. Essentially, the input is considered as the sequence of lines, among which we can find the headers of abstracts, introductions, and other sections. The tool recognizes such special lines, annotates them as the headers of appropriate sections and identifies the structural partitioning of the document based on these annotations.

```
 % input is defined as a sequence of zero or more lines
define program
  [repeat line]
end define
 % we want to distinguish between special lines, which are headers,
    and any other lines
define line
  [special_line]
 |[not special_line] [repeat token_not_newline] [opt newline]
end define
 % different types of the headers are specified
define special_line
 [abstract_line] |[introduction_line] |[conclusion_line]
 |[references_line]
end define
```

Fig. 3. A grammar fragment

When the structural markup is completed, the Cerno's engine semantically annotates phrases in the chosen sections, abstract, introduction and conclusions. Fig. 4 shows some indicators used in Bilbio in order to identify future work and contribution information. For the reasons of TXL compiler performance, we did not use a lemmatizer to handle different morphologic word forms, and instead we listed all word forms manually.

```
FutureWork : future work, future works, future development, future
research, future investigation, future investigations, in future, in the
future;
Contribution : this paper presents, this paper introduces, this paper
proposes, this paper shows, this paper demonstrates, this paper studies,
this paper explores, this paper investigates, this paper provides, this
paper describes, this paper discusses, <...>
```

Fig. 4. Indicative phrases for semantic annotation

The result of this stage is the text file normalized in its format and annotated with structural and semantic markups (Fig. 5) which is then passed on to the Biblio application for further analysis.

```
<Doc><Head><Title>ToMAS: A System for Adapting Mappings while Schemas
Evolve</Title><Authors>Yannis Velegrakis Renree J. Miller Lucian Popa
John Mylopoulos </Authors></Head>
<Citation> [1] P. Bernstein, A. Levy, and R. Pottinger. A Vision for
Management of Complex Models. SIGMOD Record, 29(4):55-63, December 2000.
</Citation>
       <...>
</Doc>
```

Fig. 5. An annotated text output

The user interacts with the Biblio GUI that allows to fully control the process of annotation, data acquisition and data manipulation. The prototype has been implemented in Borland Delphi.

The graphic user interface (GUI) of Biblio is composed of the following sections:

- *Document*: Shows the information identified by Cerno for the current document with extra fields related to identification of the document that the user can fill through the interface, e.g., page numbers, editors, conference venue, etc.
- *Citations*: Shows the citations from the current document, extracted by Cerno;
- *Author*: Allows management of the author information for the current document.
- *Advanced search*: An interface to retrieve records from the database by database query.
- *Import*: Allows selection of new papers, processing of them by the Cerno engine and upload of the extracted information to the user's database.
- *Export*: Allows export of the bibliographical data of the selected papers in different formats, such as MS Word and BibTex.

The *Document*, *Citations*, and *Author* sections are devoted to manage data details for each document loaded into the database. For each new document processed, a record with a unique identifier is created unless the database already contains the document with the same title and authors.

The Import section is the core of the application. It is composed by four components:

- *Pre-process component*: this module is responsible for the extraction of text from document. It uses different sub-program accordingly with the input document, e.g., for PDF document the PDFtoText tool is used for text extraction.
- *A normalize component*: this component is based on TXL parser and it allows to normalize the input text by uniforming the structure of sentences, paragraph.

- *The TXL component*: it annotates the document as described in section 3.2.
- *The database front-end:* it is responsible for import of the annotated text into the database, and verification of the consistency of data.

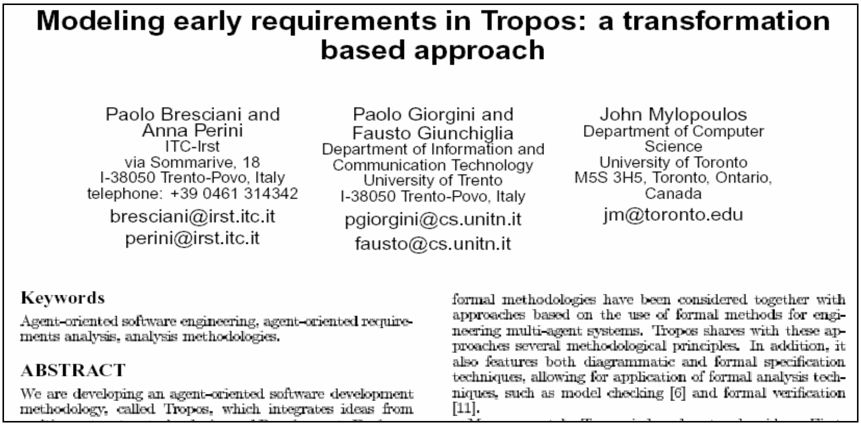
When structural and semantic annotation is completed, the relevant information is copied into a relational database. Business rules enforced on the database control the integrity of the data.

Finally, the *Export* section allows to control the export of data. The tool allows to manage different citation line generation.

4 A Running Example

In this section, we illustrate by example how Biblio processes documents.

When a user needs to write a paper, he collects a bunch of papers, and loads them into the system, see an input file example in Fig. 6.



Modeling early requirements in Tropos: a transformation based approach

Paolo Bresciani and
Anna Perini
ITC-Irst
via Sommarive, 18
I-38050 Trento-Povo, Italy
telephone: +39 0461 314342
bresciani@irst.itc.it
perini@irst.itc.it

Paolo Giorgini and
Fausto Giunchiglia
Department of Information and
Communication Technology
University of Trento
I-38050 Trento-Povo, Italy
pgiorgini@cs.unitn.it
fausto@cs.unitn.it

John Mylopoulos
Department of Computer
Science
University of Toronto
M5S 3H5, Toronto, Ontario,
Canada
jm@toronto.edu

Keywords

Agent-oriented software engineering, agent-oriented requirements analysis, analysis methodologies.

ABSTRACT

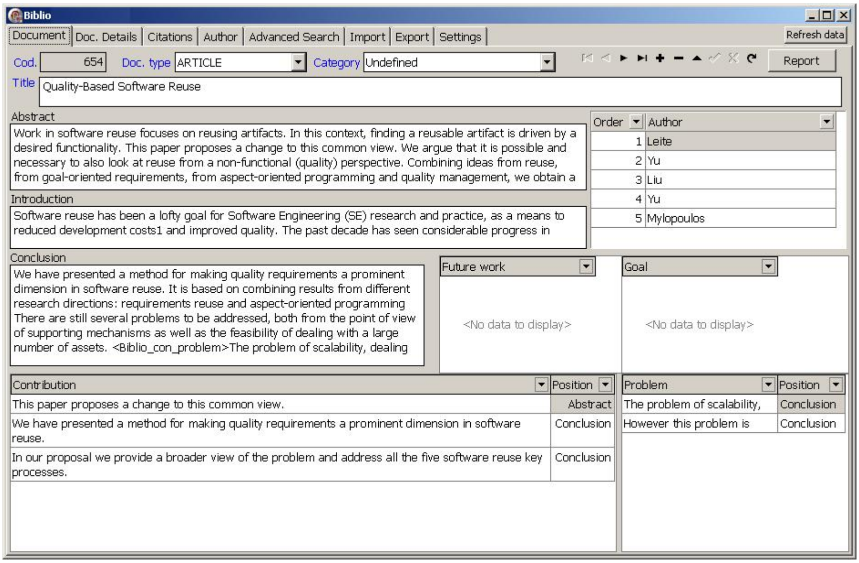
We are developing an agent-oriented software development methodology, called Tropos, which integrates ideas from

formal methodologies have been considered together with approaches based on the use of formal methods for engineering multi-agent systems. Tropos shares with these approaches several methodological principles. In addition, it also features both diagrammatic and formal specification techniques, allowing for application of formal analysis techniques, such as model checking [6] and formal verification [11].

Fig. 6. An input document

For the PDF format, the input document is passed to the PDFtoText tool (http://www.foolabs.com/xpdf/) to convert it to plain text format. In case of MS Word format, MS Office provides internal facilities to convert files into plain text. After that, the text document is passed to the Cerno's engine. The output of the processing is then handled by the interface to feed the database of bibliographical data. The records of this database are used for filling the fields of the user interface, as shown in Fig. 7.

In addition, for generating complete citations, the tool allows the user to manually add missing bibliographic information that cannot be found in the document body, such as the year of publication, journal, etc. The result can be exported to a file in MS Word or BibTEX format. For example, Fig. 8 shows a fragment of the generated bibliography.



Cod. 654 Doc. type ARTICLE Category Undefined

Title Quality-Based Software Reuse

Abstract
Work in software reuse focuses on reusing artifacts. In this context, finding a reusable artifact is driven by a desired functionality. This paper proposes a change to this common view. We argue that it is possible and necessary to also look at reuse from a non-functional (quality) perspective. Combining ideas from reuse, from goal-oriented requirements, from aspect-oriented programming and quality management, we obtain a

Introduction
Software reuse has been a lofty goal for Software Engineering (SE) research and practice, as a means to reduced development costs1 and improved quality. The past decade has seen considerable progress in

Order	Author
1	Leite
2	Yu
3	Liu
4	Yu
5	Mylopoulos

Conclusion
We have presented a method for making quality requirements a prominent dimension in software reuse. It is based on combining results from different research directions: requirements reuse and aspect-oriented programming There are still several problems to be addressed, both from the point of view of supporting mechanisms as well as the feasibility of dealing with a large number of assets. <Biblio_con_problem>The problem of scalability, dealing

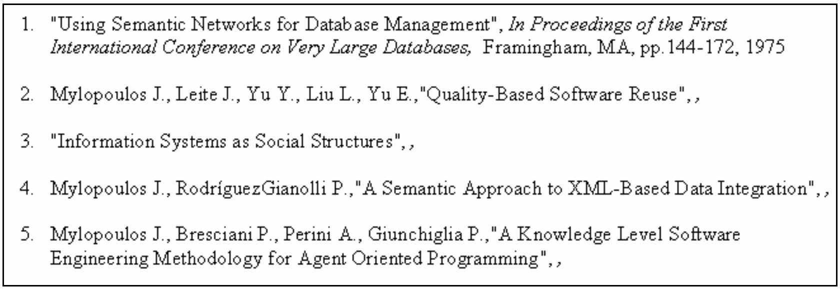
Future work
<No data to display>

Goal
<No data to display>

Contribution	Position
This paper proposes a change to this common view.	Abstract
We have presented a method for making quality requirements a prominent dimension in software reuse.	Conclusion
In our proposal we provide a broader view of the problem and address all the five software reuse key processes.	Conclusion

Problem	Position
The problem of scalability,	Conclusion
However this problem is	Conclusion

Fig. 7. GUI of Biblio


1. "Using Semantic Networks for Database Management", *In Proceedings of the First International Conference on Very Large Databases,* Framingham, MA, pp.144-172, 1975

2. Mylopoulos J., Leite J., Yu Y., Liu L., Yu E.,"Quality-Based Software Reuse",,

3. "Information Systems as Social Structures",,

4. Mylopoulos J., RodríguezGianolli P.,"A Semantic Approach to XML-Based Data Integration",,

5. Mylopoulos J., Bresciani P., Perini A., Giunchiglia P.,"A Knowledge Level Software Engineering Methodology for Agent Oriented Programming",,

Fig. 8. The generated citation list

5 Evaluation

In order to carry out a preliminary evaluation of our tool, we collected a set of 60 Computer Science publications. The articles were published in different years, on different events, in various formats and layouts.

In order to evaluate the quality of structural markup, we calculated the recall and precision rates of the markup produced by Cerno for the *Title* and *Author* fields against manual human opinions. This information is necessary for obtaining a citation line, and at the same time, can be found in the body of any published paper.

Recall is a measure of how well the tool performs in finding relevant items, while precision indicates how well the tool performs in not returning irrelevant items. In this evaluation, we also took into account partial answers, giving them a half score, as shown in formulas (1) and (2). The annotations are *partially correct* if the entity type is correct and the contents are overlapping but not identical.

$$Recall = (TP + \tfrac{1}{2}\,Partially\ Correct) / (TP + \tfrac{1}{2}\,Partially\ Correct + FN) \tag{1}$$

where TP – true positive answers, FN – false negative answers

$$Precision = (TP + \tfrac{1}{2}\,Partially\ Correct) / (TP + \tfrac{1}{2}\,Partially\ Correct + FP) \tag{2}$$

where TP – true positive answers, FP – false positive answers

The results of the evaluation, according to these quality measures, are shown in Table 1. All values are provided in percents.

Table 1. Evaluation results for most crucial structural annotations

	Title	Author
Recall	0.93	0.90
Precision	0.90	0.94

The estimated data demonstrate high quality rates in identification of structural sections of the electronic publications. However, some errors were observed. They were caused mainly by the erroneous output of the PDF converter for some documents.

To evaluate semantic annotations, we considered three concepts: *Contribution, Problem,* and *Future Work.* Manual creation of the reference annotation requires expensive domain expertise. In Table 2 we provide the evaluation for such elements.

Table 2. Evaluation results for semantic annotation.

	Contribution	Problem	Future Work
Precision	0.91	0.92	1.00
Recall	0.71	0.64	0.45

Observing the obtained estimates, we see that identification of Contribution, Problem and Future Work information was very accurate, 91, 92 and 100% respectively. Still, some false positive answers were detected. The Recall values are indicative of the need of improve patterns to identify the concepts. To resolve this problem, the Biblio domain dependent components can be improved in future.

Although, this evaluation is preliminary and can be extended to larger articles collections, the overall results suggest for appropriateness of using lightweight text processing techniques for the analysis of research publications in electronic format. With a small effort required to adapt the Cerno framework to the different domain, we achieved high performance rates for both structural markup and recognition of text fragments related to the semantic concepts.

6 Related Work

The need to extract bibliographical data was underlined by Cruz *et al.* in [9], where the authors proposed a methodology to create citation indices. A number of commercial and academic products for organization and storage of bibliographic data are already available, a detailed survey of existing tools can be found in [10]. Existing systems support the process of automatic extraction of citations from existing documents and the creation of personal or shared databases.

Noodle [11], WestCheck [12] and LawPro [13] are platforms providing support for extraction of citations from exiting document. This information is stored in a local or an on-line database, and then, citation lines in different styles can be generated.

Other academic projects, such as Citeseer [14,15], CiteBase [16], and AnaPort [17] are search engines that extract structural information and citations from on-line scientific documents. They support such document formats as PDF, MS Word, and PostScript. A citation line in different format is provided as output.

Another group of tools support automatic formatting of bibliographical data on the basis of user-defined annotations, such as on-line system for generation of citations CitationMachine [18] and Polaris on-line database which is developed and used at present at the University of Trento [19].

Our work relates well to the ShaRef project [20,21] that suggest the ShaRef tool allowing for both Web-based and offline creation, management, and sharing of reference information. This information can be exported to MS Word or Web browsers applications. The ShaRef approach is based on an XML data model. Nevertheless, no automation of the data extraction process is provided.

The proposed tool, Biblio, differs from earlier efforts particularly in extraction of semantic knowledge from documents. Biblio uses a combination of structural and lightweight semantic annotation to provide the user with information about the focus, main achievements and future work directions of the scientific documents. The tool is easily expandable to identify different information by changing the conceptual model and domain dependent inputs.

7 Conclusions

This paper addresses the problem of providing automatic support for authors who need to deal with large volumes of research literature for different purposes, such as creating a bibliography, constructing a personal bibliography, or managing a personal collection of publications.

To help the user in these tasks, we proposed Biblio, a novel user-friendly application for analysis of research publications in electronic format. The process is based on the lightweight semantic annotation Cerno. The tool allows the user to identify, visualize and store important system elements of the selected publications, and to automatically generate a bibliography from a set of chosen articles.

The tool is able to address the problems described in the introduction using a lightweight semantic annotation process. Different document formats are handled through a preprocessing phase. The document structure is captured using a combination of techniques based on mutual disposition of elements and syntax used to express such elements. Eventually Biblio can extract semantic information from documents by using the lightweight pattern-based Cerno’s analysis. The backend of Biblio is a standard relational database that can be used for querying and evaluating the system. A preliminary evaluation of structural and semantic annotations demonstrates the potential for high quality results using our method.

References

1. Kiyavitskaya, N., Zeni, N., Mich, L., Cordy, J.R., Mylopoulos, J.: Text Mining with Semi Automatic Semantic Annotation. In: Reimer, U., Karagiannis, D. (eds.) PAKM 2006. LNCS (LNAI), vol. 4333, pp. 143–154. Springer, Heidelberg (2006)
2. Kiyavitskaya, N., Zeni, N., Cordy, J.R., Mich, L., Mylopoulos, J.: Applying Software Analysis Technology to Lightweight Semantic Markup of Document Text. In: Proc. of Int. Conf. on Advances in Pattern Recognition (ICAPR , Bath, UK, 2005, pp. 590–600 (2005)
3. Google Scholar, http://scholar.google.com
4. DBLP, http://dblp.uni-trier.de/
5. Citeseer, http://citeseer.ist.psu.edu/
6. ACM Portal, http://portal.acm.org/dl.cfm
7. IEEE Digital Library, http://ieeexplore.ieee.org/Xplore/conhome.jsp
8. Cordy, J.R.: The TXL Source Transformation Language. Science of Computer Programming 61(3), 190–210 (2007)
9. Cruz, B.J.M., Lluch, O.J., Krichel, T., Pons, B.P., Velasco, S.E., Velasco, S.L.: INCISO: Automatic Elaboration of a Citation Index in Social Science Spanish Journals. AMETIST, no. 0 (September 2006)
10. Roth, D.L.: The emergence of competitors to the Science Citation Index and the Web of Science. Current Science 89, 1531–1536 (2005)
11. Noodle, http://www.noodletools.com/tools.html
12. WestCheck, http://www.westlaw.com
13. LawPro, http://www.llrx.com/
14. Bollacker, K.D., Lawrence, S.: Giles, Discovering relevant scientific literature on the Web. C.L. Intelligent Systems and Their Applications 15, 42–47 (2000)
15. Lawrence, S., Bollacker, K., Lee, C.: Giles "Indexing and retrieval of scientific literature". In: Proc. of International Conference on Information and Knowledge Management, CIKM99, pp. 139–146 (1999)
16. CiteBase, http://www.citebase.org/
17. AnaPort, http://www.ana-project.org/ref/
18. CitationMachine, http://citationmachine.net/index.php?source=11#here
19. Polaris on-line database, http://polaris.unitn.it
20. Wilde, E., Anand, S., Bücheler, T., Jörg, M., Nabholz, N., Zimmermann, P.: Collaboration Support for Bibliographic Data. International Journal of Web Based Communities, 3(1) (2007)
21. ShaRef project, http://dret.net/projects/sharef/

A New Text Clustering Method Using Hidden Markov Model*

Yan Fu[1], Dongqing Yang[1], Shiwei Tang[2], Tengjiao Wang[1], and Aiqiang Gao[1]

[1] School of Electronics Engineering and Computer Science, Peking University,
Beijing 100871, China
{fuyan,dqyang,tjwang,aqgao}@db.pku.edu.cn

[2] National Laboratory on Machine Perception, Peking University,
Beijing 100871, China
tsw@db.pku.edu.cn

Abstract. Being high-dimensional and relevant in semantics, text clustering is still an important topic in data mining. However, little work has been done to investigate attributes of clustering process, and previous studies just focused on characteristics of text itself. As a dynamic and sequential process, we aim to describe text clustering as state transitions for words or documents. Taking K-means clustering method as example, we try to parse the clustering process into several sequences. Based on research of sequential and temporal data clustering, we propose a new text clustering method using HMM(Hidden Markov Model). And through the experiments on Reuters-21578, the results show that this approach provides an accurate clustering partition, and achieves better performance rates compared with K-means algorithm.

1 Introduction

Since there is a large and continually growing quantity of electronic texts, text mining has been investigated in many different fields, especially in information retrieval [1]. Among the text mining task, text clustering is a very important part. The goal for this kind of clustering is to divide a set of text items into homogeneous groups, and precisely distinguish truly relevant information. However, involving in high dimensional spaces, text clustering is more interesting and challenging than ordinary data clustering.

The standard methods for text clustering are mainly based on VSM (Vector Space Model). VSM is a way of representing documents through the words that they contain. In this model, each document is considered as a vector in the term-space (set of document "words"). So the similarity between two documents can be measured with vector distance (e.g. Euclidean distance, Manhattan distance).

* The work described in this paper is supported by Project 60473051 under National Natural Science Foundation of China (NSFC), Project 60503037 under National Natural Science Foundation of China (NSFC), and 2006AA01Z230 under the National High-tech Research and Development of China.

Z. Kedad et al.(Eds.): NLDB 2007, LNCS 4592, pp. 73–83, 2007.

Then we can apply classical clustering method to achieve classificatory result [8]. We draw the conclusion that previous text clustering researches are based on text information, and the information are just described with "static" features. In this paper, we define "static" features as features with unchanged value (e.g. words or phrases in a document). In contrast, "dynamic" features describe features with changes. For example, if we define the cluster label as "state" for a document in clustering process, the "state" may change constantly, and we take "state" as a "dynamic" feature.

According to the actual situation, text clustering is a dynamic and step-by-step process. In traditional text clustering algorithms (e.g. K-means), the cluster label for a document always changes during clustering process. That is because clusters distribution changes continuously, in other words, relationships between documents and clusters keep on altering. These changes do correct clustering result continually. At the end of clustering process, documents with similar semantic content will be collected in one cluster. Then we present our basic hypothesis that, cluster labels for same cluster documents(i.e., documents belonging to the same cluster) may have the same or similar change trends. So, we aim to find a new method to deal with these dynamic processes, accompanied with the static text attribute.

To make use of the "dynamic" nature with "static" text attribute, based on traditional K-means method, we propose a novel approach using HMM for text clustering. In previous research, HMM is widely used for sequential or temporal data clustering. So we try to map text clustering process as sequences, and values along these sequences change continuously. Then we use text "state" transition to simulate the clustering flow. In our method, each text copy(i.e., document) is represented by a row vector using VSM. Items of the vector represent whether a word is presented in the document, and have their own transformation sequences. We take all words effects of one document together, and predict which cluster should the document belong to. To verify performance of this approach, we test it in Reuters-21578 and compare with the original K-means algorithm.

The rest of this paper is organized as following. Section 2 briefly introduces the Hidden Markov Model. In Section 3, we describe related works and our main idea for text clustering with HMM. Section 4 is our approach proposed detailedly. Subsequently, Section 5 is the experiment report and evaluation. We summarize the contributions of this paper and outline some directions for future work in Section 6.

2 Hidden Markov Model

Hidden Markov Model is a statistical modeling approach. In recent years, this model has steadily gained in popularity in temporal sequence analysis for bioinformatics and web traffic analysis, even plays an important role in speech segmentation [2] and speaker clustering [3]. These owe to its ability to capture dynamic properties of ordered observations. In this section, we describe the general HMM in detail.

In probability theory, when both transitions between states and generation of output symbols are governed by probability distributions, Hidden Markov Model can be viewed as stochastic generalizations of finite-state automata [4]. Transitions among states are called transition probabilities. In a particular state, an output or observation can be generated according to associated probability distribution. It is only the output, not the state, is visible to an external observer and therefore states are "hidden" to the outside; hence the name Hidden Markov Model [5].

In general, a HMM is composed of a five-tuple:

$$\lambda = (S, O, \pi, A, B)$$

1. $S = \{1, \ldots, N\}$ is the set of states. The state at time t is denoted S_t.
2. $O = \{O_1, \ldots, O_t\}$ is the observation sequence.
3. $\pi = \{\pi_i\}$ is initial state distribution:

$$\pi_i = P\left[q_1 = S_i\right] \quad \pi_i \geq 0, \sum_{i=1}^{N} \pi_i = 1, 1 \leq i \leq N$$

4. $A = \{a_{ij}\}$ is the state transition matrix:

$$a_{ij} = P\left[q_{t+1} = S_j | q_t = S_i\right] \quad 1 \leq i, j \leq N$$

and

$$a_{ij} \geq 0 \quad 1 \leq i, j \leq N$$

$$\sum_{j=1}^{N}(a_{ij}) = 1 \quad 1 \leq i \leq N$$

5. $B = \{b_j(k)\}$ is the probability distribution of observing k under j state.

For convenience, HMM is always denoted as a triple $\lambda = (\pi, A, B)$. The model definition can be readily extended to multi-dimensional case, where a vector of symbols is emitted at each step, instead of a single symbol.

There is an outstanding property of the Markov model that, given the present state, the conditional probability distribution of future state depends only upon the current state, i.e., it is irrelevant with historical states. Mathematically, the Markov property is described as follows [6]:

$$P\left(X_{t+1} = s_k | X_1, \ldots, X_t\right) = P\left(X_{t+1} = s_k | X_t\right)$$

3 Overview of Text Clustering Using HMM

3.1 Related Works

Clustering using HMM was first studied by Rabiner et al. [4,12], for speech recognition problem. Subsequently, the model was gradually applied to the problem of statistical modeling and multiple sequence alignment. Such as speech recognition, handwriting character recognition, DNA and protein modeling, gesture recognition, behavior analysis and synthesis, and more in general, to computer vision problem [7]. As HMM has been successfully employed for clustering sequential or temporal data, all these methods should be named as proximity-based clustering.

Previous research about HMM-based clustering is to compute similarities between sequences, using different approaches. The standard proximity-based method for clustering sequential data , using HMMs, can be summarized by the following algorithm.

Consider a given set of N sequential data $O_1...O_N$ to be clustered; the algorithm performs the following steps:

1. Train one HMM λ_i for each sequence O_i.
2. Compute the distance matrix $D = D(O_i, O_j)$, representing a similarity measure between data or between models; this is typically obtained from the forward probability $P(O_j \mid \lambda_i)$, or by devising a measure of distances between models.
   In the past, few authors proposed approaches to compute these distances. Early approaches were based on the Euclidean distance of the discrete observation probability, others on entropy, or on co-emission probability of two models, or, very recently, on the Bayes probability of error.
3. Use a pairwise distance-matrix based method (e.g. an agglomerative method) to perform clustering.

3.2 Main Idea

From view of semantics, words or phrases information determines that according document should be grouped into which cluster. So we define that, when a document is grouped to certain cluster at a time, the cluster label is document "state" at that time. At the same time, words contained in that document have the same "state". We assume word state transition has Markov property, and may be viewed as the result of a probabilistic walk along a fixed set of states. When states can be defined directly using feature values, a Markov chain model representation may be appropriate. While the set of states and the exact sequence of states may not be observed, they can be estimated based on observable behavior of dynamic system. If the state definitions are not directly observable, or it is not feasible to define using exhaustive method, they can be defined in terms of feature probability density functions. This corresponds to the hidden Markov model methodology.

Based on discussion above, the basic assumption underlying our method is that, each word has "state" transition during clustering process, and the transition is unknown but observable. The same words in different documents may have different transition sequences, those are what we can get from training process. In other words, various transition sequences compose of a mixture of

HMMs [18]. The composite HMM consists of several components, whereas these components is no "crossover". That is to say, sequence is assumed to come from one component or the other. Some sequences for one word can be generated with various model parameters. With trained parameters, we should accomplish the word pattern matching, or named sequence clustering, for document clustering.

4 Proposed Method

The main flow of our method is shown in Fig. 1, and detailed description is listed as follow:

1. Text analysis - We adopt vector space model to describe each text copy(i.e., document). Then documents are transformed to high-dimension vectors, and each item is corresponding to a single word. We primarily use single words in this article, since previous work indicates that this is the most effective way.

2. Training the HMM - In order to build model for words, we try to learn model parameters using standard HMM estimation techniques. Since our aim is to obtain a generative model for the data, likelihood maximization is exactly what we want to do. K-means and EM algorithm are employed for this purpose.

3. HMM pattern matching - Since words have different possible transition sequences, the next "state" for one word is unknown for us. So we do the matching for current "state" against relevant HMMs, to find possible transition for a word. When we take all words effects in a document into account, we can determine the final cluster label for this document.

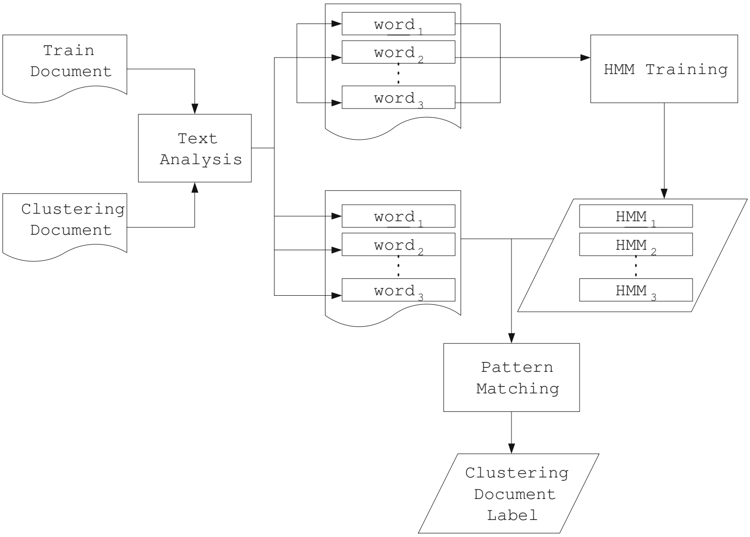


Fig. 1. Processing Flow of Text Clustering with HMM

4.1 Text Analysis

Fig. 2 shows the form of HMM used to characterize word state transitions. State transitions of a word are handled by a switch mode, from the current "state" of certain word to the next "state". The model is a first order, left-to-right, Markov model. Using vector space model, we convert documents into vectors with uniform length.

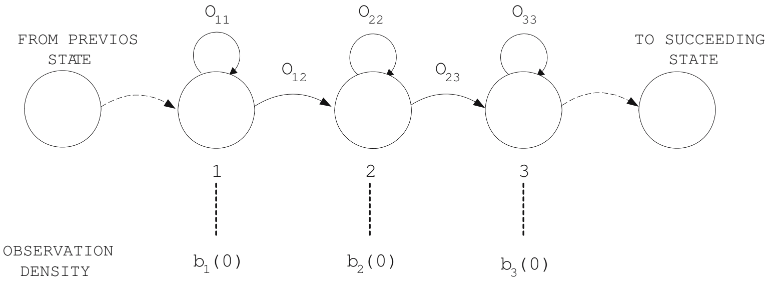


Fig. 2. Form of HMM Used to Characterize Individual Words

4.2 HMM Training

After pre-processing, documents have been transformed into real-valued vectors, and items of these vectors are corresponding to distinct words. Each word makes a contribution to document's attribution, so the document "state" transition matrix A is composed of words transition matrix. The transition matrix for a word is a $K \times K$ matrix, to record possible transfer sequences of the word(K is number of clusters). Accordingly, observation probability matrix B becomes a $N \times K$ matrix(N is the number of distinct words). Each item of B describes that if a word is presented in a document, the probability that it would be grouped to cluster K_i arouse by this word. In our algorithm, the number of states, i.e., the cluster number K, is set in advance and not learned from the data.

This step tries to find all of the possible sequences for a word, and relevant possibility parameters for mixture HMMs. Firstly, randomly choose certain percentage of documents as training set, start with the K-means algorithm for initialization. Denote K documents as the initial center of each cluster, let them be numbered from 1 to K. Then assign cluster number of the nearest cluster to each of the training document, re-calculate mean μ_i matrix for each cluster.

$$\mu_i = \frac{1}{D_i} \sum_{doc_t \in i} doc_t \quad 1 \leq i \leq K$$

where D_i is the number of documents in cluster i, and doc_t is a document in cluster i at time t. In our algorithm, we take time interval $t_n - t_{n-1}$ (n=1,2,...)

as one step in clustering process, and $\mu_i (i = 1, 2, \dots, K)$ represents the new cluster center.

The core of the parameter reestimation procedure is EM algorithm [13]. EM algorithm iterates between two steps: (i) the expectation step(E-step), and (ii) the maximization step(M-step). The EM procedure is repeated until the difference between the likelihood of the two consecutive transitions is less than a certain threshold. Like other maximum likelihood methods, this procedure may end up in local maximum values. At the end of training step, we can get various transition sequences of all words with different possibilities.

In our training procedure, the most important part is to calculate π, A and B in HMM model. Since our document vectors are composed of word items, and each item in vectors makes contribution to state transition. So every state transition for a document is corporate effects of all words. For each word in a document vector, it has its own A and B. As far as the document is concerned, we take into account of all words impact. It is the same for observation sequence. The calculation of π, A and B is defined as:

$$\pi_i = \frac{\text{Number of occurrences of} \quad doc_1 \in i}{\text{Number of occurrences in all training documents}} \quad 1 \le i \le K$$

$$a_{ij} = \frac{\text{Number of occurrences of} \quad doc_t \in i \quad \text{and} \quad doc_{t+1} \in j}{\text{Number of occurrences in all training documents}} \quad 1 \le i,j \le K$$

$$b_j(i) = \frac{\text{Number of occurrences of} \quad doc_t \in i \quad \text{and} \quad doc_t = j}{\text{Number of occurrences of} \quad doc_t \in i} \quad 1 \le i,j \le K$$

where t means time of a certain step in process, $doc_t \in i$ and $doc_{t+1} \in j$ represents document state transition in one step, $doc_t \in i$ and $doc_t = j$ means the observation is $doc_t \in i$ with the real document state is j.

4.3 HMM Pattern Matching

Once HMMs are trained, we can go ahead with text clustering. Initialize vectors for test documents with the same state probability, *1/K*. Then we can get the first observation state O_1. The algorithm to calculate the observation sequence is described below.

After initialization and training, we obtain the transition matrix and HMMs for all words. Transform the clustering documents to vectors. Then for each item of the vector, under some "state", it has several possible transition sequences. HMMs are employed to compute similarities between sequences and model segments. The output of the HMM pattern matcher is a set of candidate sequences, ordered by likelihood score. We choose the sequence with maximal possibility and do this step iteratively, until the "state" does not change anymore, that is, the final cluster label for the document.

Algorithm 1. Calculate Observation with Given HMM

```
 1: for all $document \in testset$ do
 2:     $t = 1$;
 3:     $\pi_1 = 1/K, \ldots, 1/K$;
 4:     Choose an initial state $S_t$ for document;
 5:     $O_t = S_t \cdot b_j(t)$;
 6:     $S_{t+1} = O_t \cdot a_{t(t+1)}$;
 7:     $t=t+1$;
 8:     if $t < T$ then
 9:         goto line 5;
10:     end if
11: end for
```

5 Experimental Results and Evaluations

This section presents our experiments performed on Reuters-21578, which consists of 21578 documents. It contains 135 so-called topics. To be more general, we would refer to them as “clusters”. For allowing evaluation, we restrict ourselves to the 12344 documents which have been classified manually by Reuters. Reuters assigns some of its documents to multiple classes, but we consider only the first assignment. We adopted C++ as the programming language, and experiments were performed on a PC with 2GHz CPU and 512MB of main memory.

Accuracy is the main quality concern for text clustering. Since we already know the real clusters, it is much easier to obtain the clustering quality measurement, Entropy. The definition of Entropy is specified as below.

$$E_j = -\sum_i P_{ij} \log \left(P_{ij}\right)$$

$$E_{cs} = \sum_j \frac{n_j \times E_j}{n}$$

where P_{ij} is the probability for document with cluster number i divided to cluster j , n_j is the document amount in cluster j , and n is the total count of documents. The smaller the E_{cs}, the better the clustering result. Then we randomly pick out 4 groups of data with diverse size, covering several topics, and list the average result of tests in Table 1. Since we make use of K-means method for initialization, here we compare our method with classical K-means and list the differences.

Data scope and a high number of clusters have also been shown to be troublesome aspects for clustering algorithm. So in the following set of experiments, we analyze the algorithm behavior when data scope and cluster number are various.

The second experiment is to evaluate the correlation between clustering performance with test set size, or desired cluster number. In common sense, with the growing size of test set, increment of algorithm runtime is unpredictable.

Table 1. Entropy value to describe accuracy of clustering result. "Data set size" is the actual documents amount for clustering. "Entropy of K-means" and "Entropy of HMM" respectively indicate the Entropy calculated in K-means and our method.

Data set number	Data set size	Entropy of K-means	Entropy of HMM
1	1180	0.1424	0.1766
2	1514	0.2032	0.2011
3	1981	0.1839	0.2103
4	2754	0.1647	0.1820

But as result shown in our article, with the continual increase in test documents quantity, runtime raise is close to linearity. This experiment report is presented in Fig. 3. To inspect adaptability of our algorithm, we artificially enhance the desired cluster number continually on five hundred of documents. This result can be observed in Fig. 4.

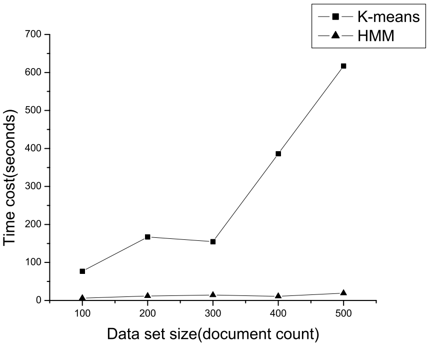


Fig. 3. Performance with varying size

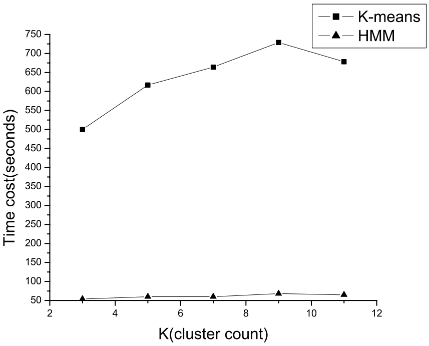


Fig. 4. Performance with varying cluster number

After a series of practical experiments, considering the tradeoff between quality and performance, we can discover that our algorithm behaves fine in various conditions, even better than classical K-means under some circumstance.

6 Conclusion and Future Work

In this paper, we address the problem of unsupervised text clustering using the HMM. Once we take temporal cluster label for a document as its "state", we can find that the "state" always changes during clustering process. In a set of documents, state transition sequences for the same word are different, and the next "state" is only determined by current "state". So we can construct HMM for each word to simulate "state" transitions. Based on classical K-means method, we train HMM parameters for each word in a completely unsupervised manner. Finally, we take into account HMM for all words, and judge which cluster should documents belong to. Tested on Reuters-21578, our algorithm is generally less computational demanding than classical K-means algorithm. In general, the way to combine static and dynamic nature for text clustering is feasible, and sometimes a better choice.

But there are still some improvements of our method for future work. First of all, effect of HMM in our algorithm is dependent on model initialization. Presently, we adopt classical K-means to initialize parameters in HMM. However, there are some intrinsic disadvantages in K-means algorithm. Such as the selection of original cluster center, and confirmation of K. Therefore, we would attempt to search for better method for initializing. And further, the desired number of clusters K, is the only parameter to appoint in advance, we would investigate how to determine the final number of clusters automatically. Secondly, since the training algorithm converges to local maximum, the resulting HMM could not be global optimal. That is to say, clustering result is probably local optimal and could not satisfy requirement of users. Thirdly, we only accomplish comparison between our method and K-means algorithm. Such experiments are insufficient and closely dependent on experimental conditions. In future work, we would implement more text clustering methods, such as AHC(Agglomerative Hierarchical Clustering) [8] and STC(Suffix Tree Clustering) [20] etc., for performance comparison.

References

1. Steinbach, M., Karypis, G., Kumar, V.: A Comparison of Document Clustering Techniques. In: Proc. TextMining Workshop, KDD 2000 (2000)
2. Ajmera, J., McCowan, I., Bourlard, H.: Robust HMM based speech/music segmentation. IEEE International Conference on Acoustics, Speech, and Signal Processing (2002)
3. Ajmera, J., Bourlard, H., McCowan, I.: Unknown-multiple speaker clustering using HMM. International Conference on Spoken Language Processing (2002)
4. Rabiner, L.R.: A Tutorial of Hidden Markov Models and Selected Applications in Speech Recognition. Proc. of IEEE 77(2), 257–286 (1989)
5. Rabiner, L.R., Juang, B.H.: An introduction to hidden Markov models. IEEE ASSP Magazine 3(1), 4–16 (1986)
6. Manning, C.D., Schutze, H.: Chapter 9: Markov Models. In: Foundations of Statistical Natural Language Processing, Papers in Textlinguistics, pp. 317–379. The MIT Press, Cambridge (1999)
7. Panuccio, A., Bicego, M., Murino, V.: A Hidden Markov Model-Based Approach to Sequential Data Clustering. LNCS, pp. 734–742. Springer, Berlin (2002)
8. Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann, San Francisco (2000)
9. Cutting, D.R., Karger, D.R., Pedeson, J.O., Tukey, J.W.: Scatter/Gather: a cluster-based approach to browsing large document collections. In: Proceedings ACM/SIGIR, pp. 318–329 (1992)
10. Zamir, O., Etzioni, O.: Web Document Clustering: A Feasibility Demonstration. In: Proc. ACM SIGIR 98 (1998)
11. Berkhin, P.: Survey of Clustering Data Mining Techniques, Technical report, Accure Softward, San Jose, CA (2002)

12. Rabiner, L.R., Lee, C.H., Juang, B.H., Wilpon, J.G.: HMM Clustering for Connected Word Recognition. In: Proceedings of IEEE ICASSP, pp. 405–408. IEEE Computer Society Press, Los Alamitos (1989)
13. Dempster, A.P., Laird, N.M., rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm, JRSS-B (1977)
14. Buckley, C., Lewit, A.F.: Optimizations of inverted vector searches, SIGIR '85, pp. 97–110 (1985)
15. van Rijsbergen, C.J.: Information Retrieval, Buttersworth, London, 2nd edn. (1989)
16. Dubes, R.C., Jain, A.K.: Algorithms for Clustering Data. Prentice Hall, Englewood Cliffs (1988)
17. Rocchio, J.J.: Document retrieval systems - optimization and evaluation. Ph.D. Thesis, Harvard University (1966)
18. Smyth, P.: Clustering Sequences with Hidden Markov Models, pp. 648–654 NIPS (1996)
19. Oates, T., Firoiu, L., Cohen, P.R.: Clustering Time Series with Hidden Markov Models and Dynamic Time Warping. In: Proc. of the IJCAI-99 Workshop on Neural, Symbolic and Reinforcement Learning Methods for Sequence Learning, pp. 17–21 (1999)
20. Zamir, O., Etzioni, O.: Web Document Clustering: a Feasibility Demonstration. In: Proceedings of the 19th International ACM SIGIR Conference on Research and Development of Information Retrieval (SIGIR'98), Melboume (1998)

Identifying Event Sequences Using Hidden Markov Model

Kei Wakabayashi and Takao Miura

Dept.of Elect.& Elect. Engr., HOSEI University
3-7-2 KajinoCho, Koganei, Tokyo, 184–8584 Japan

Abstract. In this paper, we propose a sophisticated technique for classification of topics appeared in documents. There have been many investigation proposed so far, but few investigation which capture contents directly. Here we consider a topics as a *sequence* of events and a classification problem as segmentation (or tagging) problem based on *Hidden Markov Model* (HMM). We show some experimental results to see the validity of the method.

Keywords: Topic Classification, Hidden Markov Model.

1 Introduction

Recently there have been a lot of knowledge-based approaches for documents available through internet, and much attention have been paid on classification techniques of documents. Generally in the classification we assume each document is translated into *vector* and apply *cosine-based* similarity based on vector-space model. However, it is hard to identify topics by means of this approach because we construct vectors by using words but not sentences. Our main purpose is targeted for *classification of topics*.

One of the typical approach to tackle with the problem is *Topic Detection and Tracking* (TDT) [1]. In TDT, a topic is characterized as a sequence of *events*. By an event we mean a fact or an individual affair that can be identified in a spatio-temporal manner. Event tracking tasks in TDT contain classification of documents describing events[11]. In Makkonen[7], they define a topic as a sequence of events that may fork into several distinct topics. They attempt to detect topics by connecting each events to other similar events.Our interest is, however, in similarity of each sequences.

In this investigation, we discuss a classification technique of topics. The main idea comes from topic configuration based on *Hidden Markov Model* (HMM) considering each topic as a sequence of events. There have been several approach for classification such as Decision Tree, Support Vector Machine, Self Organizing Map (SOM) and naive Bayesian[6] by which we model each topic as a vector or a collection of words (and the frequency), but not as a sequence. Thus it is hard to apply the techniques to topic classification in a straightforward manner.

Here we propose a new and sophisticated technique for the classification of topics in documents by using stochastic process. In Barzilay[3], they discuss how

Z. Kedad et al.(Eds.): NLDB 2007, LNCS 4592, pp. 84–95, 2007.

to model documents using stochastic process approach. Especially they estimate latent structure of documents by HMM with 2-gram syntax as output assuming domain specific knowledge. Compared to the approach, we put our focus on estimating *topic structure* and there is no direct relationship.

Topic Segmentation is another approach to consider each document as sequences of events without any explicit boundaries such as news broadcasting. HMM has been considered as a useful tool for segmenting the transcriptions. For example, in [9], each event corresponds to a state in HMM over collections of words and the issue is to examine whether we move to another state or not. In [4], they have extended this approach by means of an *aspect model*, called AHMM, where they put labels focusing on word distribution at each state.

Another approach is found in [10]. They examine what's going on in specific documents (transcription of cooking courses on TV program) based on stochastic process (HMM). They estimate every step in a cooking course using HMM, i.e., what kind of the art of cooking they are standing on. However they focus on single topic (cooking single cuisine) and no general discussion on concurrent multiple topics.

In this work, we discuss classification issue of topics in section 2. In section 3 we review Hidden Markov Model quickly and we develop the classification technique using HMM in section 4. Section 5 contains experimental results. We conclude our discussion in section 6.

2 Identifying Event Sequences

First of all, let us describe how we identify event sequences considered as topics. As we said, an *event* means a fact or an individual affair that can be identified in a spatio-temporal manner while a *topic* corresponds to a subject matter (or a theme). For example, we see a series of affairs described as news articles along with a common subject, then each affair is an event and the subject a topic. Let us note that a sequence of events don't always correspond to any topic, just same as news-casting.

It is hard to decide whether two topics are *similar* with each other, and a matter of taste very often. In this investigation, however, we define two topics are similar if two sequences of events are similar. For example, given two different murder cases in Tokyo and Kyoto, we can say they are similar if the sequences of the affairs are the similar, i.e., if they killed someone in distinct cities, they were wanted and arrested by police.

Let us illustrate our approach in a figure 1. Assume a document contains some topic of murder. Then we give a series of all the first paragraphs of news articles about common topic arranged in a temporal order. Here we have all the first paragraphs of the articles about some murder case concatenated into one in temporal order.

In the figure 1, we show a sequence of events. By going through the articles, we see an event that some guy was found dead. Then police examines and finds some hints about a suspicious person, and finally the person is arrested. These are all the events that constitute the topic of this *murder*. Clearly the topic sequence depends on the characteristics of the story *murder*. For instance, in a case of a suicidal act, we might have a different sequence: some guy was found dead but considered as a suicide because of a note left behind, then the person is traced to the origin. In a case of murder, the suspicious might commit suicide but it is not common as a suicidal act since some other guy had been found dead. We can say that the two sequences are *not* similar with each other, and that a murder case carries its own pattern of an event sequence (or sequences).

With the consideration that each topic carries own pattern of event sequences, we believe we can estimate topics in documents and identify them.

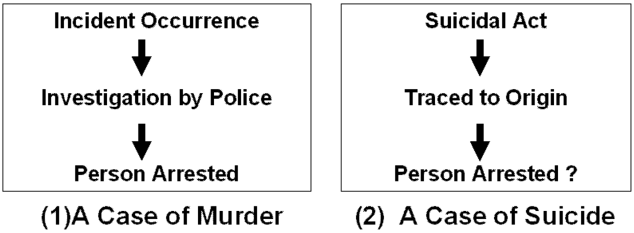


Fig. 1. Estimating Topic in a Document

3 Hidden Markov Model

A *Hidden Markov Model* (HMM) is nothing but an automaton with output where both the state transition and the output are defined in a probabilistic manner. The state transition arises according to a simple Markov model but it is assumed that we don't know on which state we are standing now[1], and that we can observe an output symbol at each state. We could estimate the transition sequences through observing output sequence.

3.1 Defining HMM

A HMM model consists of (Q, Σ, A, B, π) defined below[6]:

(1) $Q = \{q_1, \cdots, q_N\}$ is a finite set of states

(2) $\Sigma = \{o_1, \cdots, o_M\}$ is a finite set of output symbols

(3) $A = \{a_{ij}, i, j = 1, ..., N\}$ is a probability matrix of state transition where each a_{ij} means a probability of the transition at q_i to q_j. Note $a_{i1} + ... + a_{iN} = 1.0$.

(4) $B = \{b_i(o_t), i = 1, ..., N, t = 1, ..., M\}$ is a probability of outputs where $b_i(o_t)$ means a probability of an output o_t at a state q_i

(5) $\pi = \{\pi_i\}$ is an initial probability where π_i means a probability of the initial state q_i

[1] This is why we say *hidden*.

In this work, each state corresponds to an event type such as "arresting a suspicious person" and "found dead", and the set of states depends on the contents of topics. Output symbols (some words appeared in documents) should be observable and identifiable in our case.

The probability matrix A shows the transition probability within a framework of simple Markov model, which means state change arises in a probabilistic manner depending only on the current state. Thus, for instance, the (i, j)-th component of A^2 describes the transition probability from q_i to q_j with two *hops* of transitions. Similarly the output appears depending only on the current state.

3.2 Estimating State Transition

HMM is suitable for estimation of *hidden* sequences of states by looking at observable symbols. Given a set of several parameters, we can obtain the state sequence which is the most likely to generate the output symbols. The process is called a *decoding problem* of HMM, and among others, *Viterbi* algorithm is one of the well-known solutions for the decoding problem.

Here we define the most likely sequence of states as the one by which we obtain the highest probability of the output generation during the state transition. The procedure is called *Most Likelihood Estimation* (MLE). In the procedure, once we have both sequences of the states and the output symbols, we can determine the probabilities (or *likelihood*) of the state transition and of the output generation along with the state transition. Putting it more specifically, when we have the state transition $q_1 q_2 \cdots q_T$ and the output sequence $o_1 o_2 \cdots o_T$, we must have the likelihood as below:

$$\pi_{q_1} b_{q_1}(o_1) \times a_{q_1 q_2} b_{q_2}(o_2) \times \cdots \times a_{q_{T-1} q_T} b_{q_T}(o_T)$$

Given the output sequence $o_1 \cdots o_T$, a Viterbi algorithm is useful for obtaining the most likely sequence of the states by taking the highest likelihood $\delta_t(j)$ of an output o_t at a state q_i to go one step further to q_j. That is, the algorithm goes recursively as follows:

$$\delta_{t+1}(j) = \max_i(\delta_t(i) a_{ij}) b_j(o_{t+1})$$

During the recursive calculation, we put the state q_j at each time t, and eventually we have the most likelihood sequence $q_1 \cdots q_T$.

3.3 Estimating HMM Parameters

A HMM consists of (Q, Σ, A, B, π) where we should give a set of states Q and a set of output symbols Σ in advance. On the other hand, it is hard to determine definitely the transition probability matrix A, the output probability B and the initial probability π, thus we should think about this issue, how to obtain them. This problem is called a *model calculation* of HMM. Usually we do that by means of some machine learning techniques[8].

One of the typical approach is *supervised learning*. In this approach, we assume *training data* in advance to calculate the model, but the data should be correctly classified by hands since we should extract typical patterns them by examining them. Another approach comes, called *unsupervised learning*. Assume we can't get training data but a mountain of unclassified data except a few. Once we obtain strong similarity between the classified data and unclassified data (such as high correlation), we could extend the training data in a framework of Expectation Maximization (EM) approach[5].

One of the typical approach is known as a *Baum-Welch* algorithm. The algorithm has been proposed based on EM approach. That is, the algorithm adjusts the parameters many times to maximize the likelihood for the generation of the output symbols given as unsupervised data. The process goes just same as EM calculation, i.e., we calculate the expect value of the transition probability and the output probability, then we maximize them. We do that until few change happens.

Formally, let $\overline{\pi}_i$ be the expected frequency of the initial state q_i. Then we can get a new transition probability $\overline{a}_{ij}$

$$\frac{\text{Expected Transition Frequency } q_i \rightarrow q_j}{\text{Expected Total Transition Frequency from } q_i}$$

And also we can obtain a new output probability $\overline{b}_i(k)$

$$\frac{\text{Expected Frequency at } q_i \text{ with an output } k}{\text{Expected Frequency at } q_i}$$

We repeat the calculation process many times until we get to the convergence. One of the well-known problems is that the results may come to the local maxima and they depend on the initial parameters.

4 Estimating Topics

In this section let us develop our theory to estimate topics in documents using HMM.

4.1 Applying HMM

Let us restate out problem of topic estimation in terms of HMM. We illustrate our approach in a figure 2 where an event corresponds to a state and a sentence to an output symbol. Remember a topic consists of events. And estimating topics means obtaining the most likely sequence of the states. We can obtain the sequence by using Viterbi algorithm.

However, it is not easy for us to consider a whole sentence as an output since we must have a variety of words and the expressions over them while there are many non-topic-relevant words such as "apartment", "stolen goods" and "hanging" as well as many function words such as "in", "with" and "however". In fact, the latter words are not really useful to estimate topics.

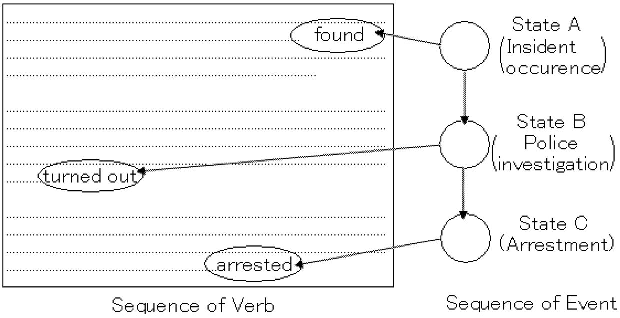


Fig. 2. Estimating Topics

We put our attention on particular parts in a document, i.e., we extract specific parts to capture *change of circumstances*. In this work, we examine a *predicate* part as the possible candidate. In any documents in Japanese, the predicate appears as a *last verb* in each sentence. We apply *morphological analysis* by using ”Chasen”[2], extract the predicate parts and give them as the output symbols as shown in the figure.

Each topic consists of various kinds of events depending on the contents. For example, in a case of ”murder” topic, there are several events such as ”suspicious person arrested” and ”some guy found dead”, while, in another case of topic such as ”suicidal act”, we have different events such as ”a note left behind”. This means there can be several collections of states Q depending upon topics. This is the reason why we need many kinds model calculation for HMM.

Before proceeding, let us define some words to avoid confusion in our investigation. A *document* is a sequence of sentences to describe some topics. As well-known, it is likely in news articles that the most important contents appear in the first paragraph. In our experiment, as we said, we put all the first paragraphs of the relevant news articles together in temporal order. We expect the document describe some topic clearly but remember this is not a *real* document. *Category* (or *class*) means a kind of topics. For example, we may have a case of murder, a case of suicide and a case of corruption.

4.2 Extracting Symbols

Given a document d, let us define a function *Symbol* in such a way that $Symbol(d)$ contains a sequence $o_1 o_2 \cdots o_n$ where o_i is an output symbol:

$$Symbol(d) = o_1 o_2 \cdots o_n$$

We assume all the sentences are delimited with punctuation marks, i.e., a period in this case. After applying morphological analysis to sentences and removing all the sentences not in *past tense* in a document, we extract the final

[2] ”Chasen” is a freeware tool to apply the analysis based on rule-based knowledge for Japanese sentence[2].

verb from each sentence remained as an output symbol[3]. We skip sentences that are not in past tense because they don't capture *change of circumstances* but very often they have some forecast or perspective.

We apply this procedure to all the sentences in a document d and generate a sequence of symbols followed by `EOS` mark as termination mark. And we define the result as $Symbol(d)$. Let us note the sequence of the symbols in $Symbol(d)$ follows the order of the sequences appeared in d.

4.3 Model Calculation of HMM

Let M_c be a model of HMM corresponded to a category c. Given a collection of documents $D_c = \{d_{c1}, d_{c2}, \cdots, d_{c|D_c|}\}$ of the category c, let us discuss how we calculate model parameters of M_c. Assume there are N_{M_c} states in Q of M_c with a common underlying set of output symbols Σ.

Initially we generate random numbers for a state transition matrix an output probability B, and an initial state probability π of M_c. Then we extract a set of symbol sequences L_c from D_c and apply the Baum-Welch algorithm to L_c:

$$L_c = \{Symbol(d_{c1}), \cdots, Symbol(d_{c|D_c|})\}$$

In our case, we give A, B and π randomly as an initial step, and we can't interpret the states in advance. Then, after applying the algorithm, we should examine the results what they mean. Let us discuss this issue later on.

4.4 Estimating Topics

Finally let us discuss how we can estimate a category of a document d by means of HMM.

First of all, we give $Symbol(d) = o_1 o_2 \cdots o_n$ and obtain estimated sequences of the states in each HMM. Given HMM of a category c, let $s_{c1} s_{c2} \cdots s_{cn}$ be the sequence of M_c.

Then we obtain the probability $P(o_1 o_2 \cdots o_n, s_{c1} s_{c2} \cdots s_{cn} | M_c)$ in M_c and maximize this value under several category. That is, we estimate the category c_d as:

$$c_d = argmax_c P(o_1 o_2 \cdots o_n, s_{c1} s_{c2} \cdots s_{cn} | M_c)$$

Let us note that the topic is the sequence estimated by M_{c_d}.

5 Experimental Results

Here we discuss some experimental results to show the usefulness of our approach. First we show how to obtain the results and to evaluate them, then we show the results. Then we examine the results.

[3] In Japanese, we can determine "past tense" easily by examining specific *auxiliary verbs*.

Table 1. Test Corpus

Topic	Training Documents	Test Documents
One-man Crime	91	45
Organizational Crime	35	17
Corruption Scandal	46	22

5.1 Preliminaries

As a test corpus for our experiments, we take all the articles in both Mainichi Newspaper 2001 and 2002 in Japanese and select 3 kinds of news topics, *one-man crime* (crimes by single or a few person such as murder and robbery), *organizational crime* (crimes by organization such as companies) and *corruption scandal.* We have extracted 256 documents in total by hands shown in a table 1. We have divided each of them into two groups, one for training and another for test. According to our procedure described in the previous section, we apply the learning of HMM to calculate models and identify test documents.

In the following experiments, we give 5 states, i.e., $N_{M_c} = 5$. Since there is no knowledge assumed in advance about events to Baum-Welch algorithm, we have tried several preliminary experiments with several number of states, and we select a model with 5 states by which we get the highest classification ratio. There is no sharp reason bust just empirical parameter assumption here. As we said, we apply "Chasen" for morphological analysis to documents in Japanese[2]. We have segmented all the sentences into words with Part-of-Speech tagging.

5.2 Results

Let us show the results. First we show what HMM models we have constructed and examine whether we can interpret them suitably or not. Then we show the classification results with some examples in detail.

Interpreting Models. In figures 3, 4 and 5, we show HMM models by learning several parameters of HMM of 3 categories, one-man crime, organizational crime and corruption scandal respectively. Each figure contains *topology* of the model, which means the HMM states, the transition probability among them and the output symbols with probabilities at each state.

We illustrate the topology of One-man Crime in a figure 3 where a circle means a state and an directed arrow between two states means the transition with the probability. There are output symbols close to a state with the probabilities (we have omitted the symbols with very small probabilities).

Let us interpret each state. It is common to interpret state 1 corresponds to "incident occurrence", because we have many output symbols such as "emergency call", "found" and "stabbed". Similarly state 4 means "arrestment" because we see specific words "arrested" and "without warrant". We have put our interpretation onto the figure in a subjective manner. But the process is easy for us since they describe sharp situation with few exception.

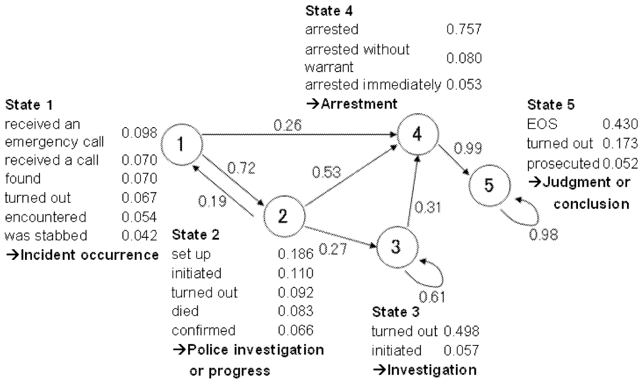


Fig. 3. One-man Crime

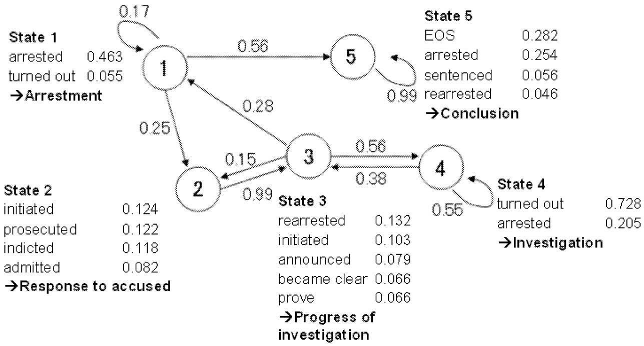


Fig. 4. Organizational Crime

It is also really easy to trace the transitions in the figure. The states with highest probability say that we have ”an incident occurrence” (state 1), ”police investigation” (state 2), ”arrestment” (state 4) and ”conclusion” (state 5) in this order. Any path through state 3 (”progress”) to itself may describe a case to be delayed.

A figure 4 describes a topology of Organizational Crime model in HMM. Unlike the previous model, a symbol ”arrested” (or ”rearrested”) appear in several states. For instance, we get the symbol with the highest probability at a state 3 which means ”progress”. This is because there are more than one persons arrested very often and arrestment can be seen as a part of progression.

In a figure 5, we show a topology of Corruption Scandal case. The model is complicated compared to the other two cases, in fact, there exist two different states (1 and 4) interpreted as ”progress”. The two state have similar probability distribution over output symbols and there is no way to separate the two states with each other. Also the state transition seems complicated and it is hard to see no sharp interpretation here.

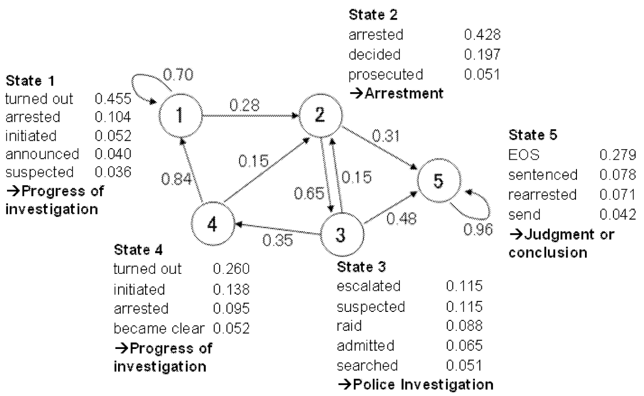


Fig. 5. Corruption Scandal

Table 2. Symbols for Estimating Topics

In the midnight today, a young lady was $\boxed{\text{found}}$ dead by police men.
Police came to a conclusion of murder, and they $\boxed{\text{set up}}$
the investigation headquarter.
About the death of Mr.XX, Tokyo police are doubtful of Mr.YY
and they $\boxed{\text{initiated}}$ to hear from him what he knows.
About the death of Mr.XX, Tokyo police have $\boxed{\text{arrested}}$
Mr.YY with the suspicion of the robbery and the murder.
Mr.YY $\boxed{\text{initiated}}$ testifying the murder of Mr.XX.
Police have $\boxed{\text{rearrested}}$ Mr.YY because of the murder suspect.

Estimating Topics. Let us describe some example of classification of test documents in a table 2.

In this table we show each output symbol with a box that can be obtained through symbol extraction process[4]. We show event sequences that we can estimate by HMM in a table 4, where all the events correspond to the states in figures 3, 4 and 5.

We are convinced of the events generated through One-man Crime case and considered consistent and well-structured. Especially a symbol "arrested" appears just before the settlement of the case, and this means there is only one person arrested within this topic. On the other hand, in a case of organizational crime, a symbol "arrested" is a part of progression in this case, and there may be plural persons arrested.

The likelihood in the table shows the probability $P(o_1 o_2 \cdots o_n, s_{c1} s_{c2} \cdots s_{cn} | M_c)$ for sequence of states and output symbols. We put the class of the maximum probability as an answer and, in this case, we say "one-man crime" class.

[4] Here we just illustrate the results. Remember the process is targeted for documents in Japanese, and no concrete procedure is applied here.

Table 3. Classification Ratio

Topic	Correctly Classified	Total	Correctness Ratio
One-man Crime	34	45	75.6 (%)
Organizational Crime	9	17	52.9
Corruption Scandal	10	22	45.5
Total	53	84	63.1

Here are our results of *correctness ratio*, i.e., how many documents have been correctly classified to each category in a table 3. We got 63.1% as the average correctness ratio, the best one is 75.6.

5.3 Discussions

Let us discuss how we can think about our experimental results and especially about our approach.

First of all, we get the better result to One-man Crime case. It is easy for us to interpret the model states and we get the better ratio. On the other hand, we see the worse situation about a case of Corruption Scandal. That is, it's hard to understand the model and the ratio is not really good. Looking at the documents more closely, we see a variety of the scandals pattern, thus there is no specific pattern of the high probability in training data. In Baum-Welch algorithm, we adjust the several probabilities in such a way that we maximize the likelihood of symbols observed. We may have several kinds of sequences of the symbols which we don't have in training documents very often. Then the likelihood can't be high. In fact, we see many symbols with small probabilities, and that's why we have missed the documents. Also we don't have the excellent quality of state transition probability because of this reason.

The second characteristic is that we got similar symbols at many states. We have training data extracted from news-articles putting stress on 3 topics, and we have similar distribution over news words such as ”arrested”, ”turned out” and ”prosecuted”. This means, in turn, that our approach doesn't depend on collections of symbols but the sequences, which is inherently different from other approach of topic classification.

Table 4. Estimated Events

Symbols	One-man Crime	Organizational Crime	Corruption Scandal
found	incident occurrence	suspicious person arrested	responsible person arrested
setting up	investigation by police	response of arrested person	progress by police
initiated	progress by police	still in progress	progress by police
arrested	offender arrested	progress of investigation	progress of investigation
initiated	concluded	still in progress	progress of investigation
rearrested	concluded	suspicious person arrested	concluded
EOS	concluded	concluded	concluded
Likelihood	6.25×10^{-9}	2.79×10^{-10}	5.87×10^{-18}

6 Conclusion

In this investigation we have proposed how to estimate topics of documents by considering topics as stochastic process. We have examined the case of news documents and discussed classification of topics looking at the event sequences which is hard to discuss traditional approach of classification. We have discussed some experimental results and shown the usefulness of our approach.

We have examined ”complete sequences of topics” (i.e., all the murders had been arrested) and classified them. But it is possible to apply our approach to incomplete sequences, in this case, we may have some sort of forecast or some hints to do for the solutions.

References

1. Allan, J., Carbonell, J., Doddington, G., Yamron, J., Yang, Y.: Topic Detection and Tracking Pilot Study: Final Report, DARPA Broadcast News Transcription and Understanding Workshop (1998)
2. Asahara, M., Matsumoto, Y.: Extended Models and Tools for High Performance Part-of-Speech Tagger, COLING (2000)
3. Barzilay, R., Lee, L.: Catching the Drift: Probabilistic Content Models, with Applications to Generation and Summarization, NAACL/HLT, pp. 113–120 (2004)
4. Blei, D.M., Moreno, P.J.: Topic segmentation with an aspect hidden Markov model, ACM SIGIR, pp. 343–348 (2001)
5. Iwasaki, M.: Statistic Analysis for Incomplete Data, EconomistSha, Inc. (In Japanese) (2002)
6. Manning, C.D., Schutze, H.: Foundations of Statistical Natural Language Processing. MIT Press, Cambridge (1999)
7. Makkonen, J.: Investigations on Event Evolution in TDT. In: Proceedings of HLT-NAACL, Student Workshop, May 2003, Edmonton, Canada, pp. 43–48 (2003)
8. Mitchell, T.: Machine Learning. McGrawHill Companies, New York (1997)
9. Mulbregt, P., van, C.I., Gillick, L., Lowe, S., Yamron, J.: Text Segmentation and Topic Tracking on Broadcast News Via a Hidden Markov Model Approach. ICSLP’98 6, 2519–2522 (1998)
10. Shibata, T., Kurohashi, S.: Unsupervised Topic Identification by Integrating Linguistics and Visual Information Based on Hidden Markov Model, COLING (2005)
11. Yang, Y., Ault, T., Pierce, T., Lattimer, C.W.: Improving Text Categorization Methods for Event Tracking. ACM SIGIR (2000)

The Dictionary-Based Quantified Conceptual Relations for Hard and Soft Chinese Text Clustering

Yi Hu[1], Ruzhan Lu[1], Yuquan Chen[1], Hui Liu[1], and Dongyi Zhang[2]

[1] Department of Computer Science and Engineering,
Shanghai Jiao Tong University, Shanghai, China
{huyi,lu-rz,yqchen,lh_charles}@cs.sjtu.edu.cn
[2] Network Management Center,
Politics College of Xi'an, Xi'an, China
zhang_zhangdyi@163.com

Abstract. In this paper we present a new similarity of text on the basis of combining cosine measure with the quantified conceptual relations by linear interpolation for text clustering. These relations derive from the entries and the words in their definitions in a dictionary, which are quantified under the assumption that the entries and their definitions are equivalent in meaning. This kind of relations is regarded as "knowledge" for text clustering. Under the framework of k-means algorithm, the new interpolated similarity improves the performance of clustering system significantly in terms of optimizing hard and soft criterion functions. Our results show that introducing the conceptual knowledge from the un-structured dictionary into the similarity measure tends to provide potential contributions for text clustering in future.

1 Introduction

Text clustering is a class of techniques that fall under the category of machine learning, which looks forward to automatically segregating texts into groups called clusters. Clusters in our study are collections of similar texts, and they can be created with unsupervised learning. A lot of clustering methods have been applied in the literature over the years, which can eliminate the need for costly manual organization for a large number of texts. On the other side, text clustering is an important step in retrieval and mining of abundant literal data on web pages.

Many clustering methods use the well-known vector space model (*VSM*), as described in the work of Salton et al. [13][14], in which a text is represented as a vector and vector terms are treated independently from one to the other. This kind of representation leads to challenging problems of text clustering: big volume, high dimensionality and complex semantics. For the first two problems, Jing et al have provided an efficient solution that use the scalable subspace clustering with feature weighting *k*-means [9]. In this paper we put more attention to the last problem.

Many researchers focus on indexing conceptual features in texts based on ontology. The nature of their work lies in increasing/decreasing the number of terms or modifying term weights in their own vectors. The results show the effectiveness of these methods [2][8][10]. But their work needs available ontologies while there does not exist available ones for Chinese. To address this issue, we present an approach to

Z. Kedad et al.(Eds.): NLDB 2007, LNCS 4592, pp. 96–106, 2007.

acquiring a set of quantified conceptual relations to replace ontologies. We focus on mining the quantified relations with the aid of the electrical *Modern Chinese Standard Dictionary* (*MCSD*) [11], an unstructured resource.

Another effort different from the ontology-based methods in our work is that these quantified relations are not used for adjusting vectors themselves but the alteration of text similarity. What we are interested in are the relations between concepts in different texts. Thus we define a new similarity measure combining the contribution of these quantified relations with the cosine similarity, i.e., an interpolated style.

In our experiments, we evaluate the interpolated similarity for both hard and soft clustering via the *k*-means algorithm. The reason why we use the *k*-means algorithms is that they are efficient and scalable. Therefore they are proper for processing large volume and high-dimensional text data [15]. The experimental results have demonstrated that the new similarity based clustering scheme is better than the pure cosine-based clustering scheme in which concepts are treated without linkage.

The rest of this paper is organized as follows. Section 2 describes the acquisition of quantified relations. The interpolated similarity is given in Section 3. Section 4 provides the hard and soft criterion function for clustering used in our experiments. Section 5 describes the experimental details and Section 6 concludes this paper.

2 Quantified Conceptual Relations from MCSD

From the view of human cognition, concepts represented by words have logical relations. Therefore we taste to mine these relations and quantify them. In this section, we introduce the idea of calculating this kind of quantified relations. The difference between our work and the others [4][7][13][16] lies in the resource used for quantification. We employ an un-structured dictionary, i.e., the electrical *Modern Chinese Standard Dictionary* (*MCSD*), and acquire the quantified relations via statistical learning. The reason for choosing an ordinary dictionary is that the explanations of entries are usually normative and correct.

In terms of constructing quantified conceptual relations, our motivation is simple. An entry *e* is explained by its definition *D* ($w_1w_2 \ldots w_n$) in the dictionary.

$$e \ : \ w_1w_2 \ldots w_n \, . \tag{1}$$

Where w_1, w_2, … and w_n indicate the words appearing in the definition of entry *e*. We give an intuitive assumption that the meaning of *e* equals to the combinational meaning of its definition. Therefore,

$$Meaning(e) = Meaning(w_1w_2 \ldots w_n) \, . \tag{2}$$

Note that to the state of the art of natural language understanding, the automatic and accurate analysis between the entry and the words in its definition is still a hard task and the semantic relations are usually variable in different contexts. Our work just constructs a kind of static quantified relations. On the other hand, this idea might be able to coarsely describe the relations.

Our idea is simple: if a word w_i appears in the definition of an entry *e* and it rarely appears in other entries' definitions, then this word might contribute more in the combinational sense, i.e., w_i has potential stronger relationship with *e*. For instance, the Chinese word "consumption" is rarely used in the other definitions besides the one of entry "economy". So we can consider "consumption" contribute more to explain "economy" and their quantified relation should be larger. This idea is suitable to be formalized by $tf * idf$.

Here note that Chinese texts have no delimiters to mark word boundaries, so preprocessing a definitions need breaking it into Chinese words, called word segmentation. Then we employ an extended $tf * idf$ formula (3) to get the quantified relations.

$$r(e,w_i)=\begin{cases}\left(1+\log\left(tf(e,w_i)\right)\right)*\log\left(\dfrac{N}{ef(w_i)}\right) & \text{if } tf(e,w_i)>0 \\ 0 & \text{if } tf(e,w_i)=0\end{cases}. \tag{3}$$

Where w_i belongs to $\{w_1, w_2 ..., w_n\}$ and

$r(e,w_i)$ denotes the relation value between e and w_i;

$tf(e,w_i)$ denotes the times of w_i appearing in the definition of e;

$ef(w_i)$ denotes the number of entries in whose definition w_i appears;

$\log\left(\dfrac{N}{ef(w_i)}\right)$ is the "inverse entity frequency";

N denotes the number of all entries in the dictionary.

We get 892,575 pairs by this method. Partial pairs are listed in Table 1 and Table 2.

Table 1. Values of relations between different entries and the same word (computer)

(e, w)	Values of Quantified Relations
(automation, electronic computer)	5.92
(program design, electronic computer)	5.92
(program controlling telephone, electronic computer)	5.92
(hard disk, computer)	5.92
(computer, electronic computer)	12.43

Table 2. Values of relations between the same entry (economy) and different words in its definition

(e, w)	Values of Quantified Relations
(economy, country)	4.07
(economy, consumption)	12.19
(economy, finance)	6.97
(economy, price)	11.83
(economy, circulation)	7.05

The instances in Table 1 are the quantified relations between variant entries and the same word w in their definitions. It is clear that the "*idf*" value of "*w*" is uniform and there will be different values of "*tf*". If the "*tf*" is also equivalent, (e, w) will obtain the same value. Instances in Table 2 are the pairs having the same entry "economy".

In the 892,575 pairs, the largest relation is (materialism, idealistic system), 30.99. The relative small-quantified relations come from the (entries, "stop-word") pairs. For example, the value of (economy, "de") is very close to zero.

Please note that we just process the entries with only one explanation and treat (w_1, w_2) and (w_2, w_1) as different pairs in this study.

3 Interpolated Similarity

Based on the definition of *VSM*, one text is represented by $d = \{b_1, b_2, ...b_l\}$. Where b_i is boolean value (1 or 0) denoting the corresponding concept w_i appears in *d* or not. With respect to this kind of representation, many measures are presented to calculate the similarity between two texts d_i and d_j, such as cosine and *Minkowski* distance including *Euclidean* distance, *Manhattan* distance and *Maximum* distance [1][6]. In our study we investigate the cosine measure that is most widely used for evaluating similarity. It calculates the cosine value of angle of two text vectors in space. A matrix can express the conceptual relations as follows.

$$M = \begin{pmatrix} r_{11} & \cdots & r_{1q} & \cdots & r_{1n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ r_{p1} & \cdots & r_{pq} & \cdots & r_{pn} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ r_{m1} & \cdots & r_{mq} & \cdots & r_{mn} \end{pmatrix}. \tag{4}$$

In this matrix, *m* is the number of word form appearing in the text d_i and *n* is the number of word form in the text d_j. Therefore, r_{pq} is the quantified relations between word w_p in d_i and w_q in d_j.

With the conceptual relations, we transform the standard cosine similarity into the following interpolated formula by:

$$Sim(d_i, d_j) = \lambda \times \frac{\sum_{q=1}^{n}\sum_{p=1}^{m} r(w_{ip}, w_{jq})}{max\{\#_{ij}\} \times MaxR} + (1-\lambda) \times cos(\angle\,(d_i, d_j)) \cdot \tag{5}$$

Seen from Equation (5), the similarity between d_i and d_j consists of the common contributions of two parts: the quantified relations and the standard cosine measure. The parameter λ is the interpolation factor that balances the two contributions. Its value ranges from 0 to 1. The bigger the value of λ is, the more contribution the relations make. In our work, the best value of λ is determined by a set of experiments. It is sure that the value of λ can also be automatically obtained by EM algorithm and that is another work in future. In the first part in interpolated formula (5), the numerator represents the sum of all the items in matrix *M*. Many of them equal to zero. The $\#_{ij}$ denotes the number of items with non-zero values and the $max\{\#_{ij}\}$ is the maximum value in these non-zero values. Therefore $\frac{\sum_{q=1}^{n}\sum_{p=1}^{m} r(w_{ip}, w_{jq})}{max\{\#_{ij}\}}$ calculates the "average value" of all relations. This "average value" is not a true average one, while it just considers those non-zero pairs. The *MaxR* is the largest value out of the 892,575 pairs and it is a normalized factor, which mapping the "average value" into [0,1]. We use three sub-collections to estimate a good $\hat{\lambda}$ in the experiment. Therefore,

$$\hat{\lambda} = \frac{\lambda_1 + \lambda_2 + \lambda_3}{3}. \tag{6}$$

4 Hard and Soft Clustering Criterion Functions

An important characteristic of many partitional clustering algorithms is that they use a global criterion function whose optimization guides the entire clustering process. This needs the criterion function to be optimized by the final solution, and an algorithm that achieves this optimization.

In our study, we only focus on two hard criterion functions, which are referred to as I_1, and G_1 [17], and two soft corresponding criterion functions, SI_1 and SG_1 [18].

4.1 Hard Clustering Criterion Function

The I_1 criterion function (Equation (7)) maximizes the sum of the average pairwise similarities between the texts assigned to each cluster.

$$maximize \quad I_1 = \sum_{r=1}^{k} n_r \left(\frac{1}{n_r^2} \sum_{d_i, d_j \in S_r} sim\left(d_i, d_j\right) \right). \tag{7}$$

Where the $sim\left(d_i, d_j\right)$ is the similarity between d_i and d_j , and the symbols n and k denote the number of texts and the number of clusters, respectively. Zhao et al use the S_1, S_2, ... S_k to denote each one of the k clusters, and n_1, n_2, . . . , n_k to denote the sizes of the corresponding clusters. These symbols have the same meaning through out this paper.

The G_1 criterion function (Equation 8) treats the clustering process as minimizing the edge-cut of each partition. Because this edge-cut based criterion function may have trivial solutions the edge-cut of each cluster is scaled by the sum of the cluster's internal edges [5]. The similarity between two texts is measured using the similarity function, so the edge-cut between the r^{th} cluster and the rest of the texts (*i.e.*, cut(S_r , S-S_r) and the sum of the internal edges between the texts of the r^{th} cluster are given by the numerator and denominator of Equation (8), respectively. For more information of these two hard criterion functions, readers can refer to the referential paper [17].

$$minimize \quad G_1 = \sum_{r=1}^{k} \frac{\sum_{d_i \in S_r, d_j \in S-S_r} sim\left(d_i, d_j\right)}{\sum_{d_i, d_j \in S_r} sim\left(d_i, d_j\right)}. \tag{8}$$

4.2 Soft Clustering Criterion Function

A natural way of introducing ***soft*** clustering solutions is to assign each text to multiple clusters [18]. This is usually achieved by using membership functions that assign a non-negative weight, denoted by $u_{i,j}$ for each text d_i and cluster S_j , therefore $\sum_j u_{i,j} = 1$. The weight indicates the extent to which text d_i belongs to the cluster S_j. In terms of a hard clustering solution, for each text d_i one of these $u_{i,j}$ values is 1 and the rest will be 0. Similarly, the *soft* size $\overline{n_r}$ of the *r*th soft cluster can be computed as $\overline{n_r} = \sum_i u_{i,r}$. A natural way of calculating the overall pairwise similarity between the texts assigned to each cluster is to take into account their membership functions. Specifically, the pairwise similarity in the r^{th} soft cluster is to be $\sum_{i,j} \mu_{i,r} \mu_{j,r} sim\left(d_i, d_j\right)$.

Using these definitions, the soft I_1 criterion function, denoted by SI_1, is defined by Equation (9).

$$maximize \quad SI_1 = \sum_{r=1}^{k} \overline{n_r} \left(\frac{1}{\overline{n_r}^2} \sum_{i,j} \mu_{i,r} \mu_{j,r} sim\left(d_i, d_j\right) \right). \tag{9}$$

To a soft version of the *G*1 criterion function, Ying Zhao et al define the edge-cut between a cluster and the rest of the texts in the collection, and the sum of the weights of the edges between the texts in each cluster. They compute the soft version of the sum of the weights of the edges between the texts of the r^{th} cluster as $\sum_{i,j} \mu_{i,r} \mu_{j,r} sim\left(d_i, d_j\right)$. Similarly, they compute the edge-cut between the r^{th} cluster and the rest of the texts in the collection as $\sum_{i,j} \mu_{i,r} (1\text{-}\mu_{j,r}) sim\left(d_i, d_j\right)$. Using these definitions, the soft version of the *G*1 criterion function, denoted by SG_1, is defined as follows:

$$minimize \quad SG_1 = \sum_{r=1}^{k} \frac{\sum_{i,j} \mu_{i,r} (1\text{-}\mu_{j,r}) sim\left(d_i, d_j\right)}{\sum_{i,j} \mu_{i,r} \mu_{j,r} sim\left(d_i, d_j\right)}. \tag{10}$$

For more information of these soft criterion functions, readers can refer to the referential paper [18].

Given a hard *k*-way clustering solution $\{S_1, S_2, ..., S_k\}$, Jing Zhao et al define the membership of text d_i to cluster S_j to be

$$\mu_{i,j} = \frac{sim\left(d_i, d_j\right)^{\eta}}{\sum_{r=1}^{k} sim\left(d_i, d_j\right)^{\eta}} . \tag{11}$$

Where C_r is the centroid of the hard cluster S_r. The parameter η in Equation (11) is the *fuzzy factor* and controls the "softness" of the membership function and hence the "softness" of the clustering solution. When η is equal to zero, the membership values of a text to each cluster are the same. On the other hand, as η approaches infinity, the soft membership function becomes the hard membership function (*i.e.*, $\mu_{i,j} = 1$, if d_i is most close to S_j ; $\mu_{i,j} = 0$, otherwise). In general, the softness of the clustering solution increases as the value of η decreases and vice versa.

5 Experimental Results

We experimentally evaluated the performance of the interpolated similarity based on these quantified relations and compared them with the pure cosine similarity via using the hard and soft criterion functions.

The greedy nature of the optimize the criterion function does not guarantee that it will converge to a global minima, and the local minima solution it obtains depends on the particular set of seed texts that are selected during the initial clustering. To eliminate some of this sensitivity, the overall process is repeated a number of times. That is, we compute 5 different clustering solutions, and the one that achieves the best value for the particular criterion function is kept. For the rest of this discussion when we refer to the clustering solution we will mean the solution that is obtained by selecting the best out of these 5 potentially different solutions.

5.1 Text Collections

We use a standard text collection in our experiments, which is provided by Fudan University and is available at http://www.nlp.org.cn/. It includes 2,816 texts and each text only belongs to one of the 10 classes: environment, computer, traffic, education, economy, military affairs, sports, medicine, arts and politics. The smallest of these classes contains 200 texts and the largest contains 505 texts. We proportionally partition the overall collection into 7 disjoint subsets (S1~S7) and assure that at least 20 texts in one class should be included in one subset. Moreover, any word that occurs in fewer than two documents is eliminated.

We design two sets of experiment to investigate the interpolated similarity. The first is to select the most suitable λ on subsets S1, S2 and S3, and the second is to compare the performance between the interpolated similarity and cosine measure on the left 4 subsets via hard and soft clustering process.

5.2 Experimental Metrics

For each one of the different subsets, we obtain a 10-way clustering solution that optimized the various hard and soft clustering criterion functions. We compare the interpolated similarity with the pure cosine measure for each criterion function. Specially, in terms of soft criterion functions, we implement them with a fuzzy factor η of different values.

The quality of a solution is evaluated using the ***pair accuracy*** measure that focuses on the relative distribution of texts pairs in each cluster. Given two texts, the correct clustering results contain two cases: <1> d_i and d_j has been assigned into one cluster and they are in the same class in the original collection; <2> d_i and d_j has been assigned into different clusters and they are also in different classes in the original collection. The ratio of the number of the correct pairs is defined as the pair accuracy:

$$Pair\ Accuracy = \frac{\sum_i \sum_j \varphi(d_i, d_j)}{\#(d_i, d_j)} \tag{12}$$

When d_i and d_j satisfy the correct cases, $\varphi(d_i, d_j)$ equals to 1, otherwise 0. The $\#(d_i, d_j)$ is the number of all text pairs in some subset. Note that the repeated pairs ($< d_i, d_j >$, $< d_j, d_i >$) have been considered in count. Equation (12) is a simple evaluating measure compared with entropy and F-Score measures [19], but it is very straightforward. Theoretically, a perfect clustering solution will be the one that leads to clusters containing texts from single class, in which case the pair accuracy will be 1. In general, the bigger the pair accuracy is, the better the clustering solution is.

5.3 Determining the Interpolated Parameter

We use the subsets *S1*, *S2* and *S3* to determine the interpolated parameter λ. This experiment changes λ from 0.1 to 0.9 and observes what the best $\hat{\lambda}$ is for interpolation. As mentioned above, every experiment is performed 5 times to eliminate the local minima problem. The best solution out of the five is the final solution of the experiment. The pair accuracies on *S1*, *S2* and *S3* are listed in Table 3.

Table 3. Look for the best $\hat{\lambda}$ on S1, S2 and S3. The best solutions arrive when λ equals to 0.5, 0.6 and 0.6, respectively.

	.1	.2	.3	.4	.5	.6	.7	.8	.9
S1	.64	.77	.79	.83	**.84**	.81	.81	.74	.78
S2	.68	.72	.74	.80	.77	**.82**	.78	.77	.77
S3	.73	.76	.78	.79	.78	**.83**	.79	.78	.79

In our work, we average the three λ values to calculate the best $\hat{\lambda}$ used in the following experiments. Therefore,

$$\hat{\lambda} = \frac{\lambda_1 + \lambda_2 + \lambda_3}{3} = \frac{0.5 + 0.6 + 0.6}{3} \approx 0.57 \tag{13}$$

5.4 Comparison of the Two Similarities in Hard and Soft Clustering

Our experiments focus on evaluating the quality of the clustering solutions produced by the various hard and soft criterion functions when they are used to compute a 10-way clustering solution via repeated *k*-means.

The results for each subset are shown in Table 4, and in each sub-table, each column corresponds to one of the four criterion functions. The results of the soft criterion functions with various fuzzy factor values are shown in the first five rows labeled by the fuzzy factor values, and those of the various hard criterion functions are shown in the last row. The entries that are boldfaced correspond to the best values obtained among the hard and various soft criterion functions with different η values for each subset.

Table 4. Pair Accuracies of hard and soft clustering on S4, S5, S6 and S7

S4	cosine		Interpolated similarity	
method	I_1	G_1	I_1	G_1
η=1	.77	.78	.81	.79
η=3	.80	.73	**.86**	.81
η=5	**.82**	**.81**	.82	.80
η=7	.75	.79	.82	.79
η=9	.77	.79	.81	.81
hard	.79	.77	.83	**.84**

S5	cosine		Interpolated similarity	
method	I_1	G_1	I_1	G_1
η=1	.73	.71	.82	**.84**
η=3	.78	**.77**	.76	.81
η=5	**.83**	.72	.76	.80
η=7	.76	.73	.79	.81
η=9	.78	.75	.81	.77
hard	.78	.73	**.83**	.75

S6	cosine		Interpolated similarity	
method	I_1	G_1	I_1	G_1
η=1	**.82**	.80	.82	.72
η=3	.80	.79	.80	.79
η=5	.80	.77	.79	.78
η=7	.76	**.83**	.74	.81
η=9	.79	.81	**.88**	**.86**
hard	.80	.81	.84	.84

S7	cosine		Interpolated similarity	
method	I_1	G_1	I_1	G_1
η=1	**.83**	.76	.82	.76
η=3	.81	.74	.85	.82
η=5	.83	.73	.83	.84
η=7	.79	.77	.85	**.86**
η=9	.81	**.79**	**.87**	.81
hard	.82	.78	.86	.83

A number of observations can be made by analyzing the results in these tables: <1> in terms of hard clustering, standard *k*-means using cosine similarity and using conceptual relations based interpolated similarity show that our new similarity measure achieves better performance by average pair accuracy. The interpolated similarity achieves average improvement of +5.3 % on I_1 criterion function and +5.5% on G_1 criterion function. The effect of quantified relations for hard clustering is obvious; <2> for the four subsets, introducing the interpolated similarity into the two soft criterion functions usually improves the quality of the clustering solutions also. The interpolated similarity via *SI*1 criterion function outperformed cosine measure on 3 datasets, among which the relative improvements are greater than +4.8% for subsets with the largest improvement of +7.3%. For SG2 criterion function, the interpolated similarity also achieves better performance when comparing the best solution of interpolated similarity and cosine on three subsets too. The relative improvements are greater than +3.6% and the largest increment is about +9.1%.

Second, the fuzzy factor values that achieve the best clustering solutions seemed to vary for different subsets, which suggests that the proper fuzzy factor values may relate to some characteristics of the subsets and their class conformations. How to automatically choose a suitable fuzzy factor is a challenging work in future.

The experimental results prove that the knowledge represented by quantified conceptual relations has potential contributions for text clustering.

6 Conclusion

In this paper we extend the classical cosine similarity between texts into an interpolated form. And the text clustering by hard and soft criterion functions are studied in our work to evaluate the interpolated similarity by *k*-means algorithm.

In our work, the conceptual relations from the dictionary *MCSD* have been quantified for replacing Chinese ontologies for text clustering. And our idea derives from a simple assumption that the meaning of entry equals to the meaning of its definition. After acquiring the quantified relations, we construct an interpolated similarity for text clustering. The interpolated parameter is estimated by experiments. We presented a comprehensive experimental evaluation involving four different datasets and some discussions about the experimental results. The analysis show that the interpolated similarity consistently promote the performance of hard clustering for the two criterion functions and usually provide contributions to soft separation between the clusters, and lead to better clustering results for most datasets.

We should explain that the method for quantifying the relations in this paper is definitely not the best one for describe the relations between words yet. Obviously, this kind of quantification does not mine the true semantics of language. How to describe the relations more accurately and how to get more valuable conceptual knowledge and how to integrate the knowledge into text clustering are the next considerations in future.

Acknowledgments. This work is supported by NSFC Major Research Program 60496326: Basic Theory and Core Techniques of Non Canonical Knowledge.

References

1. Anderberg, M.R.: Cluster analysis for applications. Academic Press, San Diego (1973)
2. Bloehdorn, S., Hotho, A.: Text classification by boosting weak learners based on terms and concepts. In: Proc. of the 4th IEEE International Conference on Data Mining, UK, pp. 331–334 (2004)
3. Caraballo, S.: Automatic construction of a hypernym-based noun hierarch from text. In: Proc. of the Annual meeting of the association for computational linguistics, USA, pp. 120–126 (1999)
4. Cimiano, P., Hotho, A., Staab, S.: Learning concept hierarchies from text corpora using formal concept analysis. Journal of Artificial Intelligence Research 24, 305–339 (2005)
5. Ding, C., He, X., Zha, H., Gu, M., Simon, H.: Spectral min-max cut for graph partitioning and data clustering. Technical Report TR-2001-XX, Lawrence Berkeley National Laboratory, University of California, Berkeley, CA. (2001)
6. Han, J., Kamber, M.: Data mining: concepts and techniques. Morgan Kaufmann, San Francisco (2001)
7. Hindle, D.: Noun classification from predicate-argument structures. In: Proc. of the Annual meeting of the association for computational linguistics, USA, pp. 268–275 (1990)
8. Hotho, A., Staab, S., Stumme, G.: WordNet improves Text Text Clustering. In: Proc. of the Semantic Web Workshop at SIGIR-2003, 26th Annual International ACM SIGIR Conference, Canada (2003)
9. Jing, L., Ng, M.K., Xu, J., et al.: Subspace clustering of text texts with feature weighting k-means algorithm. In: Proc. of the, PAKDD 2005, Vietnam, pp. 802–812 (2005)
10. Jing, L., Zhou, L., Ng, M.K., et al.: Ontology-based distance measure for text clustering. In: Proc. of the SIAM SDM on Text Mining Workshop (2006)
11. Li, X.J.: Modern Chinese Standard Dictionary. Beijing Foreign Language and Resarch Press and Chinese Press (2004)
12. Mitchell, T.M.: Machine Learning, pp. 191–196. McGraw–Hill, Boston (1997)
13. Salton, G., McGill, M.J.: Introduction to Modern Information Retrieval. McGraw-Hill, New York (1983)
14. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. Communications of the ACM 18, 613–620 (1975)
15. Steinbach, M., Karypis, G., Kumar, V.: A comparison of text clustering techniques. In: Proc. of KDD Workshop on Text Mining, USA (2000)
16. Velardi, P., Fabriani, R., Missikoff, M.: Using text processing techniques to automatically enrich a domain ontology. In: Proc. of the international conference on Formal ontology in information systems, USA, pp. 270–284 (2001)
17. Zhao, Y., Karypis, G.: Criterion functions for text clustering: Experiments and analysis. Technical Report TR #01–40, Department of Computer Science, University of Minnesota, Minneapolis, MN (2001)
18. Zhao, Y., Karypis, G.: Soft Clustering Criterion Functions for Partitional Text Clustering. Technical Report TR #01–40, Department of Computer Science, University of Minnesota, Minneapolis, MN (2001)
19. Zhao, Y., Karypis, G.: Comparison of agglomerative and partitional text clustering algorithms. Technical report, University of Minnesota, pp. 2–14 (2002)

On-Line Single-Pass Clustering Based on Diffusion Maps

Fadoua Ataa Allah[1,2], William I. Grosky[2], and Driss Aboutajdine[1]

[1] Université Mohamed V-Agdal, GSCM-LRIT,
B.P. 1014, Rabat, Maroc
[2] University of Michigan-Dearborn, Dept. CIS,
Dearborn Mi 48128, USA

Abstract. Due to recent advances in technology, online clustering has emerged as a challenging and interesting problem, with applications such as peer-to-peer information retrieval, and topic detection and tracking. Single-pass clustering is particularly one of the popular methods used in this field. While significant work has been done on to perform this clustering algorithm, it has not been studied in a reduced dimension space, typically in online processing scenarios. In this paper, we discuss previous work focusing on single-pass improvement, and then present a new single-pass clustering algorithm, called OSPDM (**O**n-line **S**ingle-**P**ass clustering based on **D**iffusion **M**ap), based on mapping the data into low-dimensional feature space.

1 Introduction

Online clustering is very useful for organizing, analyzing and retrieving data that continuously arrive online. Traditional clustering methods, specifically non-incremental ones, often become inapplicable because they are rely mainly on having the whole document set ready before applying the algorithm. Online clustering is used in many applications such as peer-to-peer information retrieval (P2P) [17,18,19]; topic detection and tracking (TDT) [15,23], and particularly event detection and tracking [1,7,5,6,26,30], and story link detection [4,7,8].

Due to its incremental nature and the fact that it is based on a pairwise distance, a single-pass algorithm is one of the widely used methods for the online clustering. Recently, the improvement of this technique was the object of many works. Some approaches were developed for specific applications such as time-based thresholding [1], and the time-window method [30]. Others had a more general aspect, as discussed in Section 2.2. But to the best of our knowledge, a single-pass algorithm applied to mapping to a reduced dimension has not, so far, been addressed in the literature .

In order to maintain an up-to-date clustering structure, it is necessary to analyze the incoming data in an online manner, tolerating not more than a constant time delay. For this purpose, we develop a new online version of the classical single-pass clustering algorithm, named **O**n-line **S**ingle-**P**ass clustering based on **D**iffusion **M**ap (OSPDM). Our method's efficiency is due mainly to the

Z. Kedad et al.(Eds.): NLDB 2007, LNCS 4592, pp. 107–118, 2007.

document dependencies and the reduced dimension resulting from the diffusion map approach, where we suggest using the singular value decomposition (SVD) updating developed by O'Brien [25] to reduce the dimensionality of the space. Moreover, this allows for a fast computation of approximate distances between data.

The rest of this paper is organized as follows. In Section 2, we provide an overview of the single-pass clustering algorithm, and its various improvements. We review the diffusion map approach in Section 3, and singular value decomposition updating in Section 4. In Section 5, we present the implementation of our clustering algorithm. Then in Section 6, we introduce test data and evaluate the performance of our algorithm. Finally, we conclude and discuss further work in Section 7.

2 Single-Pass Clustering Algorithm

Incremental clustering algorithms are always preferred to traditional clustering techniques, since they can be applied in a dynamic environment such as the web [29,31]. Since the objective of clustering is to classify similar data into different groups, or more precisely, partitioning data set into subsets (clusters) such that the similarities between objects in the same subset is high while inter-subset similarities are weaker, the difference between traditional clustering methods [16] and incremental clustering, in particular, is the ability to process new data as they are added to the data collection. This allows dynamic tracking of the ever-increasing large scale information being put on the web everyday, without having to perform complete reclustering. Various approaches have been proposed [14], including a single-pass clustering algorithm.

2.1 Algorithm

Single-pass clustering, as the name suggests, requires a single, sequential pass over the set of documents it attempts to cluster. The algorithm classifies the next document in the sequence according to a condition on the similarity function employed. At every stage, based on the comparison of a certain threshold and the similarity between a document and a defined cluster, the algorithm decides on whether a newly seen document should become a member of an already defined cluster or the centre of a new one. Usually, the description of a cluster is the centroid (average vectors of the documents representations included in the cluster in question), and a document representations consists of a term-frequency vector.

Basically, the single-pass algorithm operates as follow:
For each document d in the sequence loop

1. find a cluster C that minimizes the distance $D(C,\ d)$;
2. if $D(C,\ d) < t$ then include d in C;
3. else create a new cluster whose only document is d;

End loop.
where t is the similarity threshold value, which is often derived experimentally.

2.2 Previous Work

Recently, many studies have been done with the objective of improving the performance of the single-pass clustering algorithm. In this section, we discuss some improvement approaches; exclusively those related to the algorithm itself, independently of the application specificities, and those related to document features.

One of these studies was focused on the document ordering effect in single-pass clustering [18], because of the fact that the output of the single-pass algorithm is known to be crucially dependent on the order in which documents are input.

By comparing the cosine distance to the Euclidian distance defined for two vectors $x = (x_1, x_2, ..., x_n)$ and $y = (y_1, y_2, ..., y_n)$ in n-dimension space as follows:

$$D_{Cos|n}(x,\ y) = 1 - \frac{\sum_{i=1}^{n} x_i.y_i}{\sqrt{\sum_{i=1}^{n} x_i^2 \sum_{i=1}^{n} y_i^2}},\ D^2_{Euc|n}(x,\ y) = \sum_{i=1}^{n} (x_i - y_i)^2,$$

we remark that by adding a new vector $z = (z_1, z_2, ..., z_n, z_{n+1}, ...z_{n+p})$ to the set of documents we have that

$$D_{Cos|n+p}(x,\ z) = D_{Cos|n}(x,\ z),$$

in spite of the fact that

$$D^2_{Euc|n+p}(x,\ z) = D^2_{Euc|n}(x,\ z) + \sum_{i=n+1}^{n+p} z_i^2.$$

So, it becomes clear that the Euclidian distance of two vectors depends principally on the dimension of the space where they are represented. Thus independently of the threshold choice, the single pass clustering output will be affected by the ordering of the documents received, especially so for large values of p.

The following example makes this idea more clear: let $C_1 = (1,\ 0,\ 1)$ and $C_2 = (1,\ 1,\ 0)$ the centroids of two clusters, each consisting of one document, $x = (1,\ 0,\ 1,\ 4,\ 4,\ 4)$ and $y = (1,\ 0,\ 2)$ two vectors that will be clustered.

In the 1^{st} case, we cluster the vector x before the vector y. For that, we compute the distances between the vector x and the centroids C_1 and C_2: $D_{Cos}(C_1,\ x) = 0.80$, $D_{Cos}(C_2,\ x) = 0.90$, $D_{Euc}(C_1,\ x) = 6.92$, $D_{Euc}(C_2,\ x) = 7.07$. The results of the two metrics show that the vector x will be added to the 1^{st} cluster. Thus, we update the centroid $C_1 = (1,\ 0,\ 1,\ 2,\ 2,\ 2)$, than we cluster the vector y. The distances $D_{Cos}(C_1,\ y) = 0.19$, $D_{Cos}(C_2,\ y) = 0.68$, $D_{Euc}(C_1,\ y) = 3.60$, and $D_{Euc}(C_2,\ y) = 2.23$ show that y will belong to the 1^{st} cluster for the cosine distance, while using the Euclidian distance it will belong to the 2^{nd} cluster.

In the 2^{nd} case, we cluster the vector y before the vector x. The distances $D_{Cos}(C_1,\ y) = 0.05$, $D_{Cos}(C_2,\ y) = 0.68$, $D_{Euc}(C_1,\ y) = 1.00$, and D_{Euc} $(C_2,\ y) = 2.23$ implicate that the vector y will be added to the 1^{st} cluster for both metrics. After updating the 1^{st} centroid $C_1 = (1,\ 0,\ \frac{3}{2})$, we compute the distances between the vector x and the centroids: $D_{Cos}(C_1,\ x) = 0.80$, $D_{Cos}(C_2,\ x) = 0.90$, $D_{Euc}(C_1,\ x) = 6.94$, $D_{Euc}(C_2,\ x) = 7.07$, from which we conclude that x will be added to the 1^{st} cluster for both metrics.

Thus, we can conclude that document ordering dos not effect the single-pass clustering, announced in [18], is precisely related to the use of the cosine distance in the experiments. Hence, we can confirm that the cosine distance is the best similarity distance to use for the on-line clustering purpose in the case of a sparse matrix.

Another work was relayed on the study of the effect of the linguistic features and the weighting schemes on the single pass clustering output [15]. However, the term weighting correction procedure, due to the term-document matrix updating, was not specified. For this purpose, we suggest to use the procedure proposed in [3].

3 Diffusion Maps

Based on the diffusion framework, recently introduced by Coifman and Lafon in [9,10], the generated diffusion distance reflects the connectivity of the documents in a given corpus. Using this condition, each direction of the vector space in the new vector representation corresponds to a document in the corpus instead of a term in the classical representation (Salton's vector space) [27]. In other words, the document vectors in the diffusion space are n-dimensional, where n is the number of documents in the corpus to which the dimension will be reduced before they are clustered.

In this section, we describe in brief the construction of the diffusion map that first appeared in [9]. Given a corpus of document, D, construct a weighted function $k_\epsilon(d_i,\ d_j)$ for $d_i, d_j \in D$ and $1 \leq i, j \leq N$ with $N = crad(D)$. $k_\epsilon(d_i,\ d_j)$ is also knowen as the *kernel* and satisfies the following properties:

- k_ϵ is symmetric: $k_\epsilon(d_i,\ d_j) = k_\epsilon(d_j,\ d_i)$
- k_ϵ is positivity preserving: for all d_i and d_j in the corpus D, $k_\epsilon(d_i,\ d_j) \geq 0$
- k_ϵ is positive semi-definite: for any choice of real number $\alpha_1, ..., \alpha_N$, we have $\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j k_\epsilon(d_i,\ d_j) \geq 0$.

This kernel represents some notion of affinity or similarity between the documents of D, as it describes the relationship between pairs of documents in the corpus. In this sense, one can think of the documents as being the nodes of a symmetric graph whose weight function is specified by k_ϵ. The kernel measures the local connectivity of the documents and hence captures the local geometry of the corpus D. Several choices for the kernel are possible, all leading to different analyses of the data. In [20], the Gaussian kernel (kernel based on the Euclidian distance), defined as

$$k_\epsilon(d_i,\ d_j) = \exp(\frac{\|d_i - d_j\|^2}{\epsilon}),$$

was used. In our work, we suggest to use a kernel based on the cosine distance, because is the widely used for text documents. The latter kernel is defined as

$$k_\epsilon(d_i, d_j) = \exp(\frac{1 - d_i d_j}{\epsilon}).$$

The idea behind the diffusion map is to construct the global geometry of the data set from the local information contained in the kernel k_ϵ. The construction of the diffusion map involves the following steps. First, assuming that the transition probability between documents d_i and d_j is proportional to $k_\epsilon(d_i,\ d_j)$ we construct an $N \times N$ Markov matrix by

$$M(i,j) = \frac{k_\epsilon(d_i, d_j)}{p_\epsilon(d_i)}$$

where p_ϵ is the required normalization constant, given by

$$p_\epsilon(d_i) = \sum_j k_\epsilon(d_i, d_j).$$

For large enough values of ϵ the Markov matrix M is fully connected (in the numerical sense) and therefore has an eigenvalue $\lambda_0 = 1$ with multiplicity one and a sequence of an additional $r-1$ non-increasing eigenvalues $\lambda_1 \leq 1$ (where r is the matrix rank), with corresponding right eigenvectors ψ_1.

The stochastic matrix M naturally induces a distance between any two documents. Thus, we define the *diffusion distance* as

$$D^2(d_i, d_j) = \Sigma_l \lambda_l^2 (\psi_l(d_i) - \psi_l(d_j))^2$$

and the diffusion map as the mapping from the vector d, representing a document, to the matrix

$$\Psi(d) = \begin{pmatrix} \lambda_0 \psi_0(d) \\ \lambda_1 \psi_1(d) \\ \vdots \\ \lambda_{n-1} \psi_{n-1}(d) \end{pmatrix}$$

for a value n. By retaining only the first n eigenvectors we embed the corpus D in an n-dimensional Euclidean *diffusion space*, where $\psi_0, \psi_1, ..., \psi_{n-1}$ are the coordinate axes of the documents in this space. Note that typically $n \ll N$ and hence we obtain a dimensionality reduction of the original corpus.

4 Singular Value Decomposition

Different methods for reducing the dimensionality of the diffusion space, thereby reducing the complexity of data representation and speeding up similarity computation times, have been investigated, such as the Graph Laplacian [9,20], Laplace-Beltrami [9,10,11], the Fokker-Planck operator [9,10], and the singular value decomposition [28]. Due to the online clustering requirements, we have chosen to use the SVD-updating method [25], BDO95, [3] in our approach.

4.1 Singular Value Decomposition Background

Given an $m \times n$ matrix A, its singular value decomposition denoted by $SVD(A)$ is defined as:

$$A = USV^T \ , \tag{1}$$

where the columns of U and V are the left and right singular vectors, respectively, corresponding to the monotonically decreasing (in value) diagonal elements of S, which are called the *singular values* of the matrix A.

The first k columns of the U and V matrices and the first (largest) k singular values of A are used to construct a rank-k approximation ($k \ll min(m, n)$) to A via

$$A_k = U_k S_k V_k^T \ . \tag{2}$$

The columns of U and V are orthogonal, such that $U^T U = V^T V = I_r$, where r is the rank of the matrix A. A theorem due to Eckart and Young [13] suggests that A_k, constructed from the k-largest singular triplets[1] of A is the closest rank-k approximation (in the least squares sense) to A [2].

4.2 SVD-Updating

Suppose a matrix A has been generated from a set of data in a specific space, and the SVD of this matrix has been computed. If more data (represented by rows or columns) must be added, three alternatives for incorporating them currently exist: recomputing the SVD of the updated matrix, folding-in the new rows and columns, or using the SVD-updating method developed in [25].

Recomputing the SVD of a larger matrix requires more computation time and, for large problems, may be impossible due to memory constraints. Recomputing the SVD allows the new p rows and q columns to directly affect the structure of the resultant matrix by creating a new matrix $A^{(m+p)\times(n+q)}$, computing the SVD of the new matrix, and generating a different A_k matrix. In contrast, folding-in is based on the existing structure, the current A_k, and hence new rows and columns have no effect on the representation of the pre-existing rows and columns. Folding-in requires less time and memory but, following the study undertaken in [3], has deteriorating effects on the representation of the new rows and columns. On the other hand, as discussed in [25,3], the accuracy of the SVD-updating approach can be easily compared to that obtained when the SVD of $A^{(m+p)\times(n+q)}$ is explicitly computed.

The process of SVD-updating requires two steps, which involve adding new columns and new rows.

Overview. Let D denote the p new columns to process, then D is an $m \times p$ matrix. D is appended to the columns of the rank-k approximation of the $m \times n$

[1] The triple $\{U_i, \sigma_i, V_i\}$, where $S = diag(\sigma_1, \sigma_2, ..., \sigma_k)$, is called i^{the} singular triplet. U_i and V_i are the left and right singular vectors, respectively, corresponding to i^{the} largest singular value, σ_i, of the matrix A.

matrix A, i.e., from Equation.2, A_k so that the k-largest singular values and corresponding singular vectors of

$$B = (A_k|D) \tag{3}$$

are computed. This is almost the same process as recomputing the SVD, only A is replaced by A_k. Let T denote a collection of q rows for SVD-updating. Then T is a $q \times n$ matrix. T is then appended to the rows of A_k so that the k-largest singular values and corresponding singular vectors of

$$C = (\frac{A_k}{T}) \tag{4}$$

are computed.

SVD-Updating procedures. The mathematical computations required in each phase of the SVD-updating process are detailed in this section. SVD-updating incorporates new row or column information into an existing structured model (A_k from Equation.2) using the matrices D and T, discussed in the Overview, SVD-updating exploits the previous singular values and singular vectors of the original A as an alternative to recomputing the SVD of $A^{(m+p)\times(n+q)}$.

Updating column. Let $B = (A_k|D)$ from Equation.3 and define

$$SVD(B) = U_B S_B V_B^T.$$

$$\text{Then, } U_k^T B \begin{pmatrix} V_k & 0 \\ 0 & I_p \end{pmatrix} = (S_k|U_k^T D), \text{ since} A_k = U_k S_k V_k^T.$$

If $F = (S_k|U_k^T D)$ and $SVD(F) = U_F S_F V_F^T$, then it follows that

$$U_B = U_k U_F, V_B = \begin{pmatrix} V_k & 0 \\ 0 & I_p \end{pmatrix} V_F, \text{ and } S_B = S_F.$$

Hence U_B and V_B are $m \times k$ and $(n+p) \times (k+p)$ matrices, respectively.

Updating row. Let $C = (\frac{A_k}{T})$ from Equation.4 and define

$$SVD(C) = U_C S_C V_C^T.$$

$$\text{Then } \begin{pmatrix} U_k^T & 0 \\ 0 & I_q \end{pmatrix} C V_k = (\frac{S_k}{T V_k}).$$

If $H = (\frac{S_k}{T V_k})$ and $SVD(H) = U_H S_H V_H^T$, then it follows that

$$U_C = \begin{pmatrix} U_k^T & 0 \\ 0 & I_q \end{pmatrix} U_H, V_C = V_k V_H, \text{ and } S_C = S_H.$$

Hence U_C and V_C are $(m+q) \times (k+q)$ and $n \times k$ matrices, respectively.

5 The OSPDM Algorithm

In our approach, we are interested in taking advantage of the semantic structure and the documents' dependencies created due to the diffusion map, in addition to the resulting reduced dimensionality, which leads to significant savings of computer resources and processing time. More specifically, when we take in consideration the studies in [12,22]; where, we have established that the best reduced dimension related to the SVD method for document clustering is restricted to the first tens of dimensions.

SVD-updating is used to rewrite an arbitrary rectangular matrix, such as a Markov matrix, as a product of three other matrices - a matrix of left singular vectors, a diagonal matrix of singular values, and a matrix of right singular vectors. As the Markov matrix is symmetric, both left and right singular vectors provide a mapping from the document space to a newly generated abstract vector space. The more important characteristics of SVD-updating are the guarantee of orthogonality in the singular vectors, and its accuracy compared to the one obtained when the SVD of the updated matrix is explicitly computed.

Hence, our approach in developing the OSPDM algorithm is resumed as follow: Given a collection D of n documents, a new document d that should be added to the existing collection D, and a set C of m clusters.

1. Generate the term-document matrix A from the set D.
2. Compute the Markov matrix M for the n documents.
3. Generate $SVD(M) = U_M S_M V_M^T$.
4. Choose the best reduced dimension for the clustering task, $M_k = U_k S_k V_k^T$.
5. Update the term-document matrix A by adding the column representing the document d and the needed rows if the new document contains some new terms.
6. Update the Markov matrix M (as M is symmetric, one can update just rows R_M).
7. Apply SVD-updating for $T = (\dfrac{M_k}{R_M})$:
   (a) Put $H = (\dfrac{S_k}{R_M V_k})$, and generate $SVD(H) = U_H S_H V_H^T$.
   (b) Compute $U_T = \begin{pmatrix} U_k & 0 \\ 0 & 1 \end{pmatrix} U_H$.
   (c) Compute $V_T = V_k V_H$, and $S_T = S_H$ (for the next iteration).
8. Using the reduced dimension k of the matrix U_T, update the centroids of the m clusters.
9. Apply a step of the single-pass clustering:
   (a) Find a cluster C_i that minimizes the distance $D(C_i,\ U_{Tk}(n+1,\ 1:k))$.
   (b) If $D(C_i,\ U_{Tk}(n+1,\ 1:k)) < t$ then include d in C_i, with t as a specified threshold, and $n = n + 1$.
   (c) Else create a new cluster C_{m+1} whose represented by $U_{Tk}(n+1,\ 1:k)$, and $m = m + 1$.

6 Experiments

To evaluate the effective power of the OSPDM clustering algorithm, we have mixed documents from multiple topics arbitrarily selected from standard information science test collections. The text objects in these collections are bibliographic citations (consisting of the full text of document titles, and abstracts), or the full text of short articles. Table 1 gives a brief description and summarizes the sizes of the datasets used in our experiments.

Cisi: document abstracts in library science and related areas published between 1969 and 1977 and extracted from Social Science Citation Index by the Institute for Scientific Information.

Cran: document abstracts in aeronautics and related areas originally used for tests at the Cranfield Institute of Technology in Bedford, England.

Med: document abstracts in biomedicine received from the National Library of Medicine.

Reuters: some short articles belonging to the Reuters-21578 collection. This collection consists of news stories, appearing in the Reuters newswire for 1987, mostly concerning business and the economy.

Table 1. Size of collections

Collection name	Cisi	Cran	Med	Reuters
Document number	1460	1400	1033	425

6.1 Evaluation Criterion

We used two different metrics to evaluate the results of the single-pass clustering algorithm: *accuracy* expressing the degree of veracity, and *mutual information* denoting the degree of dependency between the origin classes and the predicted clusters.

Let l_i be the label assigned to d_i by the clustering algorithm, and α_i be d_i's actual label in the corpus. Then, accuracy is defined as $\frac{\sum_{i=1}^{n} \delta(map(l_i), \alpha_i)}{n}$ where $\delta(x, y)$ equals 1 if $x = y$ and equals zero otherwise. $map(l_i)$ is the function that maps the output label set of the single-pass algorithm to the actual label set of the corpus. Given the confusion matrix of the output, a best such mapping function can be efficiently found by Munkres's algorithm [24].

Mutual information is a metric that does not require a mapping function. Let $L = \{l_1, l_2, ..., l_k\}$ be the output label set of the single-pass algorithm, and $A = \{\alpha_1, \alpha_2, ..., \alpha_k\}$ be the actual label set of the corpus with the underlying assignments of documents to these sets. The mutual information (MI) of these two labelings is defined as:

$$MI(L, A) = \sum_{l_i in L, \alpha_j in A} P(l_i, \alpha_j).\log_2 \frac{P(l_i, \alpha_j)}{P(l_i).P(\alpha_j)}$$

where $P(l_i)$ and $P(\alpha_j)$ are the probabilities that a document is labeled as l_i and α_j by the algorithm and in the actual corpus, respectively; and $P(l_i, \alpha_j)$ is the probability that these two events occur together. These values can be derived from the confusion matrix. We map the MI metric to the $[0,1]$ interval by normalizing it with the maximum possible MI that can be achieved with the corpus. The normalized MI is defined as $\widehat{MI}(L,A) = \dfrac{MI(L,A)}{MI(A,A)}$.

6.2 Results

In the following, we evaluate the results of the single-pass clustering algorithm applied in three different vector spaces: Salton space, diffusion space and updated diffusion space, for three data sets. The evaluation is done by comparing the accuracy and the mutual information.

The data set **Cisi-Med** contains all documents of the collections Cisi and Med. In **Cran-Cisi-Med**, we mix all documents of the three collections: Cran, Cisi, and Med. In spite in **Cran-Cisi-Med-Reuters**, we use 500 documents from each collection of Cran, Cisi, and Med, mixed with the 425 Reuters documents.

Table 2. Performances of the single-pass clustering

Space	Salton		DM		Upd-DM	
Set	Acc	MI	Acc	MI	Acc	MI
Cisi-Med	87.16	65.52	91.29	72.12	91.41	72.56
Cran-Cisi-Med	60.82	37.21	80.5	69.29	79.83	68.25
Cran-Cisi-Med-Reuters	26.07	0.24	81.61	84.08	77.87	83.89

From the results of Table 2, we can see that the performance of the single-pass clustering algorithm in the diffusion space is better than the Salton space; while, it is almost similar to the performance in the updated diffusion space. More precisely, the slight performance decrease in the updated diffusion space is due to the updating process, while the dramaticall variation on the mutual information measure in Salton space, when Reuters collection is incorporated, is due to the cluster number inexactitude for this case even by trying a variety of threshold values.

On the other hand, in view of the fact that embedded space is restricted to the first ten dimensions, the single-pass algorithm requires less computation time in both the diffusion and the updated diffusion spaces than the Salton space, which is more than compensates for the updating process runtime.

7 Conclusion

Clustering data online as it arrives is a recent and challenged studied problem, due to the recent advances in technology. As a first study in this field, we have focused our approach on stationary text data, where our algorithm has enhanced efficiency. Applications on multimedia data (including images, audio, and video) and non-stationary data will be one of our further works.

References

1. Allan, J., Papka, R., Lavrenko, V.: On-Line New Event Detection and Tracking. 21^{st} ACM SIGIR conf., pp. 37–45 (1998)
2. Belkin, N., Croft, W.: Retrieval techniques. ARIST vol. 22 Ch. 4, pp. 109–145 (1987)
3. Berry, M.W., Dumais, S.T., O'Brien, G.W.: Using Linear Algebra for Intelligent Information Retrieval. SIAM Review 37(4), 573–595 (1995)
4. Brown, R.D.: Dynamic Stopwording for Story Link Detection. 2^{nd} HLT conf., pp. 190–193 (2002)
5. Chen, C.C., Chen, Y.T. Chen, M.C.: An Aging Theory for Event Life Cycle Modeling. IEEE-SMC Tran. Part A (to Appear)
6. Chen, C.C., Chen, Y.T., Sun, Y., Chen, M.C.: Life Cycle Modeling of News Events Using Aging Theory. 14^{th} Machine Learning European Conf., pp. 47–59 (2003)
7. Chen, F.R., Farahat, A.O., Brants, T.: Story Link Detection and New Event Detection are Asymmetric. HLT–NAACL Conf (2003)
8. Chen, F.R., Farahat, A.O., Brants, T.: Multiple Similarity Measures and Source-Pair Information in Story Link Detection. HLT–NAACL Conf, pp. 313–320 (2004)
9. Coifman, R.R., Lafon, S., Lee, A.B., Maggioni, M., Nadler, B., Warner, F., Zucker, S.: Geometric Diffusions as a Tool for Harmonics Analysis and Structure Definition of Data: Diffusion Maps. Proceedings of the National Academy of Sciences 102(21), 7426–7431 (2005)
10. Coifman, R.R., Lafon, S.: Diffusion Maps. Appl. Comput. Harmon. Anal. 21(1), 6–30 (2006)
11. Coifman, R.R., Lafon, S.: Geometric Harmonics: A Novel Tool for Multiscale Out-of-Sample Extension of Empirical Functions. Appl. Comput. Harmon. Anal. 21(1), 31–52 (2006)
12. Dhillon, I.S., Modha, D.S.: Concept Decompositions for Large Sparse Text Data using Clustering. Machine Learning 42(1-2), 143–175 (2001)
13. Golub, G., Reinsch, C.: Handbook for Automatic Computation II: Linear Algebra. Springer, Heidelberg (1971)
14. Hammouda, K.M., Kamel, M.S.: Incremental Document Clustering Using Cluster Similarity Histograms. IEEE-WI Conf, pp. 597–601 (2003)
15. Hatzivassiloglou, V., Gravano, L., Maganti, A.: An Investigation of Linguistic Features and Clustering Algorithms for Topical Document Clustering. 23^{rd} ACM SIGIR Conf., pp. 224–231(2000)
16. Jain, A., Dubes, R.: Algorithms for Clustering Data. Prentice-Hall, Englewood Cliffs (1988)
17. Klampanos, I.A., Jose, J.M.: An Architecture for Information Retrieval over Semi-Collaborating Peer-to-Peer Networks. ACM Symp. 2, 1078–1083 (2004)
18. Klampanos, I. A., Jose, J. M., Rijsbergen, C. J. K.: Single-Pass Clustering for Peer-to-Peer Information Retrieval: The Effect of Document Ordering. 1^{st} INFOSCALE Conf. Art. 36 (2006)
19. Krishnamurthy, B., Wang, J., Xie, Y.: Early Measurements of a Cluster-Based Architecture for P2P Systems. ACM SIGCOMM, pp. 105–109 (2001)

20. Lafon, S., Lee, A.B.: Diffusion Maps and Coarse-Graining: A Unified Framework for Dimensionality Reduction, Graph Partitioning, and Data Set Parameterization. IEEE-TPAMI Tran. 28(9), 1393–1403 (2006)
21. [LeA04] Leuski, A., Allan, J.: Interactive Information Retrieval Using Clustering and Spatial Proximity. UMUAI 14(2), 259–288 (2004)
22. Lerman, K.: Document Clustering in Reduced Dimension Vector Space. Unpublished Manuscript (1999), http://www.isi.edu/lerman/papers/papers.html
23. Makkonen, J., Ahonen-Myka, H., Salmenkivi, M.: Topic Detection and Tracking with Spatio-Temporal Evidence. 25^{th} ECIR, pp. 251–265 (2003)
24. Munkres, J.: Algorithms for the Assignment and Transportation Problems. JSTOR 5(1), 32–38 (1957)
25. O'Brien, G.W.: Information Management Tools for Updating an SVD Encoded Indexing Scheme. Master's Thesis, Knoxville University (1994)
26. Papka, R., Allan, J.: On-line New Event Detection using Single-Pass Clustering. UMASS Computer Science Technical Report, pp. 98–21 (1998)
27. Salton, G., McGill, M.J.: Introduction to Modern Information Retrieval. McGraw Hill Publishing Company, New York (1983)
28. Vaidya, U., Hagen, G., Lafon, S., Banaszuk, A., Mezic, I., Coifman, R.R.: Comparison of Systems using Diffusion Maps. 44^{th} IEEE CDC-ECC, pp. 7931–7936 (2005)
29. Wong, W., Fu, A.: Incremental Document Clustering for Web Page Classification. Int. IS Conf (2000)
30. Yang, Y. , Pierce, T., Carbonell, J.: A Study on Retrospective and On-Line Event Detection. 21^{st} ACM SIGIR Conf., pp. 28–36 (1998)
31. Zamir, O., Etzioni, O.: Web Document Dlustering: A Feasibility Demonstration. 21^{st} ACM SIGIR Conf., pp. 46–54 (1998)

Selecting Labels for News Document Clusters

Krishnaprasad Thirunarayan, Trivikram Immaneni, and Mastan Vali Shaik

Metadata and Languages Laboratory
Department of Computer Science and Engineering
Wright State University, Dayton, Ohio-45435, USA
{t.k.prasad,immaneni.2,shaik.7}@wright.edu
http://www.cs.wright.edu/~tkprasad

Abstract. This work deals with determination of meaningful and terse cluster labels for News document clusters. We analyze a number of alternatives for selecting headlines and/or sentences of document in a document cluster (obtained as a result of an entity-event-duration query), and formalize an approach to extracting a short phrase from well-supported headlines/sentences of the cluster that can serve as the cluster label. Our technique maps a sentence into a set of significant stems to approximate its semantics, for comparison. Eventually a cluster label is extracted from a selected headline/sentence as a contiguous sequence of words, resuscitating word sequencing information lost in the formalization of semantic equivalence.

1 Introduction

A scalable approach to processing large document datasets (such as Medline, News documents, etc.) can be obtained by indexing and classifying the documents by stamping each document with metadata terms from a well-defined ontology that reflects and abstracts the document's content, and then manipulating only the metadata terms in lieu of the document content. For instance, the UMLS[1] terms (metadata) can be used to construct and label a related set of Medline documents involving certain genes, diseases, or organs [14], and the SmartIndex tags (weighted metadata) can be used to construct and label a related set of News documents involving certain entites or events [12]. Metadata-based cluster labels can be significantly improved to better indicate the content of the document clusters obtained in response to entity-event search queries, by generating labels that are grounded in and extracted from the document text of the document clusters. This paper presents a simple technique to construct and select good cluster labels in the context of News documents obtained in response to search queries involving entities and events.

As an illustration, consider sentence fragments and headlines in recent News about the entity "Nokia" and the event "Mergers and Acquisitons".

– Nokia acquires Intellisync.
– Intellisync to be acquired by Nokia.

[1] Unified Medical Language System.

Z. Kedad et al.(Eds.): NLDB 2007, LNCS 4592, pp. 119–130, 2007.

- Nokia's (NOK) acquisition of Intellisync (SYNC) will not change the overall picture for company, says Greger Johansson at Redeye in Stockholm.
- The acquisition of Intellisync supports Nokia's goal to be the leader in enterprise mobility and enhances the ability of its customers to connect devices to data sources, applications and networks.
- Nokia and Intellisync have signed a definitive agreement for Nokia to acquire Intellisync.
- Nokia's Intellisync buy.
- Nokia's purchase of Intellisync.

Besides a reference to the explicitly searched entity (that is, "Nokia") and the event (that is, "acquires", "buy" etc.), the cluster label should contain other relevant information (such as "Intellisync") about the queried subjects (analogous to answer extraction), to provide a concise highlight of the document collection. For the above example, electing "Nokia acquires Intellisync" seems reasonable for the following reasons:

- It is *sound*, containing "Nokia", and a reference to "Mergers and Acquisions" via "acquires".
- It is *complete*, containing additional relevant information "Intellisync".
- It is *well-supported*, with majority of the document fragments providing supporting evidence for it.

The issue of selecting cluster labels natually arises in the context of construction of timelines of trends (e.g., Google Trends [11] response for "Chemistry vs Physics"), implementation of entity-event-duration timelines with call-out labels (e.g., Microsoft and Mergers & Acquisitions in the Year 2005) from the News Document dataset, etc.

In summary, we address the issue of formalizing sentence fragments of News documents that abstracts the meaning of sentences adequately and is lightweight so as to be scalable. Section 2 formalizes the selection of "good" cluster labels in two steps: Section 2.1 motivates and specifies the selection of a promising sentence and analyzes various other alternatives to justify the superiority of the chosen criteria. Section 2.2 explains how to delimit a concise and comprehensible label from the chosen sentence. Section 3 considers the restricted situation when only document headlines are available but not the document contents, for proprietary reasons. Section 4 discusses the implementation of the Timeline application in News documents context. Section 5 briefly reviews some of the recent related work. Section 6 concludes with suggestions for future work.

2 Construction and Election of Cluster Labels

We make the pragmatic assumption that the documents in a News document cluster contain sentences and/or headlines that can yield cluster labels, and propose an approach to selecting cluster labels by extracting a content phrase from a chosen high-scoring sentence or headline containing the queried subjects (entities and events). We assume that sentences include the headline.

2.1 Sentence Selection from Metadata Screened Document Cluster

Problem Statement: Consider a cluster of documents $CD = \{D_1, D_2, ..., D_m\}$ for an entity EN and an event EV. Extract, from the cluster documents, a *well-supported* sentence that contains phrasal references to EN and EV.

Informally, a well-supported sentence is obtained by maximizing the number of documents that support the sentence, by maximizing the degree of overlap with a sentence in each document, and by minimizing its length, subject to the constraint that the sentence contains phrasal references to EN and EV. The rest of this section assumes the existence of such sentences and formalizes the notion of well-supportedness. (In other words, this paper takes a shallow, scalable approach to extracting a "good" label, as opposed to understanding the content of the document cluster and synthesizing a label from its meaning.)

Proposed Approach

1. Let $sen(D_i)$ refer to the set of sentences in document D_i that each contain phrasal references to the entity EN and the event EV. We call them significant sentences of the document D_i.

   The phrases corresponding to an entity or an event can be obtained from the domain knowledge used to stamp the documents with metadata terms. This part of the domain knowledge is encapsulated as Metathesaurus in UMLS or as Concept Definitions in the context of News documents. It includes synonyms, acronyms, and other equivalent usages for a metadata term. A mature indexing and search engine can be used to determine if an entity phrase and an event phrase co-occur in a sentence (or a paragraph) of a document. (Otherwise, such an engine can be built using open source APIs such as Apache Lucene [13] by materializing sentence/paragraph separators. We skip the implementation details here because they are peripheral to the main theme of this paper.) In the case of News documents, applications can be built to determine and extract metadata terms from each sentence of a document using special purpose indexing APIs. Note also that the co-occurrence within a sentence (or within a paragraph) is a better yardstick of semantic relationship than just the incidental positional proximity of an entity phrase and an event phrase in the document text that may be the result of co-occurrence in two neighboring sentences or in two neighboring paragraphs (worsened for compound News documents containing multiple stories).
2. To each sentence s, we associate an abstraction (meaning), $M(s)$, a set of strings that best approximates the semantics of s. For specificity, let $M(s)$ be the set of stems obtained from s as follows: collect all the words in s into a set, eliminate the stop words, and then stem each word. The rationale is that this normalization better captures the semantics (that can be used to check for semantic similarity between sentences through purely syntactic manipulation). Observe that:
   – This removes overly discriminating word sequencing information from a sentence such as due to active/passive voice changes.
   – It is not doomed by the well-known problems associated with the bag-of-words model of documents because the focus is only on sentences, and not the entire document.

- The word sequencing information will be resuscitated in Section 2.2 while generating the final terse cluster labels.

Other alternatives that we regard as inferior to $M(s)$ are:

(a) $M_1(s)$ is the (non-contiguous) *sequence* of words of s obtained by eliminating the stop words. This normalization is inadequate because it is overly discriminating. For example, consider "The merger of America Online and Time Warner will create the world's largest media company" vs "The merger of Time Warner and America Online will create the world's largest media company". These two sentences do not match because Time Warner and America Online have been swapped.

(b) $M_2(s)$ is the *set* of words from s obtained by eliminating the stop words. This normalization is an improvement over $M_1(s)$ but it has its limitations when we consider semantics-preserving voice changes (active to passive, and, passive to active). For example, consider "Nokia acquires Intellisync." vs "Intellisync acquired by Nokia." vs "Nokia's acquisition of Intellisync." $M(s)$ is an improvement over $M_2(s)$ because stemming comes to rescue, treating "acquires", "acquired" and "acquisition" as equivalent. Recall that stemming reduces different forms of a word to the same root, and the risk of two semantically different words getting reduced to the same stem (using scalable Porter stemming algorithm) is minimal, in our limited context.

(c) $M_3(s)$ can be defined in such a way that two words are treated as equivalent if they have the same stems or have been asserted so via the codified domain knowledge (that is, in the Metathesaurus or through the concept definition). For example, consider "Nokia acquires Intellisync." vs "Nokia buys Intellisync." vs "NOK purchases SYNC." If the domain knowledge implies that "acquires", "buys", and "purchases" are synonymous in the "Mergers and Acquisition" context, all the three phrases are equivalent.

Note that $M_1(s)$ refines $M_2(s)$, $M_2(s)$ refines $M(s)$, and $M(s)$ refines $M_3(s)$.

3. To compute the support a document D_j accords to a sentence $s \in sen(D_i)$, we use the following scoring strategy:[2]

$$score(s, D_j) = \frac{MAX_{t \in sen(D_j)} |M(s) \cap M(t)|}{|M(s)|}$$

and for cumulative support score for sentence s due to the entire cluster

$$score(s) = \Sigma_{j=1}^{m} score(s, D_j)$$

Observe that:

- The support that a document D_j accords to a sentence s is proportional to the maximum number of overlapping stems in a D_j-sentence, scaled by the number of significant stems in s. As such, each document D_j can contribute at most 1 to the score for s.

[2] |...| is the set cardinality function.

- The appearance of significant stems of s in t is a plus. Also, $M(s) \subseteq M(t) \Rightarrow score(s) \geq score(t)$.
- If every document contains the same two significant sentences, then both sentences will garner equal score irrespective of the document content or length. (In the context of "on-topic" News, this does not lead us astray.)
- However, if there is one document D that contains a sentence s with smaller number of significant stems than a sentence t, then the contribution from D to the overall score of s will be higher than that for t, due to the scaling with respect to the number of significant stems. That is, $M(s) \subset M(t) \Rightarrow score(s) > score(t)$.

Reconsider the example in Section 1, reproduced below for convenience.

S0: Nokia acquires Intellisync.
S1: Intellisync to be acquired by Nokia.
S2: Nokia's (NOK) acquisition of Intellisync (SYNC) will not change the overall picture for company, says Greger Johansson at Redeye in Stockholm.
S3: The acquisition of Intellisync supports Nokia's goal to be the leader in enterprise mobility and enhances the ability of its customers to connect devices to data sources, applications and networks.
S4: Nokia and Intellisync have signed a definitive agreement for Nokia to acquire Intellisync.
S5: Nokia's Intellisync buy.
S6: Nokia's purchase of Intellisync.

According to our scoring criteria, the sentences [S0:] and [S1:] are treated as equivalent. They are supported by [S2:], [S3:] and [S4:], and to a much lesser degree by [S5:] and [S6:]. Furthermore, [S0:] and [S1:] score higher than [S2:], [S3:] and [S4:] on the basis of their "scaled" length. All this seems reasonable because we prefer a sentence that has strong "verbatim" support from document sentences of the cluster.

4. A well-supported sentence s for cluster label for the cluster of documents CD is the one that has the maximum cumulative support score.

$$candidate?(s) = \forall t \in \cup_{j=1}^{m} sen(D_j) : score(s) \geq score(t)$$

$$well\_supported\_sentences(CD) = \{\ s \in \cup_{j=1}^{m} sen(D_j)\ \mid\ candidate?(s)\ \}$$

2.2 Phrase Selection for Cluster Label

A *candidate cluster label* is the shortest sequence of words (in terms of string length) of a sentence in a document that contains a phrasal reference to the entity EN, the event EV, and significant words that appear in all well-supported sentences.

$$common\_stems(CD) = \cap\ \{M(s)\ \mid\ s \in well\_supported\_sentences(CD)\}$$

$$label\_pool(CD) = \{ss\ \mid\ \exists s \in well\_supported\_sentences(CD) \wedge substring(ss, s)$$

$$\wedge \; [\, \forall t \in common\_stems(CD) : \; substring(t, ss) \,] \\ \wedge \; preceded\_and\_followed\_by\_delimiter(ss) \\ \wedge \; contains\_entity\_event\_reference(EN, EV, ss) \}$$

Note that

- $common\_stems(\_)$ check has been incorporated to extract meaningful additional information to highlight,
- $preceded\_and\_followed\_by\_delimiter(\_)$ check (that determines whether the preceding and succeeding character is a blank or a punctuation mark) has been incorporated to generate a sequence of words as opposed to some substring, and
- $contains\_entity\_event\_reference(\_)$ check has been incorporated to ensure that the label contains some concrete reference to queried entity and event. This check can be as simple as a verbatim match to as complex as requiring alias resolution (for example, involving acronyms, coreferences, etc).

$$candidate\_cluster\_labels(CD) \;=\; \{p \in label\_pool(CD) \;\mid\; \forall q \in label\_pool(CD) : \; |p| \le |q|\}$$

A *cluster label* can be any one of the candidate cluster labels.

Observe that the cluster label is required to be a *contiguous* sequence of words from a well-supported sentence of a document in the cluster, and that it is not unique in general. For example, if $label\_pool(CD)$ contains “Nokia acquires Intellisync”, “Intellisync to be acquired by Nokia”, “Nokia’s (NOK) acquisition of Intellisync (SYNC)”, “acquisition of Intellisync supports Nokia’s”, and “Nokia to acquire Intellisync”, then “Nokia acquires Intellisync” is the chosen cluster label. For the dataset containing {S0, S5, S6}, each of S0, S5 and S6 is in $label\_pool(CD)$ but only S5 can be the cluster label.

Other alternative is to define *cluster label* as a shortest label (in terms of number of words) among those in the cluster label pool. For the first example, “Nokia acquires Intellisync” again wins the cluster label competition. For the dataset containing {S0, S5, S6}, both “Nokia acquires Intellisync” and “Nokia’s Intellisync buy” are equally acceptable as cluster labels, while “Nokia’s purchase of Intellisync” is rejected (on the feeble grounds that it contains the stop word “of”). For the dataset containing {S2, S3, S4}, “Nokia to acquire Intellisync” is the selected cluster label.

3 Selection of Cluster Labels from Headlines Alone

For proprietary reasons, only the metadata associated with a News document that includes the headline may be available, instead of the entire document or the APIs for extracting metadata from sentences/documents. Cluster labels can then be generated from headlines alone by adapting the above solution.

Problem Restatement: Consider a cluster of documents with headlines $\{D_1, D_2, ..., D_m\}$ for an entity EN and an event EV. Extract a *well-supported* headline from the documents of the cluster as follows.

Modified Approach

1. For each $i \in 1 \ldots m$, let h_i refer to the headline of the document D_i.
2. Similarly to the approach discussed in Section 2.1, to each headline h, we associate an abstraction, $M(h)$, a set of strings that best approximates the semantics of h. *Define $M(h)$ as the set of stems obtained from h by collecting all the words in h into a set, eliminating the stop words, and then stemming each word.* This normalization removes overly discriminating word sequencing information from a headline, and generates word forms that are better amenable to syntactic manipulation for gleaning semantics.
3. To compute the support a document D_j accords to a headline h_i, we use the following scoring strategy:

   $$score(h_i, D_j) = \frac{|M(h_i) \cap M(h_j)|}{|M(h_i)|}$$

   and for the cumulative support score for headline h_i due to the entire cluster

   $$score(h_i) = \Sigma_{j=1}^{m} score(h_i, D_j)$$

   Observe that: $M(h_i) \subseteq M(h_j) \;\Rightarrow\; score(h_i) \geq score(h_j)$.
4. A well-supported headline h for cluster label is the one that has the maximum cumulative support score.

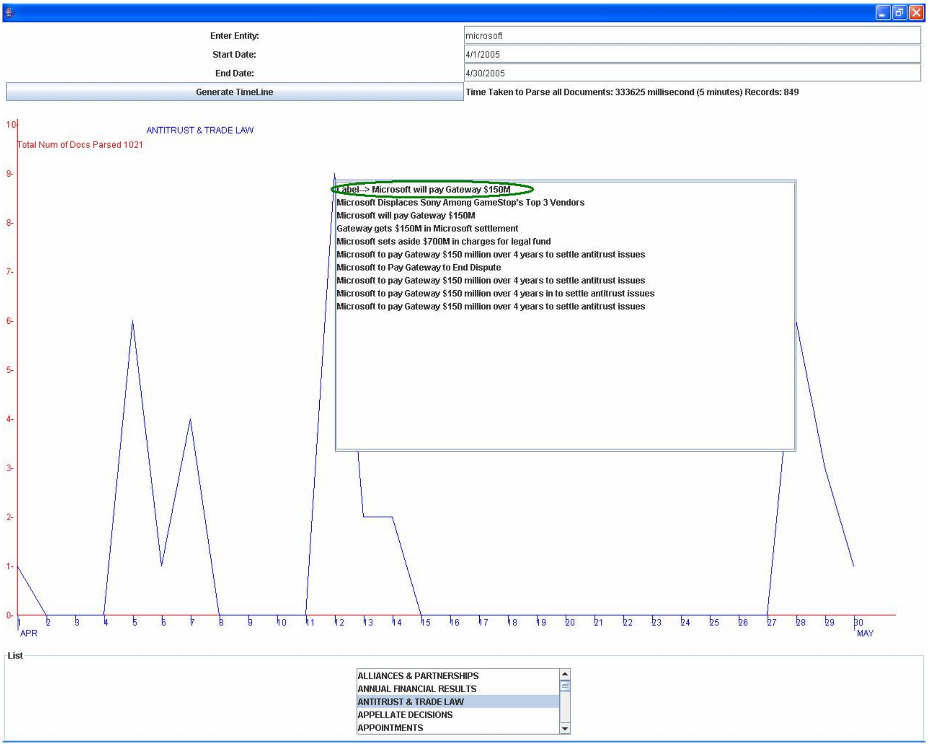
   $$well\_supported\_headlines(\cup_{j=1}^{m}\{D_j\}) = \{h \mid \forall j \in 1 \ldots m :\ score(h) \geq score(h_j)\}$$

   Any one of the well-supported headlines can be used as a cluster label. Note that in entity-event-duration timeline application, there is usually a unique candidate headline for a cluster of documents involving an entity and an event on a day because several news stories are correlated.

4 Application Context and Implementation Details

We have implemented an entity-event-duration timeline application in Java 5. The application pre-processes a year's worth of metadata-tagged News stories (provided in the form of XML documents) (150GB), indexing them for efficient access. The application takes an entity, an event, and a time-duration, and generates a timeline based on the number of News stories involving the entity and the event. As depicted in Figure 1, for each date, the GUI can pop-up a listbox showing the headlines of all the relevant documents and a generated cluster label. (It can also display the contents of a chosen News document.) The prototype works acceptably in practice, and we are investigating quantitative metrics to evaluate such systems in the absence of standard benchmarks or human analysts.

We now discuss several concrete examples to bring out the nature of the News documents and the behavior of our prototype. (The examples were chosen to be realistic as opposed to idealistic.) For example, on *April 12, 2005*, for the entity *Microsoft*, and for the event *Computer Operating Systems*, the generated headline cluster label is: *In next Windows release, Microsoft to use hardware for security*, based on the headlines:



Enter Entity: microsoft

Start Date: 4/1/2005

End Date: 4/30/2005

Generate TimeLine

Time Taken to Parse all Documents: 333625 millisecond (5 minutes) Records: 849

Label--> Microsoft will pay Gateway $150M

Microsoft Displaces Sony Among GameStop's Top 3 Vendors

Microsoft will pay Gateway $150M

Gateway gets $150M in Microsoft settlement

Microsoft sets aside $700M in charges for legal fund

Microsoft to pay Gateway $150 million over 4 years to settle antitrust issues

Microsoft to Pay Gateway to End Dispute

Microsoft to pay Gateway $150 million over 4 years to settle antitrust issues

Microsoft to pay Gateway $150 million over 4 years in to settle antitrust issues

Microsoft to pay Gateway $150 million over 4 years to settle antitrust issues

Fig. 1. Entity-Event-Duration Timeline Application

Microsoft unveils more details of next Windows release
In next Windows release, Microsoft plans to use hardware to lock down security
In next Windows release, Microsoft to use hardware for security
Microsoft ships Windows for 64-bit computers
Microsoft Gives Details on Windows Release
New Windows Operates on 64-Bit Computers
Microsoft ships Windows for 64-bit computers
In next Windows release, Microsoft to use hardware for security
Microsoft unveils more details of next Windows release
Microsoft ships Windows for 64-bit computers
Microsoft unveils more details of next Windows release
In next Windows release, Microsoft will use hardware for security
Microsoft plans to use hardware to lock down security in Windows
Microsoft ships Windows for 64-bit computers
In next Windows release, Microsoft to use hardware for security
Gates shows off features of next-generation Windows system

Our criteria effectively chooses the most frequent “short” headline such as *In next Windows release, Microsoft to use hardware for security* (or *In next Windows release, Microsoft will use hardware for security*) as the cluster label, while ignoring other headlines such as *Microsoft ships Windows for 64-bit computers.* (The majority criteria was chosen to eliminate "noise".) Several documents share a headline due to correlated News sources (such as Associated Press, Reuters, AFX News, etc.). Each such document can be viewed as providing an independent endorsement. Unfortunately, this approach can miss multiple headlines for different News stories that happen to have the same event and entity metadata tags, and occur on the same day. In fact, the "best" comprehensive headline for the above example is: *Microsoft unveils more details of next Windows release.* So, our approach can be further improved by clustering headlines on the basis of similarity, or ranking headlines on the basis of support and cutoff thresholds.

The other approach to cluster label generation elects a significant sentence from the cluster documents and clips it. For example, on *April 4, 2005*, for the entity *BHP Billiton*, and for the event *Takeovers*, the relevant document sentences from three separate News documents are:

1. Anglo-Australian resources giant BHP Billiton has been given the green light by Treasurer Peter Costello for its $9.2 billion takeover of Australian miner WMC Resources.
2. WMC had been the focus of a hostile takeover by Swiss-based Xstrata that had gained attention from the government backbench before BHP put in its bid.
3. Mr Costello, who had the ability to block the takeover or set impossible restrictions, only set two conditions on BHP and its proposal, both relating to uranium.
4. BHP chief executive Chip Goodyear welcomed the decision, saying the treasurer's conditions were acceptable and the company would abide by them.
5. BHP has offered $7.85 for each WMC share, with the takeover bid due to close at 7.30 pm (AEST) on May 6.

1. The federal government had raised no objection to the proposed takeover of WMC Resources by BHP Billiton, Treasurer Peter Costello said today.
2. In a statement, Mr Costello set two conditions for the proposed $9.2 billion takeover of WMC by BHP Billiton.

1. BHP Billiton chief executive Chip Goodyear welcomed the government's approval of the WMC bid.
2. The company said the conditions attached to the announcement by the Treasurer today were acceptable to BHP.

The well-supported sentence to summarize the cluster is: *In a statement, Mr Costello set two conditions for the proposed $9.2 billion takeover of WMC by BHP Billiton.*, yielding the cluster label: *takeover of WMC by BHP Billiton.*

Our cluster label generator can also implement *contains_entity_event_reference*(_) using tagging APIs that looks for co-occurrence of the queried terms in a paragraph or in a sentence, as opposed to using existing document-level XML tags. (We do not have license to use proprietary concept definitions (associations between metadata terms and document phrases employed by the tagger) to develop our own tagger.) To see the limitations of the current sentence-based approach, it is instructive to consider cluster labels generated from the 2005 News dataset given below. Note that we have intensionally excluded headlines to see how reliable document sentences are in yielding suitable cluster labels. If the headlines were included, they seem to dominate the cluster labels for obvious reasons.

Entity	Event	Date	Cluster Label
Toyota	Automotive Sales	June 3, 2005	Toyota Motor Corp. posted a 0.5 percent sales drop to 201,493 units
Google	Mergers & Acquisition	April 22, 2005	Google, (GOOG) the No. 1 search engine, said its first-quarter profit
Google	Internet & WWW	June 22, 2005	Google, which depends upon online
Google	Search Engine	June 22, 2005	Google to sell content through its search engine
Google	Online Advertising	June 22, 2005	Google will develop another source of revenue besides online advertising
Sprint	Mergers & Acquisition	June 2, 2005	Sprint Corp. shareholders are expected to vote in early July on the company's planned merger

In order to see the reliability difference between paragraph level vs sentence level co-occurrence for inferring associations, consider the document for entity *Microsoft* and event *Mergers & Acquisition* on *April 19, 2005* containing the fragment:

> . . . Amazon already offers e-books and more than 1 million e-documents on its site, using downloadable software from *Microsoft Corp.* and Adobe Systems Inc. The *purchase* of Mobipocket will allow Amazon to use its own software to diversify product distribution methods, rather than relying on third-party providers. . . .

The indexing metadata tags *Microsoft* and *Mergers & Acquisition* are associated with the phrases 'Microsoft' and 'purchase'. If document-level or paragraph-level co-occurrence of phrases is used for inferring associations, we get a false positive. As the phrases appear in successive sentences, sentence-level co-occurrence can improve reliability.

5 Related Work

Our entity-event-duration timeline application resembles Google Trends [11] which tries to determine relative interest in a topic on the basis of the number of searches on the topic (Search volume graph) and the number of News stories involving the topic (News reference volume graph). It also summarizes search query distribution in terms of their geographical location of origination (such as city, country, etc), language of the search query, etc. The spikes in search volume graph are further annotated by the headline of an automatically selected Google News story written near the time of that spike. Our entity-event timeline interface allows you to display all the News documents for the year 2005 that carry the corresponding entity and event index terms (with scores higher than a programmable cutoff).

Our work on Timeline generation can be viewed as a means to cluster search results using temporal attribute which happens to be the News story creation date [1]. Our label generation work is related to Vivisimo's post-retrieval tagging that provides more meaningful labels than those found in general tagging vocabularies [16].

Several proposals use frequest-item sets to derive labels [2,6]. Even though these approaches have a more general appeal, our approach provides more comprehensible and comprehensive labels in the context of News documents because it takes into account the semantics of the words using encoded domain knowledge, and the sequencing of the words by extracting from documents a concise phrase containing the "frequent items" to serve as the cluster label.

The techniques for cluster label generation described in [3,4,7,8,9] deal with the problem of abstracting sequencing information to improve precision and to rank labels in the general text documents context. Our approach addresses this issue by first using set-based abstraction to deal with semantic equivalence problem, and eventually restore word sequencing information to arrive at palatable labels in the more restrictive News documents query results set context.

The cluster description formats discussed in [10] are similar in spirit to our work but it deals with clusters of numerical records rather than text documents.

QCS information retrieval system [5] extracts documents relevant to a query, clusters these documents by subject, and returns summary of a cluster with the corresponding list of documents. This was originally developed for newswire documents, but has been used on Medline abstracts too. Our approach differs from QCS in that our focus is on terse cluster label generation that is indicative of the content of the cluster, rather than produce multi-document summary.

6 Conclusions and Future Work

We proposed a strategy for deriving sentence fragments from documents text that can serve as a cluster label for result set of a search query, specifically in the context of News documents involving queries centered around entities, events, and their relationships. The cluster label also provides additional information that serves as "answer" or "missing detail" relevant to the query. Even though our technique abstracts the meaning of a sentence as a set of stems of significant words in the sentence, it nicely incorporates preference for shorter labels and resuscitates word sequencing information for the label eventually. Thus, the final cluster labels are adequate, informative, and grounded in the document text, even though there are several examples in which they seem rather long. This approach also has potential to serve as a framework for generalizing or specializing clusters into an hierarchy. Currently, we are studying reliability, efficiency, and scalability of the entity-event timeline visualization application.

Acknowledgements

We wish to thank Don Loritz for enlightening discussions throughout this project.

References

1. Alonso, O., Gertz, M.: Clustering of Search Results using Temporal Attributes. In: Proceedings of 29th ACM SIGIR Conference, pp. 597–598 (2006)
2. Beil, F.F., Ester, M., Xu, X.: Frequent term-based text clustering. In: Proceedings of the 8th International Conference on Knowledge Discovery and Data Mining (KDD-2002), pp. 436–442 (2002)
3. Campos, R., Dias, G.: Automatic Hierarchical Clustering of Web Pages. In: Proceedings of the ELECTRA Workshop with 28th ACM SIGIR Conference, pp. 83–85 (2005)
4. Del Corso, G.M., Gulli, A., Romani, F.: Ranking a stream of news. In: Proceedings of 14th International World Wide Web Conference, Chiba, Japan, pp. 97–106 (2005)
5. Dunlavy, D., Conroy, J., O'Leary, D.: QCS: A Tool for Querying, Clsutering, and Summarizing Documents. In: Proceedings of HLT-NAACL, pp. 11–12 (2003)
6. Fung, B.C.M., Wang, K., Ester, M.: Hierarchical document clustering. In: Wang, J., (ed.) Encyclopedia of Data Warehousing and Mining, Idea Group (2005)
7. Ferragina, P., Gulli, A.: The anatomy of a hierarchical clustering engine for web-page, news and book snippets. In: Proceedings of the 4th IEEE International Conference on Data Mining, pp. 395–398 (2004)
8. Ferragina, P., Gulli, A.: A personalized search engine based on web-snippet hierarchical clustering. In: Proceedings of 14th International World Wide Web Conference, pp. 801–810 (2005)
9. Gulli, A.: The anatomy of a news search engine. In: Proceedings of 14th International World Wide Web Conference, pp. 880–881 (2005)
10. Gao, B.J., Ester, M.: Cluster description formats, problems and algorithms. In: Proceedings of the 6th SIAM Conference on Data Mining (2006)
11. http://www.google.com/trends
12. http://www.lexisnexis.com/
13. http://jakarta.apache.org/lucene/docs/index.html
14. http://www.nlm.nih.gov/
15. Osinski, S., Weiss, D.: A concept-driven algorithm for clustering search results. IEEE Intelligent Systems 20, 48–54 (2005)
16. http://vivisimo.com/docs/tagging.pdf

Generating Ontologies Via Language Components and Ontology Reuse

Yihong Ding[1], Deryle Lonsdale[2], David W. Embley[1], Martin Hepp[3],
and Li Xu[4]

[1] Department of Computer Science, Brigham Young University, U.S.A.
{ding,embley}@cs.byu.edu

[2] Department of Linguistics, Brigham Young University, U.S.A.
lonz@byu.edu

[3] Digital Enterprise Research Institute (DERI), University of Innsbruck, Austria
martin.hepp@deri.org

[4] Department of Computer Science, University of Arizona South, U.S.A.
lxu@email.arizona.edu

Abstract. Realizing the Semantic Web involves creating ontologies, a tedious and costly challenge. Reuse can reduce the cost of ontology engineering. Semantic Web ontologies can provide useful input for ontology reuse. However, the automated reuse of such ontologies remains underexplored. This paper presents a generic architecture for automated ontology reuse. With our implementation of this architecture, we show the practicality of automating ontology generation through ontology reuse. We experimented with a large generic ontology as a basis for automatically generating domain ontologies that fit the scope of sample natural-language web pages. The results were encouraging, resulting in five lessons pertinent to future automated ontology reuse study.

1 Introduction

Ontology construction is a central research issue for the Semantic Web. Ontologies provide a way of formalizing human knowledge to enable machine interpretability. Creating ontologies from scratch is, however, usually tedious and costly. When the Semantic Web requires ontologies that express Web page content, the ontology engineering task becomes too expensive to be done manually. Many Semantic Web ontologies may have overlapping domain descriptions because many Web sites (or pages) contain information in common domains. It is inefficient to redo ontology engineering for pre-explored domains. These issues illustrate the importance of automated ontology reuse for the Semantic Web.

Ontology reuse involves building a new ontology through maximizing the adoption of pre-used ontologies or ontology components. Reuse has several advantages. First, it reduces human labor involved in formalizing ontologies from scratch. It also increases the quality of new ontologies because the reused components have already been tested. Moreover, when two ontologies share components through ontology reuse, mapping between them becomes simpler because

Z. Kedad et al.(Eds.): NLDB 2007, LNCS 4592, pp. 131–142, 2007.

mappings between their shared components are trivial. One can also simultaneously update multiple ontologies by updating their commonly reused components. Hence ontology reuse also improves the efficiency of ontology maintenance.

Despite the many advantages of (automated) ontology reuse, the topic is not well explored in the literature. There are many reasons for this. Before the advent of the Semantic Web, few ontologies existed. Due to the difficulty of constructing ontologies, as well as to the challenges of using ontologies in applications, researchers were less interested in ontology development. With the advance of Semantic Web technologies, the number of ontologies has significantly increased recently. When the use of ontologies in Semantic Web applications improves system performance, more people will realize the benefit of using ontologies. In the meantime, most existing ontologies are hard to reuse. The benefits of manual ontology reuse are often unclear since the overhead of seeking and understanding existing ontologies by humans may be even greater than simply building an ontology from scratch. At the same time, many existing ontologies simply do not support effectively automated ontology reuse. The corresponding information in these ontologies is hard to retrieve for automated ontology reuse.

The work we describe below[1] offers three contributions for automated ontology reuse. We first sketch the state of the art in ontology reuse (Section 2). We then present our generic ontology reuse architecture and our implementation (Section 3). Next, we discuss experimental results obtained by using our implementation on real-world examples, as well as five lessons we have learned from this work (Section 4). We conclude with possible future directions (Section 5).

2 Related Work

Ontology reuse has been studied for years. Most of the earlier research focuses on the study of reusable ontology repositories. In 2001, Ding and Fensel [7] surveyed these earlier ontology libraries. Due to the lack of ontologies, however, very few studies on practically reusing ontologies exist prior to this survey. Uschold and his colleagues [14] presented a "start-to-finish process" of reusing an existing ontology in a small-scale application. According to the authors, the purpose was a "feasibility demonstration only." They concluded that reusing an ontology was "far from an automated process" at that time.

With the growth of semantic web research, more and more ontologies have been created and used in real-world applications. Researchers have started to address more of the ontology reuse problem. Typically, there are two strands of study: theoretical studies of ontology reusability [2,4,8], and practical studies of ontology reuse [1,12,13]. Previous studies of ontology libraries showed that it was difficult to manage heterogeneous ontologies in simple repositories. Standardized modules may significantly improve the reusability of ontologies. One major purpose of modular ontology research concerns the reusability of ontologies [2,4,8]. There are, however, fewer ontology reuse studies quantifying how modular ontologies may improve the efficiency of ontology reuse. Hence one of our purposes

[1] See also www.deg.byu.edu

is to argue for the use of modular ontologies in real-world, automated ontology reuse experiments.

Meanwhile, there are also several studies on practical ontology reuse. Noy and Musen [12] introduced "traversal views" that define an ontology view, through which a user can specify a subset of an existing ontology. This mechanism enables users to extract self-contained portions of an ontology describing specific concepts. Stuckenschmidt and Klein [13] described another process for partitioning very large ontologies into sets of meaningful and self-contained modules through a structure-based algorithm. Alani and his colleagues [1] coined a new term for reusing existing ontologies: ontology winnowing. The intuition of their research is that individual semantic web applications more profitably use smaller customized ontologies rather than larger general-purpose ontologies. They therefore described a method for culling out—which they called winnowing—useful application-specific information from a larger ontology.

A common implicit assumption in all these practical ontology reuse studies is that source ontologies must be reusable for a target domain. Although this assumption simplifies the problem, it does not address the general situation. Besides our work, to the best of our knowledge, the only research that has addressed (albeit implicitly) the domain-specific ontology reuse problem is by Bontas and her colleagues [3]. Their case studies on ontology reuse identified difficulties due to end-user unfamiliarity with the complex source structure. Although this assessment is reasonable, we found a further reason for the difficulty they encountered. Even though source ontologies often declare a target domain, the corresponding information is irretrievable for automated ontology reuse. This is the real bottleneck for automated ontology reuse.

3 Automated Ontology Reuse

Figure 1 shows our generic architecture for automated ontology reuse. The reuse procedure takes at least two inputs: natural language (NL) documents and source ontologies. NL documents express the projecting domain and they can encompass different types. Typical NL documents could include collections of competency questions [15] or collections of sample Web pages [5].

In this architecture, ontology reuse consists of three sequential steps: concept selection, relation retrieval, and constraint discovery. These correspond to the three fundamental components in ontologies: concepts, relationships, and constraints. The concept selection process identifies reusable ontology concepts from source ontologies based on the descriptions in NL documents. NL documents must contain sufficient information for a system to identify all the necessary domain concepts. The identification methodologies vary with respect to different types of NL documents. The relation retrieval process retrieves relationships among selected concepts from the previous step. These relationships can be automatically gathered from source ontologies or perhaps even (optionally) recoverable from the NL documents. Figure 1 represents these optional requirements with dotted lines. The constraint discovery process discovers constraints for

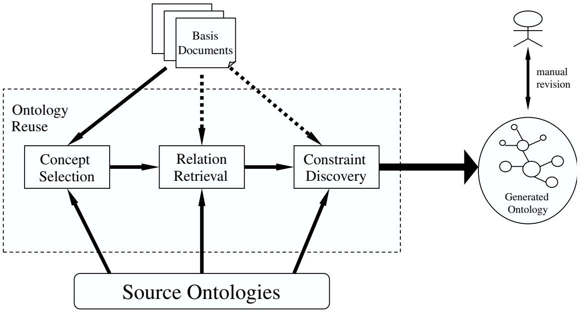


Fig. 1. Generic Architecture of Automated Ontology Reuse

previous selected concepts and relationships. An ontology reuse system should be able to gather existing information about constraints from source ontologies or even perhaps from NL documents.

After these three sequential steps, the system composes the selected concepts, relationships, and constraints together into a unified ontology. Human experts then inspect and revise these auto-generated ontologies.

We have implemented a prototype automated ontology reuse system based on this generic architecture. Our system reuses existing ontologies to create small domain ontologies within the scope of describing individual Web pages. In the rest of this section we describe the system in fuller detail.

Preparation of Input. We first take a small set of sample Web pages as input NL documents, and pre-process them to focus on the domain of interest (i.e. removing advertisement and side bars). Only the main body of each page remains, which constitutes the focus of interest for readers.

Proper preparation of source ontologies is essential for ontology reuse automation. Poorly integrated source ontologies create a very complex ontology integration problem during final composition of the output ontology. Two options exist: either we can directly adopt a single large-scale ontology, or we can manually pre-integrate several small ones. For simplicity, we chose the first option (and will discuss the second option later). Specifically, we adopted the MikroKosmos (μK) ontology [11], a large-scale ontology containing more than 5000 hierarchically-arranged concepts (excluding instances). These concepts cover various domains, a desideratum for flexible experimentation. The μK ontology has an average of 14 inter-concept links/node, providing rich interpretations for the defined concepts.

To automate our ontology reuse process, we pre-integrated the leaf concepts from the μK ontology with external lexicon dictionaries and declarative data recognizers, as Figure 2 shows. These augmentations are essential for automated ontology-concept recognition. Most of these lexicons and data recognizers are collected from the Web. For example, for the ontology concept CAPITAL-CITY we used a web browser to locate lists of all the capital cities of the independent countries in the world. Since we collected information from varied resources, we found that synonym identification became critical for the performance of

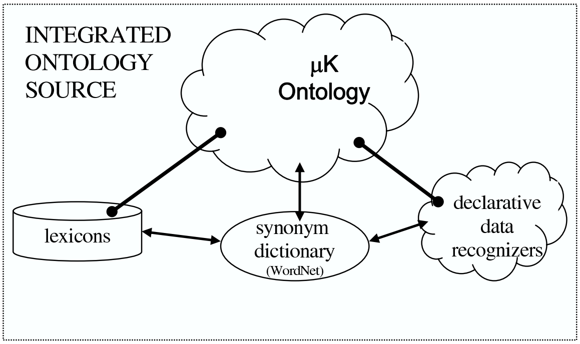

INTEGRATED ONTOLOGY SOURCE
μK Ontology
lexicons
synonym dictionary (WordNet)
declarative data recognizers

Fig. 2. Pre-Integrated Source Ontology

ontology reuse. We therefore adopted WordNet[2] for our synonym resource in this work; see [10] for related work involving a hierarchical terminology resource.

Although this source ontology preparation process is quite involved, it is a one-time effort. This integrated ontology source thus become static and constant for all downstream ontology reuse applications.

Ontology Reuse Process. Figure 1 shows how our system extracts an appropriate sub-domain from a larger, integrated source ontology by executing concept selection, relation retrieval, and constraint discovery. Since any ontology can be viewed as a conceptual graph, our algorithm is implemented to find nodes, edges, and specific constraints in graphs.

(1) Concept selection: We have implemented the concept-selection process as two concept-recognition procedures (which could be executed in parallel) and a concept-disambiguation procedure. In particular, the two recognition procedures involve concept-name matching and concept-value matching. Concept-name matching associates content in NL documents with concept names in source ontologies. For example, this procedure matches the word "capital" in the sentence "Afghanistan's capital is Kabul and its population is 17.7 million." to the ontology concepts CAPITAL-CITY and FINANCIAL-CAPITAL. The system hence associates both of the concepts as candidate concepts in the target domain. Concept-value matching associates content in NL documents with concept recognizers that have been pre-integrated during the source-ontology preparation stage. For example, in the same sentence above, this procedure matches the word "Kabul" with the concept CAPITAL-CITY.

The concept disambiguation procedure follows the previous two recognition procedures. Since the source ontology contains thousands of concepts whereas the target domain may only contain dozens, we often encounter considerable ambiguity on selected concept candidates. In the previous example the system recognizes both CAPITAL-CITY and FINANCIAL-CAPITAL as matched for a common data instance. Concept disambiguation solves this type of problem.In

[2] http://wordnet.princeton.edu/

our example, the system knows that CAPITAL-CITY and FINANCIAL-CAPITAL cannot both be valid candidates because the word "capital" should have one and only one meaning in the sentence. At the same time, since the concept-value procedure recognizes another instance "Kabul" of CAPITAL-CITY, but no more instances of FINANCIAL-CAPITAL, the system accepts CAPITAL-CITY and eliminates FINANCIAL-CAPITAL. We have implemented four concept disambiguation rules [5].

(2) Relation retrieval: The crux of the relation retrieval process is about finding appropriate edges between two concept nodes. An obvious resolution is to find all possible paths between any two candidate concepts. It would be easier for users to reject inapplicable edges rather than to add new relations. But this resolution has a serious performance difficulty. To find all paths between two graph nodes NP-complete.[3] Hence we must seek an alternative resolution.[4]

Different paths in an ontology graph refer to relations with different meanings. From our studies we have found that in general a shorter path represents a closer or more popular relationship between two concepts. On the contrary, an extra-long path often means a very uncommon relation between the two concepts within the domain. Hence it is reasonable to set a threshold length to reduce the search space and thus the complexity of the edge-searching algorithm.

In the implementation, we adapted the well-known Dijkstra's algorithm. Although the original algorithm computes only the shortest path, it can be easily extended by repeatedly computing the next shortest path until a threshold length is reached. Since Dijkstra's algorithm has polynomial time complexity and the threshold length is fixed and finite, the time complexity of this updated algorithm is also polynomial.

After this edge-searching procedure, the system perform a subgraph-detection procedure to finalize the target domain. Quite often, the edge-searching procedure results in multiple unconnected subgraphs. Normally, two separate subgraphs represent two independent domains. We simply assume that the largest subgraph contains the majority of concepts of interest, and thus the system keeps only the largest generated subgraph. By rejecting concepts that are not in the selected subgraph, we further improve accuracy of domain recognition.

(3) Constraint Discovery. Ontologies involve numerous types of constraints; this paper cannot possibly enumerate them all or discuss the methods for reusing constraints. For our purpose in demonstrating automated ontology reuse, we limited our study to cardinality constraints and their discovery. Unlike many other constraints in ontologies, cardinality constraints contain quantitative scales, which render the automatic discovery process particularly interesting.

[3] Finding the longest path between two graph nodes is a well-known NP-complete problem [9]. If we could solve finding all paths in polynomial time, by sorting the results in polynomial time, finding the longest path could also be solved in polynomial time—a contradiction.

[4] A reviewer rightly suggests that the input NL documents might help in relation retrieval, but we expect the NL processing would likely prove prohibitively costly.

We have implemented a cross-counting algorithm to discover cardinality constraints from NL documents.[5] Each cardinality constraint consists of a pair with a minimum number and a maximum number $[min : max]$. The cross-counting algorithm counts the instantiated numbers of paired concepts, from which the system can decide these minimum and maximum numbers. For example, suppose that in document $D1$ concept A is instantiated by $a1$, and there are no instantiations for concept B in the same document. In another document $D2$, however, concept A is instantiated by the same $a1$ and concept B is instantiated by $b2$. With these two documents, we can determine that the minimum cardinality constraint of concept A to the relation AB is 0 because for an instance $a1$ of A, it may not always have an instance of B appearing at the same time. The details of this algorithm are presented elsewhere [5].

Ontology Refinement. After finding concepts, relations, and constraints, composing them together into an ontology is straightforward. The result probably will not precisely describe the target domain. There are four basic human operations for revising the components in an automatically generated ontology: (1) remove the unexpected, (2) rename the inappropriate, (3) modify the incorrect, and (4) add the missing. In general, a preferred ontology reuse procedure will produce outputs requiring less revision operations on (3) and (4), especially the latter. It is in general much easier for users to reject unexpected components than to add something totally new into an ontology by themselves. Based on this refinement perspective, our ontology reuse system preserves as much useful information as possible, minimizing the need for addition by users.

4 Experiments and Discussions

This section describes a series of experiments with our ontology reuse system; full details on the experimental methodology, results, and evaluation are available in [7]. We cast our discussion in five lessons that we believe are pertinent to future automated ontology reuse studies.

Lesson 1. Ontology coverage is best specified by the leaf concepts.

For ontology reuse, the coverage of an ontology is the reusable domain described by an ontology. Users for ontology reuse would be justified in believing that we can straightforwardly determine the coverage of an ontology by its root definition. For example, when the root concept of an ontology is *Book*, this ontology should cover the domain of books; when the root concept is FINANICAL REPORT, this ontology must cover the domain of financial reports. Since the root of the μK ontology is *ALL* (i.e. everything), as naïve users we began our study believing that we could reuse the μK ontology to describe arbitrary domains.

Our initial experiments produced disappointing output. Usually we either got no result or the composed ontologies were outside the expected domains. Careful study of the results located the problem: the real coverage of an ontology is not

[5] The original μK ontology does not contain information about cardinality constraints.

determined by its root definition. Although theoretically the root definition of an ontology should be an abstract characterization of the entire domain, often ontology developers do not properly circumscribe the domain and thus a significant portion of the domain is often not reusable. Instead, the real reusable domain of an ontology (i.e. the real coverage of an ontology) is primarily determined by the union of its leaf-level concepts, a subset of the root-specified domain. For example, if a *NATION* ontology contains leaf-level concepts like *USA*, *Russia*, *China*, *Australia*, etc., but lacks *Montenegro*, the concept *Montenegro* is not reusable with respect to this ontology. This observation is fairly simple though critical for ontology reuse research; interestingly, previous ontology reuse publications miss this point.

Lesson 2. Extend ontology coverage with lexicons and data recognizers.

To improve the degree of reusability of existing ontologies, we want to boost the coverage of an ontology so that it is closer to its root definition. We refer to this as the "applicable coverage" of an ontology, where the term "applicable" means the new concepts can be evaluated by an ontology reuse program.

To boost the applicable coverage of our source ontology during the source-ontology preparation stage, we associated lexicons and data recognizers with the leaf-level concepts. We have named the result "instance recognition semantics", or formal specifications that identify instances of a concept C in ordinary text [6]. These are essential to automating ontology reuse.

We further populate the ontology with some upper-level ontology concepts. For example, prior to June 3, 2006 Montenegro was not an independent nation, so the original μK ontology did not have a leaf concept *Montenegro* under *NATION*. This portion of the ontology becomes non-reusable for many situations involving Montenegro after June 3, 2006. It is a very complicated issue to get permissions and then properly modify an ontology that is created by somebody else. For the purpose of automated reuse, however, we developed a simple and effective (though imperfect) alternative. We simply bind a lexicon set to the non-leaf concept *NATION*, thus adding the name of Montenegro into the lexicon after June 3, 2006. Although we still have not formally specified Montenegro as a country in the ontology, we have rendered the original source ontology reusable for situations involving the new country Montenegro. In the new generated ontology, instead of a specific concept *Montenegro* as an independent nation, we can correctly generate an upper-level concept—*NATION*, and thus all the properties of *NATION* become applicable in this new generated domain ontology. With such a technique, we artificially boost the applicable coverage of the source ontology.

In our experiments we augmented lexicons and data recognizers for leaf-level concepts in the μK ontology and their superclasses up to 2 levels above (on average). The union of these augmented concepts and their relations composes the applicable coverage of the source ontology in our experiments.

Lesson 3. For known target domains, ontology reuse is already possible and even valuable.

After having prepared the source ontology, we started our real experiments. Based on Lesson 1, we decided to focus our experiments on several selected domains rather than on arbitrary domains. We want human inspection to assure that the projecting domains have significant overlap with the applicable coverage of our source ontology. In particular, we chose experiments in three narrow domains: car advertisements, apartment rentals, and nation descriptions. This paper only briefly summarizes our results; see [5] for details.

First we list some basic settings and statistics of our experiments. Each of the three target domains contains a dozen to twenty concepts. For each domain, we feed four to seven cleaned sample Web pages (NL documents) to the ontology reuse system. The source ontology has been pre-integrated and augmented by its applicable coverage. In order to evaluate the performance of our outputs, we had human experts separately create ontologies for each target domain. We adopted the human-created ontologies as a gold standard to which the automatically generated ontologies were compared for precision and recall.

In general, we obtained low precision results. In the three target domains, the best precision was 48% for concept generation, 14% for relation generation, and 10% for cardinality constraint generation. The news is not all bad. Low precision implies the need for more rejections of corresponding components within a generated ontology. For humans, as mentioned earlier, rejecting inappropriate ontology components is much easier than adding new ontology ones. Hence our strategy is to favor greater recall values (i.e. less addition) over greater precision values (i.e. less rejection).

We updated the traditional recall calculation equation as follows:

```
updated recall = # correctly-reused / # existing-in-source
```

where the numerator is the number of component types (i.e. either concept, relationship, or constraint) correctly reused in a generated ontology; the denominator is the number of component types contained in input sources (both from NL documents and source ontologies). We use this formula because not everything defined in the human-created ontology is also identifiable by the inputs. For example, human experts have defined a concept *FEATURE* in the car-ads ontology, a concept missing from the source μK ontology. Hence it is impossible for a system to reuse an non-pre-existing concept. To be more equitable, our recall calculation must eliminate this type of error.

With the new formula, in the three testing domains our worst recall values were 83% (concept generation), 50% (relation generation), and 50% (cardinality constraint generation). All the best recall values were close or equal to 100%. Our ontology reuse system performs quite well even though it still is a prototype. The recall values show that we may reduce at least half of the human effort in ontology construction through ontology reuse when a target ontology is properly contained in the applicable coverage of the source ontology. Considering the expense of training professional ontologists and the time they need to build and tune ontologies, 50% already represents substantial savings. There are many ways to further improve the performance of the system. Already, though, our experiments demonstrate that ontology reuse is no longer "far from an automated process" [14].

Lesson 4. Ontology modularization facilitates automated ontology reuse.

During our experiments, another metric studied was running time. In general the system took about 1000 seconds to resolve all the ontology components with respect to about 50 to 100 candidate concepts on a Pentium 800 MHz single processor machine. This execution time is rather short compared to the time required for manually creating an ontology of the same scale. Our benchmark showed that almost 90% of execution time was spent on the relation retrieval process. Though we may further improve this time performance by optimizing our implementation, the problem lies mainly in the magnitude of the source ontology (over 5000 concepts and over 70000 relationships to explore).

Reducing the execution time of relation retrieval should be possible by using modular ontologies rather than a single large-scale one. Modular ontologies are usually small and designed to be self-contained. An ontology module is self-contained if all of its defined concepts are specified in terms of other concepts in the module, and do not reference any other concepts outside the module. As soon as several major concepts in a module are selected as candidate concepts, an ontology reuse system may decide to directly reuse the entire module rather than perform a costly relation retrieval algorithm. Hence the execution time for relation retrieval can be significantly reduced.

To pursue this issue, we manually pruned several comparatively independent clusters of ontology components from our source ontology and used them as individual modules. Since these clusters originated from a previously unified ontology, we did not need to further integrate them. The same experiments were re-run with these multiple "modular" ontologies. On average the system took less than 300 seconds—saving more than 70% of run time—to resolve all the ontology components for about 50 to 100 candidate concepts. Because these pruned clusters were not true, self-contained modular ontologies, the performance in terms of precision and recall decreased in this experiment. Saving execution time by replacing a large unified ontology with multiple small modular ontologies is thus a convincing strategy. By using really well-designed modular ontologies, our ontology reuse system achieves both higher precision and recall values, as well as faster run-time performance.

Lesson 5. Sample documents may help us mine "latent" knowledge from texts.

We also carefully studied our low-precision experimental results. Many reused concepts and relations were beyond the scope of the expert-created ontologies. Yet they were not all meaningless or useless. On the contrary, we found that useful information—latent in the document but beyond the topic directly at hand—could be gleaned from the results.

For example, we have applied our tool to process some U.S. Department of Energy (DOE) abstracts. The expert who created a reference ontology was only interested in the generic information about these abstracts, such as the theme of a document, the number of figures and tables, etc. But our ontology reuse tool found much more useful information. For instance, in one sample abstract it generated some concepts and relations indicating that the crude oil price dropped in the year 1986. Although this was not what the human expert originally expected and it was outside the expert-specified domain of interest, we could not deny that this type of information could be very valuable.

Such latent information is not really what people cannot find. But they are easily overlooked by human readers, especially when reading through many such documents. Especially within the business domain, people want to mine this type of latent information from numerous financial documents and business news. We believe that the automated ontology reuse mechanism may provide the business community an alternative solution for seeking valuable latent information.

5 Conclusion

We have presented an automated ontology reuse approach. Although we only applied our system to reuse the μK ontology, our methodology supports automated ontology reuse in general. Informed by our experiments on real-world examples, we have summarized five lessons that are constructive for future exploration of ontology reuse studies. In essence, we conclude that ontology reuse is no longer “far from an automated process” [14].

In the meantime, a few critical problems remain to be solved. One is to automatically decide whether a target domain is within the reusable coverage of an integrated source ontology. If the majority of a target domain lies outside the source ontology, ontology reuse becomes nothing but extra overhead. Also, we need to experiment with applying modular ontologies for ontology reuse. Until now, the research of modular ontologies is still at the stage of theoretical analysis. We need practical study cases to push this research field forward. The study of instance recognition semantics should be paired with modular ontology research to improve the reusability of modular ontologies. Last but not least, mining latent information through ontology reuse is an interesting research topic. More exploration on this topic may bring many benefits to users, especially in the business domain.

So far there are few published studies on automated ontology reuse research. We hope that our results draw more attention to this field and facilitate wider public adoption of the Semantic Web.

Acknowledgements

This work was funded in part by U.S. National Science Foundation Information and Intelligent Systems grants for the TIDIE (IIS-0083127) and TANGO (IIS-0414644) projects. Part of the work was also supported by the European Commission under the projects DIP (FP6-507483), SUPER (FP6-026850), and MUSING (FP6-027097), by the Austrian BMVIT/FFG under the FIT-IT project myOntology (grant no. 812515/9284), and by a Young Researcher’s Grant from the University of Innsbruck.

References

1. Alani, H., Harris, S., O'Neil, B.: Ontology winnowing: A case study on the AKT reference ontology. In: Proc. Int'l Conference on Intelligent Agents, Web Technology and Internet Commerce (IAWTIC'2005), Vienna, Austria (November 2005)
2. Bao, J., Caraagea, D., Honavar, V.: Modular ontology – a formal investigation of semantics and expressivity. In: Proc. First Asian Semantic Web Conference (ASWC 2006), Beijing, China, September 2006 (In press)
3. Bontas, E., Mochol, M., Tolksdorf, R.: Case studies on ontology reuse. In: Proc. 5th Int'l Conf. on Knowledge Management (I-Know'05), Graz, Austria (2005)
4. Bouquet, P., Giunchiglia, F., van Harmelen, F., Serafini, L., Stuckenschmidt, H.: Contextualizing ontologies. Journal of Web. Semantics 1(4), 325–343 (2004)
5. Ding, Y.: Semi-automatic generation of resilient data-extraction ontologies. Master's thesis, Brigham Young University, Provo, Utah (June 2003)
6. Ding, Y., Embley, D., Liddle, S.: Automatic creation and simplified querying of semantic web content: An approach based on information-extraction ontologies. In: Mizoguchi, R., Shi, Z., Giunchiglia, F. (eds.) ASWC 2006. LNCS, vol. 4185, pp. 400–414. Springer, Heidelberg (2006)
7. Ding, Y., Fensel, D.: Ontology library systems: The key for successful ontology reuse. In: Proc. First Semantic Web Working Symposium (SWWS'01), Stanford, CA, (July 2001)
8. Grau, B., Parsia, B., Sirin, E., Kalyanpur, A.: Modularizing OWL ontologies. In: Proc. KCAP-2005 Workshop on Ontology Management, Banff, Canada (October 2005)
9. Hochbaum, D.: Approximation Algorithms for NP-Hard Problems. PWS Publishing Company, Boston, MA (1997)
10. Lonsdale, D., Ding, Y., Embley, D.W., Melby, A.: Peppering knowledge sources with SALT: Boosting conceptual content for ontology generation. In: Semantic Web meets Language Resources: Papers from the AAAI Workshop, Menlo Park, CA, pp. 30–36. Technical Report WS-02-16, AAAI Press, Stanford (2002)
11. Mahesh, K.: Ontology development for machine translation: Ideology and methodology. Technical Report MCCS-96-292, Computer Research Laboratory, New Mexico State Univeristy (1996)
12. Noy, N., Musen, M.: Specifying ontology views by traversal. In: Proc. Third International Semantic Web Conference (ISWC 2004), Hiroshima, Japan, pp. 713–725 (November 2004)
13. Stuckenschmidt, H., Klein, M.: Structure-based partitioning of large class hierarchies. In: Proc. Third International Semantic Web Conference (ISWC 2004), pp. 289–303, Hiroshima, Japan (November 2004)
14. Uschold, M., Healy, M., Williamson, K., Clark, P., Woods, S.: Ontology reuse and application. In: Proc. International Conference on Formal Ontology and Information Systems (FOIS'98), Trento, Italy, pp. 179–192 (June 1998)
15. Uschold, M., King, M.: Towards a methodology for building ontologies. In: Proc. Workshop on Basic Ontological Issues in Knowledge Sharing in conjunction with IJCAI-95, Montreal, Canada (1995)

Experiences Using the ResearchCyc Upper Level Ontology

Jordi Conesa[1], Veda C. Storey[2], and Vijayan Sugumaran[3]

[1] Estudis d'Informatica i Multimedia
Universitat Oberta de Catalunya
Rambla del Poblenou, 156
E-08018 Barcelona
jconesac@uoc.edu
[2] Department of Computer Information Systems
J. Mack Robinson College of Business
Georgia State University, Box 4015
Atlanta, GA 30302
vstorey@gsu.edu
[3] Department of Decision and Information Sciences
School of Business Administration, Oakland University
Rochester, MI 48309
sugumara@oakland.edu

Abstract. Repositories of knowledge about the real world and how it functions are needed to advance research in intelligent, knowledge-intensive systems. The repositories are intended to serve as surrogates for the meaning and context of terms and concepts. These are being developed at two levels: 1) individual domain ontologies that capture concepts about a particular application domain, and 2) upper level ontologies that contain massive amounts of knowledge about the real world and are domain independent. This paper analyzes ResearchCyc, which is a version of the most extensive base of common sense knowledge, the upper level ontology, Cyc. It does so to summarize the current state of the art in upper level ontology development in order to suggest areas for future research. The paper also describes various problems encountered in applying ResearchCyc to web query processing.

1 Introduction

As the Internet, social communities, and business globalization initiatives continue to expand, there is an increasing need for sophisticated software to support "intelligent" applications. These systems are knowledge intensive in that they require access to large amounts of domain knowledge to compliment public and organization-specific knowledge. Examples include those systems that support the Semantic Web, web queries, web services, heterogeneous databases and multi-agent systems. One way to support the development of sophisticated systems is through the incorporation of domain knowledge into systems design. Borrowed from their role in philosophy where they serve as general descriptions of what can exist in the world [3], domain ontologies specify concepts, relationships between concepts, and inference rules for

Z. Kedad et al.(Eds.): NLDB 2007, LNCS 4592, pp. 143–155, 2007.

an application domain (e.g. travel reservation, soccer, gourmet food). Ontologies, in general, are increasingly needed for software design, including information sharing among heterogeneous data sources [21], interpreting unstructured data on the Web [1], creating and evaluating conceptual models [22], web queries and others.

With the exception of the DAML ontologies (www.daml.org) for the Semantic Web, there is a severe lack of libraries of domain ontologies for general use. It has been suggested that it is both feasible and necessary to automate the development of domain ontologies [8]. Besides the efforts to develop individual ontologies, there has been great interest in developing upper level or large scale ontologies. These are intended to be domain independent and to provide a way to capture and represent the semantics of the real world in order to support knowledge intensive software applications. To make the ontologies useful, however, we need to understand the current state of the art and suggest how it might progress.

The most massive effort in developing an upper level ontology is the Cyc project [14]. Cyc attempts to capture and encode large amounts of common sense knowledge about the real world [6]. However, according to Minsky [17], it is unlikely to expect that the first effort to capture such knowledge will be successful and that competing projects need to be undertaken. In order to provide an agenda for future research, it would be useful to analyze how well the existing efforts are working in order to provide a concrete agenda for future research. Some work on defining an agenda for upper level ontologies has been undertaken by the Upper Level Ontology Summit [19]. However, more detailed analysis is needed. An academic version of Cyc, called ResearchCyc (research.cyc.com) has been created and provides a unique opportunity for such detailed analysis.

The objectives of this paper are to analyze ResearchCyc and make suggestions about future research on how upper level ontologies can be used to support knowledge intensive systems. The contribution of the paper is to provide insights that can be used to further the state of the art of large scale ontologies for knowledge intensive applications. Such insights could be useful in the creation of new, large-scale ontologies to improve their structure and make them more manageable. This research is particularly relevant because of the increased interest and development of very large ontologies such as the GeneOntology[1] and UMLS [2].

2 Domain and Upper Level Ontologies

Research on the semantic web, database, artificial intelligence, and information integration have increasingly focused on the need for ontologies to serve as surrogates for representing knowledge about the real world and how it operates. One approach to capturing such knowledge is through the development of rather small, domain ontologies that capture the concepts of a particular application, the relationships between these concepts, and other related information. These can be organized into libraries of domain ontologies [9, 23]. Probably the most well-known collection of individual domain ontologies is the DAML ontologies (www.daml.com). These were developed specifically for the Semantic Web and stored in a library of approximately 200 ontologies. The need for libraries such as this has long been recognized [18]. In

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

fact, Embley [7] suggests that the development of ontologies is the key to solving the "grand semantics" problem of information systems, which is the most challenging problem today.

For domain ontologies, versatile languages, tools and development environments are available such as Protégé [11], OWL [25], or SPARQL [26]. These tools have experienced different levels of maturity, completeness, and efficiency. There have also been attempts to develop efficient semantic integration mechanisms [10, 15], which are mostly based on mapping techniques.

According to Wikipedia, an upper level ontology "describes very general concepts that are the same across all domains. It is usually a hierarchy of entities and associated rules (both theorems and regulations) that attempt to describe those general entities that do not belong to a specific problem domain." Examples of upper level ontologies include Cyc, Basic Formal Ontology (BFO), DOLCE and DnS, General Formal Ontology (GFO), WordNet (wordnet.princeton.edu), GJXDM/NIEM, Suggested Upper Merged Ontology, and the Biomedical Ontology. Upper level ontologies tend to be formal. They do not come with well-developed "user guidelines" that we find with standard technologies such as cars, iPods, video machines, etc. This makes the integration and the use of ontologies difficult.

The potential usefulness of upper level ontologies is enhanced by the fact that they are domain independent. However, one of the reasons for their lack of adoption is that it is difficult to integrate them. This fact is highlighted by the 2006 Upper Ontology Summit [19], which was convened to provide solutions to problems using upper level ontologies. Doing so requires a thorough analysis and understanding of existing upper level ontologies.

3 ResearchCyc

The Cyc ontology is a knowledge repository developed to capture and represent common sense. It contains more than 2.2 million assertions (facts and rules) describing more than 250,000 terms, including 15,000 predicates. A full version of the Cyc ontology, called ResearchCyc, has been released for the scientific community. ResearchCyc contains both intensional information (entity types, relationship types, integrity constraint) and extensional information (representation of individuals and their relationship to space, time and human perception). Depending on the abstraction level of the knowledge, it may be classified as:

- **Upper Ontology:** This represents very general concepts and relationships between them. For example, it contains assertions such as every event is a temporal thing, every temporal thing is an individual, and every individual is a thing. Thing is ResearchCyc's most general concept.
- **Core Theories:** These represent general facts about space, time, and causality and are essential to almost all common sense reasoning. Examples are geospatial relationships, human interactions and everyday items and events.
- **Domain-Specific Theories:** These are more specific than core theories and deal with special areas of interest such as military movement, the propagation of diseases, finance, and chemistry.

- **Facts:** These represent extensional information also known in ResearchCyc as ground-level facts.

The first three layers describe intensional information (conceptual information) and the last one extensional information (facts). The general knowledge of ResearchCyc covers a broad range and can be classified as:

- Temporal knowledge that describes the temporality of the concepts and their temporal relationships, such as something happens before something else.
- Spatial knowledge that describes spatial properties of concepts such as the superposition of objects, connection, nearness and location, and part of relationships.
- Event, information that describes the most common events that can happen, the actors involved in the events, and their constraints.
- Geography information that facilitates describing the geographic area of the concepts.
- Other general information such as emotion information.

A sample fragment from ResearchCyc related to cars is shown in Figure 1. Cars are represented by the concept *Automobile* in Cyc. This concept participates in 315 constructions, including: subtypes, instances, relationship types, heuristics and constraints. There are more than 300 relationships involving the concept *Automobile.* Not all Cyc concepts are shown in the figure.

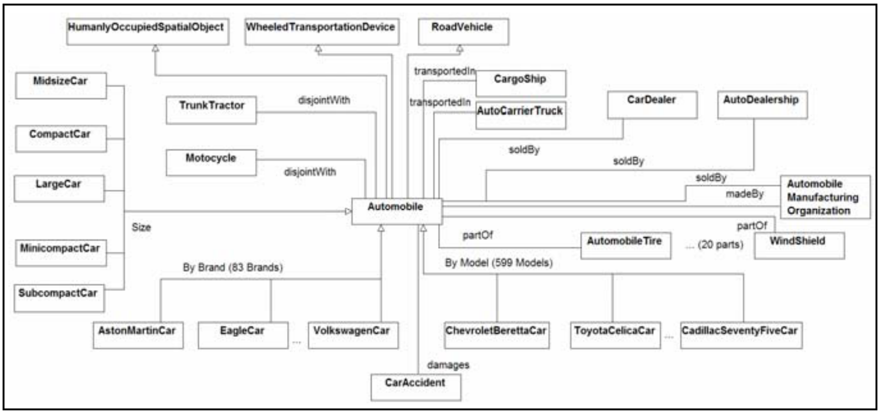


Fig. 1. Car information from ResearchCyc

3.1 Domain Knowledge Represented in ResearchCyc

ResearchCyc represents knowledge through microtheories (Mt). A microtheory represents a domain, and all of its valid assertions. Every assertion must be attached to one or more Mt's.

ResearchCyc has a taxonomy of Mt's in which a microtheory M_1 is the subtype of another microtheory M_2 so that all the facts that are true for supertype M_1 are also true for the subtype M_2. Unfortunately, the Mt's Taxonomy in ResearchCyc is not very usable because:

1. there are many redundant subtype relationships that make it difficult to determine its taxonomical structure;
2. some of the Mt's are almost empty, but are difficult to discard;
3. not all the Mt's follow a standard representation of knowledge.

The parent of all the microtheories is called *BaseKB*. However, determining the hierarchical structure of the Mt's is difficult, manual, and error prone because all the Mt's in ResearchCyc are direct subtypes of *BaseKB* even if they are also its indirect subtypes. For example, Mt *ClothingGMt* describes general information about clothing and is a direct subtype of *BaseKB*. It is also an indirect subtype of *BaseKB* because it is a subtype of *ArtifactGMt*, which is also a subtype of *ArtifactGVocabularyMt*, which, in turn, is a subtype of *BaseKB*.

It is common to define two Mt's for the same domain. In the examples *ArtifactGMt* and *ArtifactGVocabularyMt*, or *ProductGMt* and *ProductGVocabulary*; one of them defines the vocabulary used in the domain (the entity types and relationship types), and the other the rules about the domain (the heuristics, general integrity constraints and derivation rules). However, this distinction is not done for all the microtheories.

Due to the number of microtheories and the high redundancy in their taxonomy, the study of metaclasses of the microtheories is a better approach to summarize the knowledge included in ResearchCyc. On the other hand, the Microtheories taxonomy is useful if one wants to study in more detail whether a domain is well represented.

Table 1 shows the main Metaclasses of Microtheories, the kind of information its instances represent, its instances' purpose and number. We have omitted some metaclasses because they do not have any domain, their microtheories do not cover the domains they represent, or their instances are only used to exemplify the ontology or test some tool.

Some of the discarded microtheories that may be useful for specific domains or future versions of the ontology are: 1) microtheories specific to certain applications (*ApplicationContext*), because such information is too specific and unusable outside of the project or company for which they were conceived, and 2) microtheories representing cultural and belief knowledge because they contain too little information to be usable (*BeliefSystemMicrotheory*).

4 Accessing ResearchCyc

It is difficult to use a knowledge base as large as ResearchCyc. The main problem is discovering whether the information one is looking for is defined in the ontology. Doing so manually is difficult because ResearchCyc has only a textual interface accessed using a browser. It does not provide any facility to query and understand its knowledge. The deficiencies in the linguistic knowledge of ResearchCyc make the searching process even more difficult. Even if we are able to find the knowledge we are looking for, the problem is the large amount of knowledge retrieved. This makes it impossible to automate any process without using heuristics to automatically discard the information that is irrelevant for a particular context or to infer its semantics.

Table 1. Metaclasses of ResearchCyc and Instances

	Name	Purpose	#Inst
Intensional knowledge	BroadMicrotheory	Generic linguistic information, bindings between ResearchCyc concepts and linguistic terms and general vocabulary	7
	GeneralMicrotheory	General information of domains, excluding microtheories dealing with specific situations	179
	VocabularyMicrotheory	Specify the vocabulary related to some topic or domain	127
Factual Knowledge	DataMicrotheory	Information about individuals of a certain kind, such as specific persons, organizations or geographical places	259
	ProposionalInformation Think	Represents documental information such as demographic information, sensory information, news, or published sources	36
	ApplicationContext	Specific to certain application	183
Linguistic Knowledge	Language-Specific	Information about natural languages (only English languages)	65
	LexicalMicrotheory	Lexical information about English languages	56

The classification in Table 1 can be used to check if a given domain is defined in ResearchCyc. For example, if we are looking for biological knowledge we know it should be an instance of *generalMicrotheory* or *VocabularyMicrotheory*. Hence, we can search their instances to find a microtheory that deals with biological knowledge (*BiologyMt*). After locating the microtheory that contains the relevant knowledge, we would like to know to what extent the domain is represented in ResearchCyc. To determine this, we must take into account the microtheory (*BiologyMt*) and its super microtheories (*BiologyVocabularyMt* and *BiochemistryMt*). The definition of a local taxonomy of the selected microtheory is useful.

Regarding the inferences in ResearchCyc, the queries are executed using a microtheory as a context. Therefore, it is quite important to identify the correct microtheory for each query. Executing a query using a wrong Mt means that a query that may be answered using the ResearchCyc knowledge will have no results. For example, the query "in which city the liberty bell is located" (represented as *(#$objectFoundInLocation #$LibertyBell ?CITY)*) has no answers under the *BaseKB* microtheory. However, if we carry out the same query using the *CurrentWorldDataCollectorMt* Mt, it returns Philadelphia as a result. Unfortunately, there is no Mt that fits all the queries because the correct microtheory depends on the context of the query. For example, a query that deals with today's facts may need the *CurrentWorldDataCollectorMt*, and a query that deals with linguistic information may need *EnglishMt* microtheory. This problem worsens in web queries because the context of a web query cannot be obtained automatically from its terms. Therefore, even when ResearchCyc has relevant knowledge for a query, we may be unable to retrieve it because we cannot figure out on which microtheory we should focus.

The amount of knowledge included in ResearchCyc should be simplified when we want to use it automatically. The ResearchCyc ontology contains 82 relationship types that signify an object that is part of another object (*partOf*), such as *subOrganization* or *capitalCity* relationship types. Our program does not need to know the exact semantics of each of them, but can deal with them as simple *partOf* relationships. This simplification helps the program identify when a part-of relationship exists and its part and whole participants.

4.1 Using ResearchCyc for Improving Web Queries

ResearchCyc has different kinds of knowledge: concepts and generalization relationships that represent the intensional information about a domain and individual and classification relationships that represent extensional information. Furthermore, a concept of the ontology may represent either semantic or linguistic information. For example, ResearchCyc states that there is a concept called *DomesticPet*, which is a noun and is the denotation of the word *Pet* in such a language.

ResarchCyc also contains heuristics and integrity constrains. Although, in general, heuristics and integrity constraints are not required in an ontology, they enable inference and may be useful for detecting concepts that are related to the context that the user is interested. For example, the heuristic which indicates that most pets have a pleasant personality may be used to infer that a user who is searching for a friendly animal is interested in pets. This inference is possible because there is a generalization relationship between *Pet* and *Animal*, a synonym relationship between friendly and pleasant, and a heuristic which indicates that Pets tend to be friendly.

One application of ResearchCyc is web query expansion [5]. Suppose a user wants to learn about the flutes (glasses) made with bohemian glass and he/she submits the query “flute Bohemian Drink” to Google. It will return 57,900 hits with only 5 relevant results from the first 20.

Suppose ResearchCyc is used to improve the query before its final submission to a search engine. If we search ResearchCyc for the term Flute, we obtain two results: the concept *ChampagneFlute* (that represents the glass) and the concept *Flute* (that represents the musical instrument). The concept *ChampagneFlute* has as a supertype, the concept *DrinkingGlass*. Taking into account that the query also contains the word drink, and that the concept *DrinkingGlass* is a noun for “Drink”, we can deduce that the user is not interested in the instrument. Furthermore, searching for the word Bohemian returns two regions of the Czech Republic. These two concepts are sub regions of the Czech Republic, so one could deduce that the user is interested in the Flute glasses made in such a country. Note that the word drink has been necessary to disambiguate the flute concept, but is not necessary anymore. Hence, the query can be refined as “Champagne + Flute + glass + Czech”. Submission of this query to Google results in 153,000 hits. 16 of the first 20 results are relevant to the user’s query.

5 Lessons Learned and Suggestions

5.1 Overall Problems

We have identified and classified various problems working with ResearchCyc. This classification is based in the ISO/IEC 9126 quality model standard [12], which defines a Software Quality Model applicable to every kind of software. The reason for choosing such a standard is because the problems we found are not only related to the knowledge base of ResearchCyc, but also to its implementation, that is, the tools and browser that Cyc provides to manage it. Since these are part of the functionality, the current metrics of ontologies do not fit our purpose.

Our classification (Figure 2) differentiates four kinds of problems:

- Functionality problems: problems with the tools that support the use of ResearchCyc and their functions.
- Usability problems: problems that deal with the understandability, the difficulty in learning the ontology knowledge, and difficulty in managing it.
- Portability problems: technical problems that occur when we try to execute or install ResearchCyc in some environments.
- Reliability problems: problems with the maturity of the ontology knowledge.

The problems may also be organized using other schemes. For example, problems may be classified based on whether they are related to: a) the content of the ontology (C), b) the support tools for the ontology (T), c) the adopted language (L), d) the documentation (D), and e) other factors (O). Even though some of the problems we found with ResearchCyc are related to ontology content (maturity and understandability problems), most of them are not. In order to better clarify these problems we have annotated them with the afore mentioned letters (C, T, L, D or O).

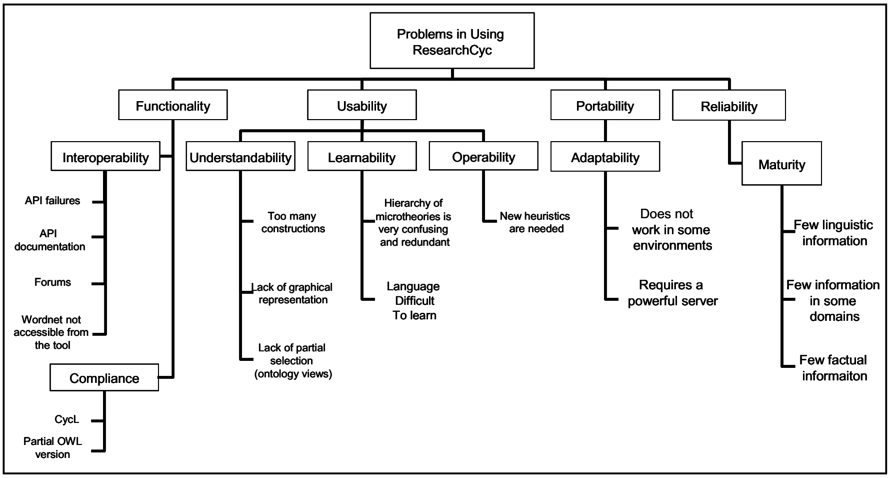

Problems in Using ResearchCyc
Functionality
Usability
Portability
Reliability
Interoperability
Understandability
Learnability
Operability
Adaptability
Maturity
API failures
API documentation
Forums
Wordnet not accessible from the tool
Compliance
CycL
Partial OWL version
Too many constructions
Lack of graphical representation
Lack of partial selection (ontology views)
Hierarchy of microtheories is very confusing and redundant
Language Difficult To learn
New heuristics are needed
Does not work in some environments
Requires a powerful server
Few linguistic information
Few information in some domains
Few factual informaiton

Fig. 2. Classification of problems with ResearchCyc

The following problems were found with ResearchCyc, classified by their type:

- Functionality problems:
  - *Interoperability problems* that occur due to the necessity to interact with other systems. In our case it contains the problems we found with the ResearchCyc API and the ontology browser provided with the ontology.

1. *API failures* (T): some API calls do not work properly. For example, we experimented that the calls that return all the instances of a given concept (*getAllInstances*) does not work when the concept has many instances.
2. *The lack of useful documentation of the ResearchCyc API* (D). There is very little documentation about the different classes within the API and how to use them. Sometimes, understanding the source code of the API is the only way to identify whether the needed functions are available, to discover the possible values of an enumerated type in the API, or to identify how to solve an undocumented error. For example, although there is a function in the API that returns the concepts that represent the meanings of a term (getDenotationsOf), it is not defined in the API documentation. Therefore, one has to inspect the source code of the API where it is defined (the class CycAccess) to discover its existence.
3. *The lack of attention people give to the forums of the ResearchCyc API* (D). These forums are rarely updated, and the doubts and errors the users find are usually not solved nor answered. For example, a message posted from the authors in January 2006, is yet to be answered.
4. *WordNet is not accessible from the Cyc Browser* (T): Cyc contains references to WordNet to state the synonymy information about concepts. However, the ontology browser provided with Cyc does not exploit that information. Therefore, if we want to know the synonym of a concept, we have to take the WordNet reference from the Cyc browser and then search WordNet to find such a reference.

- *Compliance problems*: These problems arise because standards, conventions or regulations have not been applied.
  1. *CycL language* (L): The language used to write the ontology is not a de facto standard. It is difficult to use because it requires effort to learn the language before dealing with it.
  2. *Partial OWL version* (L): there is an OWL version of the Cyc ontology, but it contains only a small subset of the Cyc ontology.

- Usability problems: they may be classified as:
  - *Understandability problems*: they increase the users' effort to understand the ontology knowledge.
    1. *Too many constructions and non-standard names* (L): CycL, which is the language in which Cyc is represented, contains a large number of different constructions. Some of these constructions do not use standard names. For example, power types are represented by using a constructor called *façade.* The *instanceOf* relationship type is called *IsA* in Cyc, which may cause interpretation errors because the name *IsA* is used with another meaning in conceptual modeling (to represent generalizations instead of instantiations).
    2. *Lack of graphical representation* (T): Due to the lack of graphical representation, it is very difficult to figure out the context and related knowledge for a concept.
    3. *Lack of techniques that show ontology views* (T): since the ontology is very large, we need some mechanism to retrieve only the part of the ontology

that is relevant to the user. Since that functionality does not exist, the user has to navigate up the taxonomy in order to see, for example, whether a car model has a name.

- *Learnability problems*: the process of learning and discovering Cyc knowledge is difficult.
  1. *CycL is difficult to learn* (L): since CycL is difficult to learn and there is no graphical representation of the ontology, it is quite difficult to figure out the knowledge contained in the ontology.
  2. *Very confusing microtheories hierarchy* (C): as mentioned in section 3.1, the taxonomy of microtheories is not very usable. This makes it especially difficult to know whether a given domain is represented in the ontology.
- *Operability problems* (O): problems that appear when users are trying to work with the ontology, i.e., using the ontology knowledge.
  1. The huge quantity of knowledge incorporated in ResearchCyc makes it necessary to define new heuristics to identify which kind of knowledge will be useful and which kind to be discarded.

– Portability problems (O): we found portability problems, which are problems we encountered when using ResearchCyc in different environments:
  1. We ran into problems running ResearchCyc on other operating systems. Specifically, we have been unable to run the API under several Windows XP systems and DEVIAN releases of the Linux operating system.
  2. Execution of ResearchCyc requires a powerful server to run efficiently.

– Reliability problems (C): the only reliability problems using ResearchCyc has been related to its maturity. These problems are:
  1. There is very little linguistic information represented in ResearchCyc compared with other lexical ontologies such as WordNet. In particular, in ResearchCyc there are no defined denotations of all the concepts, and only few synonyms and antonyms are represented.
  2. There are some domains not very deeply specified in the ontology.
  3. Even though the intensional information of ResearchCyc is huge, its factual information is very limited.

5.2 Suggestions for Improving Upper Level Ontologies

To develop better upper level ontologies, such as ResearchCyc, we need improvements in the following areas:

1. Better documentation is needed.
2. Tools for browsing and navigating the content must be developed.
3. Tools for searching and summarizing the concepts stored in the ontology are needed. There are some techniques that summarize conceptual schemas that may be applied to ontologies, such as the abstraction class diagrams [27]. Using that technique we can create an ontology that allows the user to see the most relevant terms the ontology deals with and the main relationships between them. Other techniques such as ontology pruning [24, 4] or segmentation [13] may be used to select the part of the ontology in which the user is interested.

4. Graphical visualization of content is needed for usability reasons.
5. Representation is usually geared towards human consumption, but should be focused on the consumption by applications and software agents.
6. Better classification of concepts in different domains (or contexts) is necessary. A non-redundant domain taxonomy that facilitates searching and helps the user determine the extent of knowledge represented in a domain would be useful.

Incorporating these suggestions for creating ontologies and building appropriate tools should significantly reduce the problems associated with large scale ontologies.

6 Conclusion

Upper level or large scale ontologies are intended to support a wide range of applications that require knowledge about the real world. This research has distinguished these large scale ontologies from domain ontologies and analyzed ResearchCyc, a version of the most well known upper level ontology. The use of ResearchCyc for web query improvement and other applications has the potential to be very successful. However, the quantity of information it contains and the lack of mechanisms that organize this information in a usable way, makes ResearchCyc difficult to deal with. This may be why only few researchers are currently using it. Furthermore, ResearchCyc has very little linguistic information compared with other lexical ontologies such as WordNet. In particular, in ResearchCyc the denotations of all the concepts are not defined, with only few synonyms and antonyms represented.

Although problematic, it would be worthwhile to use ResearchCyc to support web queries. The reason is that, as far as we know, it is the only ontology that contains linguistic, semantic and factual knowledge. Also, its knowledge covers more domains than other current ontologies. Furthermore, there are some efforts to improve the Cyc knowledge with linguistic information from WordNet and factual information from the World Wide Web [16, 20]. These efforts would reduce its maturity problems.

It is obvious that small ontologies are more usable than large ontologies such as Cyc. The question is whether the knowledge contained in large ontologies is better compared to the knowledge contained in domain ontologies. To answer this question we plan to carry out, as part of our future work, experimentation in which knowledge of different domains are compared with the knowledge represented by large ontologies. To do so, we will use tools that provide support for ontology creation, such as Protegé, to import domain ontologies. Then, we will compare these ontologies with each other, and against other large ontologies such as ResearchCyc. Our future work will also include investigating whether the problems found with ResearchCyc are generalizable to other upper level ontologies.

Acknowledgement. This work has been partly supported by the Ministerio de Educacion y Ciencia under project TIN 2005-06053, Georgia State University, and Oakland University.

References

1. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web, Scientific American, pp. 1–19 (May 2001)
2. Bodenreider, O.: The Unified Medical Language System (UMLS): integrating biomedical terminology. Nucleic Acids Res, 1(32) (2004)
3. Bunge, M.: Treatise on basic philosophy: vol. 3: Ontology 1: The furniture of the world. Reidel, Boston (1977)
4. Conesa, J., Olivé, A.: A Method for Pruning Ontologies in the Development of Conceptual Schemas of Information Systems. Journal of Data Semantics (2006)
5. Conesa, J., Storey, V.C., Sugumaran, V.: Using Semantic Knowledge to Improve Web Query Processing. In: Kop, C., Fliedl, G., Mayr, H.C., Métais, E. (eds.) NLDB 2006. LNCS, vol. 3999, Springer, Heidelberg (2006)
6. Cyc: Cyc Ontology from http://www.cyc.com
7. Embley, D.W.: Toward Semantic Understanding: An Approach Based on Information Extraction Ontologies. In: 15th Conf. on Australian Databases, New Zealand (2004)
8. Embley, D.W., Liddle, S.: TANGO Project (2005), http://tango.byu.edu (NSF Grant IIS-0083127)
9. Gruber, T.R.: Every ontology is a treaty - a social agreement - among people with some common motive in sharing (Tomas R. Gruber's Interview). SiG SEMIS - Semantic Web. and Information Systems 1(3), 4–8 (2004)
10. IBM: alphaWorks: IBM Integrated Ontology Development Toolkit from http:// www.alphaworks.ibm.com/tech/semanticstk
11. Informatics, S.M.: The Protege Ontology Editor and Knowledge Acquisition System. (2007), Retrieved January 22, 2007 from http://protege.stanford.edu/
12. ISO/IEC International Standard 9126-1: Software Engineering-Product Quality-Part 1: Quality Model (2001)
13. Seidenberg, J., Rector, A.L.: Web ontology segmentation: analysis, classification and use. WWW 2006, pp. 13–22 (2006)
14. Lenat, D.B.: CYC: a large-scale investment in knowledge infrastructure. Communications of the ACM 11(38), 33–38 (1995)
15. Li, M., Wang, D., et al.: Ontology Construction for Semantic Web: A Role-Based Collaborative Development Method. In: Zhang, Y., Tanaka, K., Yu, J.X., Wang, S., Li, M. (eds.) APWeb 2005. LNCS, vol. 3399, pp. 609–619. Springer, Heidelberg (2005)
16. Matuszek, C., et al.: Searching for common sense: populating Cyc from the web. In: Proceedings of the Twentieth National Conference on Artificial Intelligence (2005)
17. Minsky, M.: A Conversation with Marvin Minsky About Agents. Communications of the ACM 37(7), 22–29 (1994)
18. Noy, N.F.: Semantic Integration: A Survey of Ontology-Based Approaches. SIGMOD 33(4), 65–70 (2004)
19. Obrst, L., Cassidy, P., Ray, S., Smith, B., Soergel, D., West, M., Yim, P.: The 2006 Upper Ontology Summit Joint Communique (2006)
20. O'Hara, T., et al.: Inducing criteria for mass noun lexical mappings using the Cyc KB, and its extension to WordNet, Workshop on Computational Semantics: 2003 Tilburg (2003)
21. Ram, S., Park, J.: Semantic Conflict Resolution Ontology (SCROL): An Ontology for Detecting and Resolving Data and Schema-Level Semantic Conflicts. IEEE Trans on Knowledge and Data. Engineering 16(2), 189–202 (2004)

22. Sugumaran, V., Storey, V.: The Role of Domain Ontologies in Database Design: An Ontology Management and Conceptual Modeling Environment. ACM Transactions on Database Systems 31(3), 1064–1094 (2006)
23. Swartout, W.R., Patil, R., et al.: Toward Distributed use of Large-Scale Ontologies. In: Proc. Tenth Knowledge Acquisition for Knowledge-Based Systems Workshop, Canada (1996)
24. Volz, R., Studer, R., et al.: Pruning-based Identification of Domain Ontologies. Journal of Universal Computer Science 9(6), 520–529 (2003)
25. W3C: OWL Web Ontology Language-Overview. W3C Recommendation, from http:// www.w3.org/TR/owl-features/
26. W3C: SPARQL Query Language for RDF. http://www.w3.org/TR/rdf-sparql-query/
27. Egyed, A.: Automated Abstraction of Class Diagrams. ACM Transactions on Software Engineering and Methodology 11(4), 449–491 (2002)

From OWL Class and Property Labels to Human Understandable Natural Language

Günther Fliedl, Christian Kop, and Jürgen Vöhringer

Institute of Applied Informatics,
Alpen-Adria-Universität Klagenfurt

Abstract. The ontology language OWL has become increasingly important during the previous years. However due to the uncontrolled growth, OWL ontologies in many cases are very heterogeneous with respect to the class and property labels that often lack a common and systematic view. For this reason we linguistically analyzed OWL class and property labels focusing on their implicit structure. Based on the results of this analysis we generated a first proposal for linguistically determined label generation which can be seen as a prerequisite for mapping OWL concepts to natural language patterns.

1 Introduction

The importance of ontologies has grown significantly during the previous years. OWL (Web Ontology Language) [11] is a W3C recommendation for the semantic web and now very commonly used for representing knowledge provided by domain experts in enterprises as well as research groups. OWL ontologies are based on RDF Schemata [15] and provide a specific XML representation [16] of classes and hierarchies from a specific domain. Adapting XML and RDF to the OWL format offers an extended structure and formal semantics in order to store expert knowledge of a certain domain by describing classes, properties, relations, cardinalities etc. One big advantage of using a „formal" ontology like OWL for describing your domain vocabulary is that they can be automatically machine-interpreted which makes further knowledge processing easier.

Because of the uncontrolled growth, OWL ontologies have become very heterogeneous and therefore hard to integrate from a generic viewpoint. While numerous OWL-example ontologies are available for free, their class and property labels lack a systematic structure. This has a major drawback during support of machine-readability. In particular OWL ontologies are very hard to understand for human readers. Thus specific OWL ontologies are commonly criticized for being difficult to reuse, to transform to other domains or languages and to integrate with other ontologies. For this reason we (the NIBA[1] workgroup) decided to linguistically

[1] NIBA (German acronym for Natural Language Information Requirements Analysis) is a long term research project sponsored by the Klaus-Tschira-Foundation in Heidelberg, Germany dealing with the extraction of conceptual models from natural language requirements texts. [3], [4], [10].

Z. Kedad et al.(Eds.): NLDB 2007, LNCS 4592, pp. 156–167, 2007.

analyze OWL class and property labels and systematize their labeling strategies. Therefore we investigated some of the underlying basic default patterns according to their usefulness for the development of labeling style guides. Subsequently we began to build some example grammars for the generation of NL (natural language) sentences and sentence lists from specific ontology concepts. This step was necessary for improving the explicit readability within and beyond OWL classes, which in turn enhanced the usability. In this paper we focus on systematizing some aspects of property-labels, since we believe that they can be seen as one of the core problems of ontology interpretation.

The paper is structured in the following way: in chapter 2 we give a short overview of the OWL concepts that are relevant for our argumentation. We also describe some problems concerning the diversity of class and property labeling methods used by the OWL community. In chapter 3 we briefly address related work that focuses on the verbalization of ontologies, e.g. Attempto and Swoop. Chapter 4 describes the NIBA approach for OWL verbalization, i.e. the filtering of linguistic patterns, the development of labeling style guides and the creation of Prolog-interpretable DCG (Definite Clause Grammar) rules for the creation of NL sentences encoding OWL concepts. Our paper concludes in chapter 5 with an outlook on future work.

2 OWL Concepts Relevant for Labeling

Because OWL is a W3C recommendation for the semantic web it has gained major importance in the previous years. OWL is application-oriented, e.g. it was developed mainly for the automatic processing of domain knowledge instead of preparing content for humans. As mentioned above, OWL can be seen as an extension of RDF Schemata using classes, properties and instances for application environments in the WWW. The OWL extension of RDF allows the specification of restrictions like properties or cardinality constraints.

Since the specification of property labels is frequently based on class or subclass label names we shortly discuss the class labeling problem. Subsequently we go into the specification of OWL property labels and the related problems.

2.1 The OWL Way of Defining Classes and Individuals

Many default concepts in a given domain should correspond to classes, functioning as roots. Every OWL individual is automatically a member of the class owl:Thing and every user-defined class is implicitly a subclass of owl:Thing. Domain specific root classes are defined by simply declaring a named class. In the well-known wine-example-ontology[2] the root classes are *Winery*, *Region* and *ConsumableThing*, which are defined in the following way:

[2] This example were taken respectively from *http://www.w3.org/2001/sw/WebOnt/guide-src/wine.owl* and *http://www.w3.org/2001/sw/WebOnt/guide-src/food.owl*

```
<owl:Class rdf:ID="Winery"/>
<owl:Class rdf:ID="Region"/>
<owl:Class rdf:ID="ConsumableThing"/>
```

As these class labels show, class-names in the wine ontology are rather simple and straight-forward. The only potentially problematic label-name in these examples is *ConsumableThing*, since it consists of two words having been merged using the upper case as a delimiter strategy. However since no guidelines are available for creating these labels, other ontologies contain labels constructed with very different naming and delimiter strategies, which leads to an uncontrolled growth of labeling patterns, as can be seen in table 1 [17]:

Table 1. Typical class labels

ActionType
Activate
Activvated-carbon-equipment
Activated_p21cdc42Hs_Kinase
Acute_Myeloid_Leukemia_in_Remission
ADM-DIV-BARBADOS-PARISH
AdministartiveStaffPerson
Glycogen-Rich-Carcinoma

The members of classes are called individuals. In the wine ontology specific wine grapes like *CabernetSauvignonGrape* would be an individual of the class *WineGrape*. This relationship is defined in the following way:

```
<WineGrape rdf:ID="CabernetSauvignonGrape" />
```

Likewise *CentralCoastRegion* is a specific member of the class *Region* and therefore an individual, which can be defined as follows:

```
      <Region rdf:ID="CentralCoastRegion" />
      <owl:Thing rdf:ID="CentralCoastRegion" />
```

Another way of defining this information is the following one:

```
      <owl:Thing rdf:about="#CentralCoastRegion">
         <rdf:type rdf:resource="#Region"/>
      </owl:Thing>
```

Obviously the same labeling strategies for individuals are available as for classes. Looking at the examples on the OWL Standard web page, arbitrarily merged multi-terms with upper case delimiters are preferred. The internal semantics of the terms and the sub-terms is not discussed any further.

2.2 The OWL Way of Defining Relational Object Properties

In OWL classes and individuals are extended via property definitions (see OWL definition [11]). In OWL a property definition consists of the definitions of a domain and a range, which can be seen as restrictions. The concepts ObjectProperty, *rdfs:domain*, *rdfs:rang*e are used for defining properties of objects which can be described as relations between classes. For the definition of object properties we again make use of the wine ontology:

```
<owl:ObjectProperty rdf:ID="madeFromGrape">
  <rdfs:domain rdf:resource="#Wine"/>
  <rdfs:range rdf:resource="#WineGrape"/>
</owl:ObjectProperty>
```

The three lines above are related to each other with an implicit conjunction operator: "*The object property madeFromGrape has domain Wine* ***and*** *a range WineGrape*".

With the property definition, it is now possible to expand the definition of Wine to include the notion that a wine is made from at least one grape (*WineGrape*). As in property definitions, class definitions include multiple subparts that are implicitly conjoined.

```
<owl:Class rdf:ID="Wine">
      <rdfs:subClassOf rdf:resource="&food;PotableLiquid"/>
         <rdfs:subClassOf>
           <owl:Restriction>
             <owl:onProperty rdf:resource="#madeFromGrape"/>
             <owl:minCardinality
    rdf:datatype="&xsd;nonNegativeInteger">1</owl:minCardin
    ality>
           </owl:Restriction>
         </rdfs:subClassOf>
         ...
       </owl:Class>
```

Table 2 contains parts of a list of property labels that are provided by the community [17].

Table 2. Typical property labels

BaseOnBalls
basePublicationURI
behind_Generality
concessive-RST
hasProduct
hasDiameter_of_size

As you can see the main problem of the listed property labels is that the internal structure of the various entries doesn't allow any kind of systematic interpretation concerning the relation between the sub-terms and their function.

3 A Linguistic Way of Solving the OWL Labeling Problem

As can be seen in the previous chapter, no consistent labeling strategies exist for OWL class and object property labels. This makes the manual and automated interpretation and further processing of these labels more difficult. Our aim is to provide style guides for label generation in order to define a framework for systematic ontology engineering. After proposing linguistically motivated style guides which could help to optimize the ontology creation process, we identified relevant linguistic patterns. This in turn facilitates further processing steps.

3.1 Labeling Style Guides

We discovered that utilizable OWL labels are mainly created following style guides of programming languages. Computer Science programming, languages or models have unambiguous syntax, additional explicit style guidelines for using them are common. Adhering to these guidelines leads to models and programs that can be much easier interpreted. As an example of such guidelines see for instance [1], [8], [9], [13] which use the Pascal and Camel Notation for Classes and Methods. We claim that the definition and use of style guidelines should be extended to ontology engineering. Our special concern is the linguistically motivated setting of labels which is not restricted at all in OWL.

Table 3 lists guidelines for defining OWL class and individual labels, object property labels and general labeling guidelines. We also give examples for each guideline. The guidelines are based on the idea that general linguistic concepts should be used extensively when creating class and object property labels. With these guidelines machine and human interpretability are combined in a meaningful way. We assume that the use of linguistic categories (Verb, Noun, Participle etc.) is very common and therefore already known.

Table 3. OWL labeling guidelines

No	Guideline	Example	(C)lass/ (I)ndividual/ (P)roperty
1	All labels should be written in English.	producesWine	C,I,P
2	If a label consists of more than one term, a definite delimiter between the term must be used. Here we follow the guideline that an upper case character works as delimiter (Pascal Notation or CamelNotation).	VintageYear, hasIntrinsicPattern	C,I,P
3	Abbreviations must be expanded (e.g. instead of No. → Number, instead of org → Organization)	calculateNumber, Organization	C,I,P
4	Acronyms in a label should be written like normal nouns starting with an upper case letter.	FpsSeason, hasHtmlSource, statusGui	C,I,P

Table 3. (*continued*)

5	Class labels must be specified by either • atomic nouns, • compound nouns, • adjective + noun (component noun) • or URLs	Grape, WineGrape, http:// zorlando.drc.com/ daml/ontology/ Glossary/current/ intensionalDefinition	C
6	Individual labels must be specified either by • URLs, • Acronyms, • proper names • or compound proper names	Merlot, CabernetSauvignonG rape, Html	I
7	If Class and Individual labels are described by atomic nouns / proper names then they must start with upper case.	Pizza, Loire	C,I
8	If Class and Individual labels are described by compound nouns / proper names or adjective + noun (compound noun) then they must start with upper case.	PizzaTopping, WhiteLoire	C,I
9	Singular forms should be used on nouns in Class and Individual labels	Winery, Color, Region, PizzaTopping	C,I
10	Property labels must be written in mixed case starting with lower case starting with either a Verb in 3rd person singular; Participle verb; “Has”; “is”	hasColor, madeFromGrape	P
11	Labels starting with has, can have the form has [+ Adjective] + Noun (=Range)	hasBrightColor	P
12	Labels starting with is, must have the form Is [+ Adjective | Participle] [+ Noun (=Domain)] [+ Preposition] + Noun (=Range)	isLocatedInRegion, isBrotherOfPerson	P
13	Labels starting with a verb in 3rd person singular must have the form Verb [+ Preposition] + Noun (=Range) [+Preposition]	ownsCar, producesWine, sendsTo, receivesFrom, sendsToRecipient, sendsLetterTo	P

3.2 Filtering Linguistic Patterns

Presupposing the application of the above listed labeling style guides, label names can be evaluated by mapping them to general linguistic patterns, consisting of common linguistic category symbols. Since property labels have a more interesting and complex linguistic structure than class labels, we explicitly deal with object property labels in this chapter. Table 4 shows some typical examples for linguistic patterns which satisfy the guidelines.

Table 4. Linguistic patterns of OWL property labels

No	Example	Linguistic patterns

1	<owl:ObjectProperty rdf:ID="**isLocatedInRegion**">
	<rdf:type rdf:resource="&owl;TransitiveProperty" />
	  <rdfs:domain rdf:resource="&owl;Thing" />
	  <rdfs:range rdf:resource="#Region" />
		Is +(verbal) Participle + Preposition + Noun (=Range)

2	<owl:ObjectPropertyrdf:ID="**isMadeFromWineGrape**">
	  <rdfs:domain rdf:resource="#Wine"/>
	  <rdfs:range rdf:resource="#WineGrape"/>
		Is + (verbal) Participle + Preposition + Noun (=Range)

3	<owl:ObjectProperty rdf:ID="**hasAdjacentRegion**">
	  <rdf:type rdf:resource="&owl;SymmetricProperty" />
	  <rdfs:domain rdf:resource="#Region" />
	  <rdfs:range rdf:resource="#Region" />
	</owl:ObjectProperty>
		Has + Adjective + Noun

4	<owl:FunctionalProperty rdf:ID=„**ownsPassportNumber**"> <rdfs:domain rdf:resource="#Person"/> <rdfs:range rdf:resource="#PassportNo"/> </owl:FunctionalProperty>	Verb [3rd person singular] + Compound Noun

5	<owl:ObjectProperty rdf:ID="**hasWineDescriptor**">
	  <rdfs:domain rdf:resource="#Wine" />
	  <rdfs:range rdf:resource="#WineDescriptor" />
	</owl:ObjectProperty>
		has + Compound Noun

4 Approaches to OWL Verbalization

Since "formal" ontology representations of knowledge lack easy traceability for humans, certain methods for the verbalization of ontologies like OWL have been developed. In the following we very briefly describe two approaches to OWL verbalization and their shortcomings. The chapter concludes with a draft of our own approach to verbalization.

4.1 State of the Art in OWL Verbalization

There are many software systems which graphically present ontologies (e.g. [2], [6], [7], [12], [14]). To our knowledge two approaches for verbalizing OWL ontologies currently exist: Attempto Controlled English (ACE) [5] and Swoop [6].

ACE is a subset of the English language. It uses reduced (controlled) patterns for "verbalizing" OWL ontologies. These translations are more easily interpretable by human readers and can always be translated back to the ontology representation. The Attempto approach uses its own grammar rules for constructing simple sentences[3] which don't allow ambiguities, sentence gaps and any sorts of fuzziness, as the verbalization example below shows:

An arbitrary property definition in the OWL-wine-ontology

```
<owl:ObjectProperty rdf:ID="madeFromGrape">
<rdfs:domain rdf:resource="#Wine"/>
  <rdfs:range rdf:resource="#WineGrape"/>
```

and its usage with min cardinalities between the domain "Wine" and the range "WineGrape"

```
<rdfs:subClassOf>
- <owl:Restriction>
    <owl:onProperty rdf:resource="#madeFromGrape" />
    <owl:minCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeIntege
r">1</owl:minCardinality>
      </owl:Restriction>
      </rdfs:subClassOf>
```

results in the following ACE translation:

Every Wine madeFromGrapes at least 1 things.

Since ACE does not split up the class and property labels the translation is suboptimal for our purpose.

Swoop on the other hand is an ontology engineering toolkit, which implements an algorithm by using some standard NL techniques for translating OWL ontologies to NL patterns. This can be seen as an extension to ACE for the property labeling problem. Swoop uses general linguistic category symbols like V, NP, VP etc. for a shallow analysis of OWL labels and it proposes a fixed set of expansion rules for the linguistic patterns. See for example [6]:

(has) NP
– Examples: email, hasColor
– Expansions: X has a color Y
– Alternate (if Y is an AdjP): X has Y color

The heuristics which are proposed for resolving them appear to be quite simple currently and not sufficient enough for general application. The SWOOP engine uses a Part-Of-Speech Tagger for automatically detecting linguistic categories of words and generating corresponding NL sentences. This strategy does not allow a sufficient solution of the ambiguity problem. Therefore the authors of SWOOP propose simple disambiguation strategies like giving priorities to verbal forms, which presuppose currently non-existing ontology guidelines.

[3] see the Attempto OWL verbalizer web interface at: *http://attempto.ifi.unizh.ch/site/docs/verbalizing_owl_in_controlled_english.html*

Hence both of the evaluated verbalization approaches have weaknesses which we respond to with our own approach. In the following section we give a brief outline of our approach including some first results.

4.2 Generation of Natural Language Patterns

We propose a step-by-step approach for a linguistically based and elaborated ontology verbalization. The approach presupposes the labeling style guidelines and the linguistic patterns of OWL labels defined in chapter 3. The generation of natural language patterns for OWL classes and properties is based on the NTMS[4]-Paradigma and DCG rules, which have been developed during the early stages of the NIBA-Project.

The proposed grammar uses NTMS-category labels like v3(=sentence node), n3 (= nominal phrase), a0 (= adjective), n0 (= noun), aux0(= auxiliary), v0(= verb), <pass> (= passivation) and <tvag2> (= transitive, agentive verb> and pp (= past participle). It produces binary trees containing categorical and lexical nodes, which are identical with natural language words. Relationships between class labels and property labels are transformed to simple sentences.

The following extract of the OWL-wine-ontology has been transformed to linguistic objects using the DCG rules underneath, presupposing the fact that labels inside the XML representation can easily be cut out according to our labeling guidelines.

The concept fragments underneath

```
 <owl:Class rdf:ID="Sauterne">
       ........
<rdfs:subClassOf>
    <owl:Restriction>
    <owl:onProperty rdf:resource="#locatedIn" />
    <owl:hasValue rdf:resource="#SauterneRegion" />


<owl:Class rdf:about="#WhiteLoire">
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty rdf:resource="#madeFromGrape" />
```

are transformed to a set of parser rules. These parser rules produce sentences and syntactically relevant phrase nodes which can be enriched with attributes like class, loc (= location), locV (= locative verb), pass, tvag2 etc. The two output examples are both represented in bracketing and graphical tree format, allowing a better visualization of the encoded grammatical structure:

```
v3(v2(n3_class(spz0(['The']), n2(n0(['Sauterne']))),
v2(v1(v0(aux0_loc([is]), v0_locV([located])), p2(p0([in]),
n3(spz0([the]), n0_loc(['Region']),
n0_value(['Sauterne'])))))))
```

[4] Acronym for Natural Theoretic Morpho-Syntax [3].

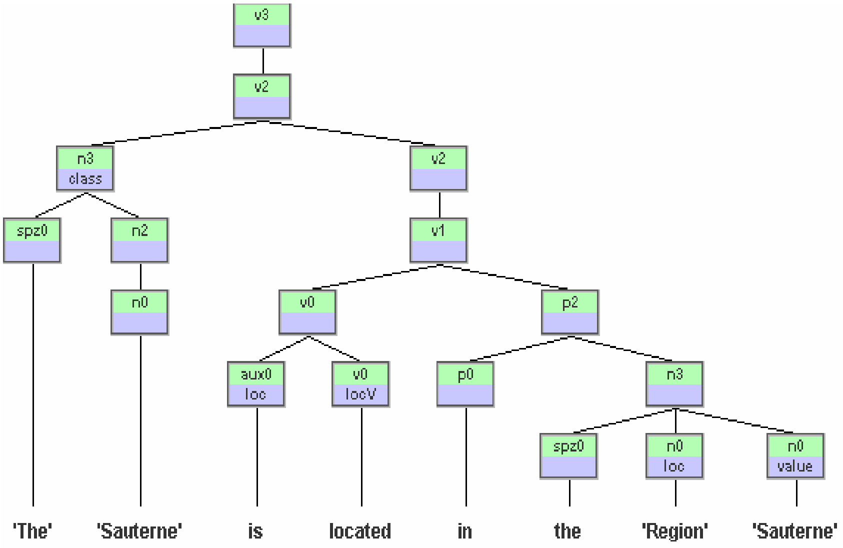


Fig. 1. First parse tree

```
v3(v2(n3_class(spz0(['The']), n2(a2(a0([white])),
n0(['Loire']))), v2(v1(v0(aux0_pass([is]), v0_tvag2([made])),
p2(p0([from]), n3(n0_source(['Grape'])))))))
```

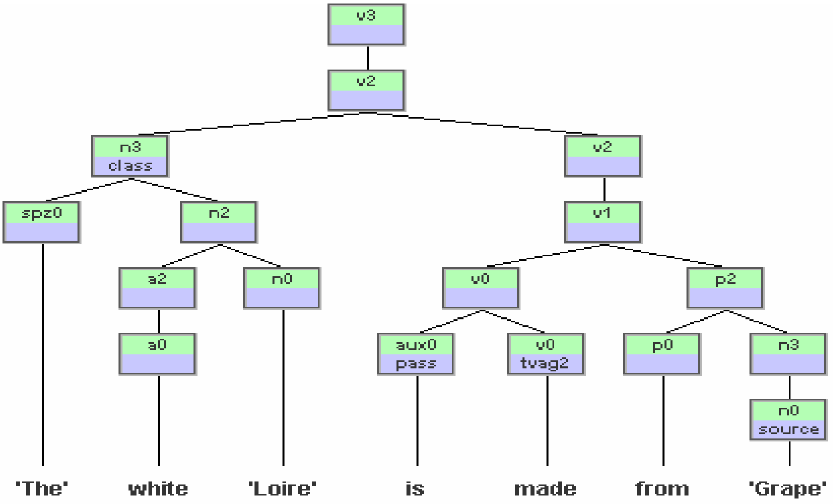


Fig. 2. Second parse tree

The OWL concept is linguistically represented as a tree structure, which is used as a method for mapping conceptual relationships to a linguistically determined linearization frame of label-internal terms.

5 Conclusion and Future Work

For transforming OWL concepts to NL patterns we use elaborated DCG parsing techniques, which allow transforming and splitting of synthesized labels. Our approach presupposes a systematic definition of OWL labels based on linguistic patterns and labeling style guidelines.

Future work should include a systematic way of defining grouping rules for sentence blocks and a well-defined, finer-granulated set of style guidelines for OWL label generation. For the lexicalization purposes of revised OWL class and property labels we can use KCPM[5], which allows a glossary-representation of the cleared and split up labeling contents.

References

1. Amber S.W.: UML 2 Class diagram Guidelines http://www.agilemodeling.com/style/classDiagram.htm#ClassGuidelines
2. Bechhofer, S., Horrocks, I., Goble, C., Stevens, R.: OilEd: a Reasonable Ontology Editor for the Semantic Web. In: Baader, F., Brewka, G., Eiter, T. (eds.) KI 2001. LNCS(LNAI), vol. 2174, pp. 396–408. Springer, Heidelberg (2001)
3. Fliedl, G.: Natürlichkeitstheoretische Morphosyntax – Aspekte der Theorie und Implementierung. Gunter Narr Verlag Tübingen (1999)
4. Fliedl, G., Kop, C., Mayr, H.C., Winkler, C., Hölbling, M., Horn, T., Weber, G.: Extended Tagging and Interpretation Tools for Mapping Requirements Texts to Conceptual (Predesign) Models. In: Montoyo, A., Muńoz, R., Métais, E. (eds.) NLDB 2005. LNCS, vol. 3513, pp. 173–180. Springer, Heidelberg (2005)
5. Fuchs, N.E., Höfler, S., Kaljurand, K., Rinaldi, F., Schneider, G.: Attempto Controlled English: A Knowledge Representation Language Readable by Humans and Machines. In: Eisinger, N., Małuszyński, J. (eds.) Reasoning Web. LNCS, vol. 3564, pp. 213–250. Springer, Heidelberg (2005)
6. Hewlett, D., Kalyanpur, A., Kolovski, V., Halaschek-Wiener, C.: Effective Natural Language Paraphrasing of Ontologies on the Semantic Web. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, Springer, Heidelberg (2005)
7. Kalyanpur, A., Parsia, B., Sirin, E., Cuenca-Grau, B., Hendler, J.: Swoop: A 'Web' Ontology Editing Browser. Journal of Web. Semantics 4(2), 144–153 (2005)
8. Kristiansen, F.: PHP Coding Standard http://www.dagbladet.no/development/phpcoding standard/#names
9. Krüger, M.: C# Coding Style Guide http://www.csharpfriends.com/Articles/getArticle.aspx?articleID=336#8
10. Mayr, H.C.; Kop, C.: A User Centered Approach to Requirements Modeling. In: Proc. Modellierung'2002, GI-Edition LNI P-12, pp. 75–86 (2002)
11. McGuinness, D.L., van Harmelen, F.: OWL Web Ontology Language Overview http://www.w3.org/TR/owl-features/
12. Noy, N., Sintek, M., Decker, S., Crubezy, M., Fergerson, R., Musen, M.: Creating semantic web contents with Protege-2000, IEEE Intelligent Systems, pp. 60–71(2001)

[5] KCPM: Klagenfurt Conceptual Predesign Model [10].

13. Sun Microsystems, Code Conventions for the JavaTM Programming Language (1999), http://java.sun.com/docs/codeconv/
14. Sure, Y., Erdmann, M., Angele, J., Staab, S., Studer, R., Wenke, D.: OntoEdit: Collaborative Ontology Engineering for the Semantic Web. In: Horrocks, I., Hendler, J. (eds.) ISWC 2002. LNCS, vol. 2342, pp. 221–235. Springer, Heidelberg (2002)
15. Resource Description Framework, http://www.w3.org/RDF/
16. Extensible Markup Language (XML), http://www.w3.org/XML/
17. DAML Ontology Library, http://www.daml.org/ontologies/

Ontological Text Mining of Software Documents

René Witte[1], Qiangqiang Li[1], Yonggang Zhang[2], and Juergen Rilling[2]

[1] Institut für Programmstrukturen und Datenorganisation (IPD)
Universität Karlsruhe (TH), Germany
[2] Department of Computer Science and Software Engineering
Concordia University, Montréal, Canada

Abstract. Documents written in natural languages constitute a major part of the software engineering lifecycle artifacts. Especially during software maintenance or reverse engineering, semantic information conveyed in these documents can provide important knowledge for the software engineer. In this paper, we present a text mining system capable of populating a software ontology with information detected in documents.

1 Introduction

With the ever increasing number of computers and their support for business processes, an estimated 250 billion lines of source code were being maintained in 2000, with that number rapidly increasing [1]. The relative cost of maintaining and managing the evolution of this large software base now represents more than 90% of the total cost [2] associated with a software product. One of the major challenges for software engineers while performing a maintenance task is the need to comprehend a multitude of often disconnected artifacts created originally as part of the software development process [3]. These artifacts include, among others, source code and corresponding software documents, e.g., requirements specifications, design description, and user's guides. From a maintainer's perspective, it becomes essential to establish and maintain the semantic connections among all these artifacts. Automated source code analysis, implemented in integrated development environments like *Eclipse*, has improved software maintenance significantly. However, integrating the often large amount of corresponding documentation requires new approaches to the analysis of natural language documents that go beyond simple full-text search or information retrieval (IR) techniques [4].

In this paper, we propose a Text Mining (TM) approach to analyse software documents on a semantic level. A particular feature of our system is its use of formal ontologies (in OWL-DL format) during both, the analysis process and as an export format for the results. In combination with a source code analysis system for populating code-specific parts of the ontology, we can now represent knowledge concerning *both* code *and* documents in a single, unified representation. This common, formal representation supports further analysis of the knowledge base, like the automatic establishment of traceability links. A general overview of the proposed process is shown in Fig. 1.

Z. Kedad et al.(Eds.): NLDB 2007, LNCS 4592, pp. 168–180, 2007.

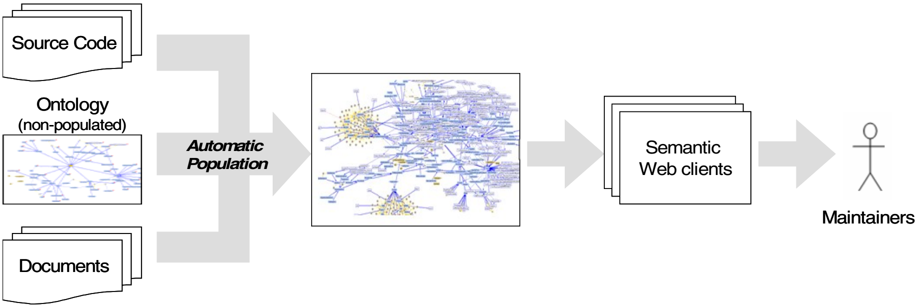


Fig. 1. Ontological Text Mining of software documents for software engineering

2 Ontological Text Mining for Software Documents

In this section, we present a brief motivation and overview of our ontology-based software environment and then discuss the text mining component in detail.

2.1 Software Engineering and NLP

As software ages, the task of maintaining it becomes more complex and more expensive. Software maintenance, often also referred to as software evolution, constitutes a majority of the total cost occurring during the life span of a software system [1,2]. Software maintenance is a multi-dimensional problem space that creates an ongoing challenge for both the research community and tool developers [5,6]. These maintenance challenges are caused by the different representations and interrelationships that exist among software artifacts and knowledge resources [7,8]. From a maintainer's perspective, exploring [9] and linking these artifacts and knowledge resources becomes a key challenge [4]. What is needed is a unified representation that allows maintainers to explore, query and reason about these artifacts, while performing their maintenance tasks [10].

Information contained in software documents is important for a multitude of software engineering tasks, but within this paper, we focus on a particular use case: the concept location and traceability across different software artifacts. From a maintainer's perspective, software documentation contains valuable information of both functional and non-functional requirements, as well as information related to the application domain. This knowledge often is difficult or impossible to extract only from source code [11]. It is a well-known fact that even in organizations and projects with mature software development processes, software artifacts created as part of these processes end up to be disconnected from each other [4]. As a result, maintainers have to spend a large amount of time on synthesizing and integrating information from various information sources in order to re-establish the traceability links among these artifacts.

Our approach is based on a common formal representation of both source code and software documentation using an ontology in OWL-DL format [12]. Instances are populated automatically through automatic code analysis (described in [13]) and text mining (described in this paper).

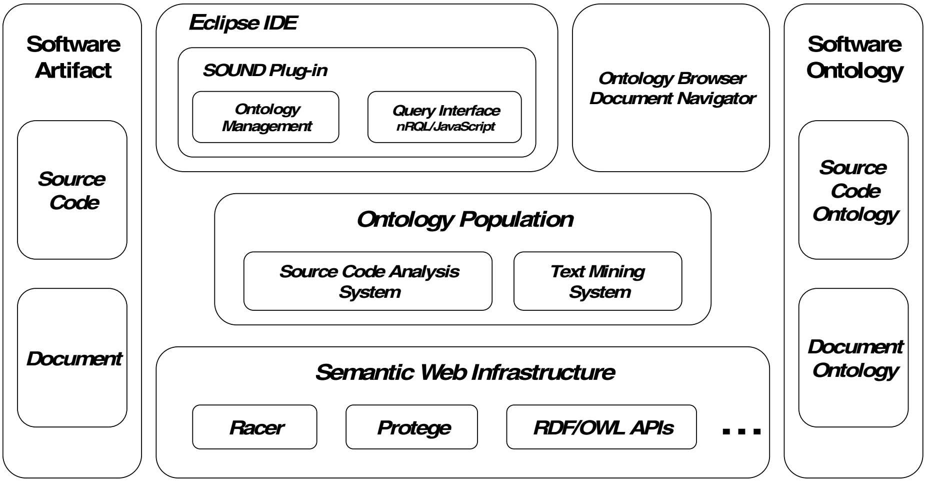


Fig. 2. Ontology-based program comprehension environment overview

2.2 System Architecture and Implementation Overview

In order to utilize the structural and semantic information in various software artifacts, we have developed an ontology-based program comprehension environment, which can automatically extract concept instances and their relations from source code and documents (Fig. 2).

An important part of our architecture is a software ontology that captures major concepts and relations in the software maintenance domain. This ontology consists of two sub-ontologies: a *source code* and *document* ontology, which represent information extracted from source code and documents, respectively. The ontologies are modeled in OWL-DL and were created using the OWL extension of Protégé,[1] a free ontology editor. Racer [14], an ontology inference engine, is integrated to provide reasoning services. Racer is a highly optimized DL system that supports reasoning about instances, which is particularly useful for the software maintenance domain, where a large amount of instances needs to be handled efficiently. Automatic ontology population is handled by two subsystems: The source code analysis, which is based on the JDT Java parser[2] provided by Eclipse [13]; and the document analysis using the text mining system discussed in this paper. The query interface of our system is a plug-in that provides OWL integration for Eclipse, a widely used software development platform. The expressive query language nRQL provided by Racer can be used to query and reason over the populated ontology. Additionally, we integrated a scripting language, which provides a set of built-in functions and classes using

[1] Protégé ontology editor, http://protege.stanford.edu/

[2] Eclipse Java Development Tools (JDT), http://www.eclipse.org/jdt/

the JavaScript interpreter Rhino.[3] This language simplifies querying the ontology for software engineers not familiar with DL-based formalisms.

2.3 Software Document Ontology

The documentation ontology consists of a large body of concepts that are expected to be discovered in software documents. These concepts are based on various programming domains, including programming languages, algorithms, data structures, and design decisions such as design patterns and software architectures. Additionally, the software documentation sub-ontology has been specifically designed for automatic population through a text mining system. In particular, we included: (1) A *Text Model* to represent the structure of documents, e.g., classes for sentences, paragraphs, and text positions, as well as NLP-related concepts that are discovered during the analysis process, like noun phrases (NPs) and coreference chains. These are required for anchoring detected entities (populated instances) in their originating documents. (2) *Lexical Information* facilitating the detection of entities in documents, like the names of common design patterns, programming language-specific keywords, or architectural styles. (3) *Lexical normalization rules* for entity normalization. (4) *Relations* between the classes, which extend the ones modeled in the source code ontology. These allow us to automatically restrict NLP-detected relations to semantically valid ones. For example, a relation like <variable> implements <interface>, which can result from parsing a grammatically ambiguous sentence, can be filtered out since it is not supported by the ontology. Finally, (5) *Source Code Entities* that have been automatically populated through source code analysis can also be utilized for detecting corresponding entities in documents, as we describe below.

2.4 Ontology Population Through Text Mining

We developed our text mining system for populating the software documentation ontology based on the GATE *(General Architecture for Text Engineering)* framework [15]. The system is component-based, utilizing both standard tools shipped with GATE and custom components developed specifically for software text mining. An overview of the workflow is shown in Fig. 3. In the following discussion, we omit several standard NLP analysis steps, like part-of-speech (POS) tagging, noun phrase (NP) chunking, or stemming. For readers unfamiliar with these tasks, we refer to the GATE user's guide.[4]

Ontology Initialization. When analysing documents specific to a source code base, our text mining system can take instances detected by the automatic code analysis into account. This is achieved in two steps: first, the source code ontology is populated with information detected through static and dynamic code analysis [13]. This step adds instances like method names, class names, or detected design patterns to the software ontology. In a second step, we use this information as additional input to the OntoGazetteer component for named entity recognition.

[3] Rhino JavaScript interpreter, `http://www.mozilla.org/rhino/`

[4] GATE user's guide, `http://gate.ac.uk/sale/tao/index.html`

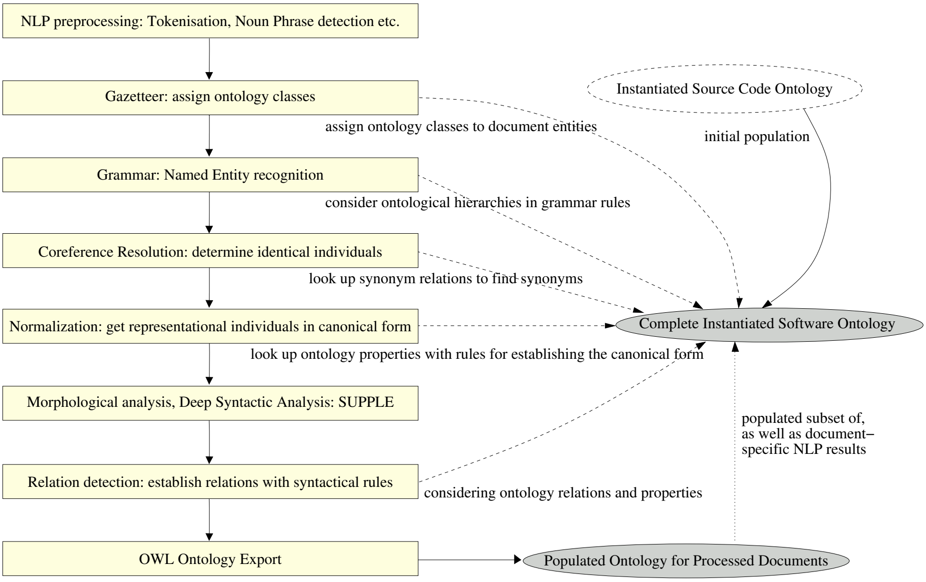

NLP preprocessing: Tokenisation, Noun Phrase detection etc.
Gazetteer: assign ontology classes
Grammar: Named Entity recognition
Coreference Resolution: determine identical individuals
Normalization: get representational individuals in canonical form
Morphological analysis, Deep Syntactic Analysis: SUPPLE
Relation detection: establish relations with syntactical rules
OWL Ontology Export
Instantiated Source Code Ontology
initial population
assign ontology classes to document entities
consider ontological hierarchies in grammar rules
look up synonym relations to find synonyms
Complete Instantiated Software Ontology
look up ontology properties with rules for establishing the canonical form
populated subset of, as well as document–specific NLP results
considering ontology relations and properties
Populated Ontology for Processed Documents

Fig. 3. Workflow of the software text mining subsystem

Named Entity Detection. The basic process in GATE for recognizing entities of a particular domain starts with the gazetteer component. It matches given lists of terms against the tokens of an analysed text and, in case of a match, adds an annotation named `Lookup` whose features depend on the list where the match was found. Its ontology-aware counterpart is the *OntoGazetteer*, which incorporates mappings between its term lists and ontology classes and assigns the proper class in case of a term match. For example, using the instantiated software ontology, the gazetteer will annotate the text segment *method* with a `Lookup` annotation that has its `class` feature set to "Method." Here, incorporating the results from automatic code analysis can significantly boost recall (cf. Section 3), since entity names in the software domain typically do not follow naming rules, like in the biological domain.

In a second step, grammar rules written in the JAPE[5] language are used to detect and annotate complex named entities. Those rules can refer to the `Lookup` annotation generated by the OntoGazetteer, and also evaluate the ontology directly. For example, when performing a comparison like `class=="Keyword"` in a grammar rule, the ontological hierarchy is taken into account so that also a Java_keyword matches, since a Java keyword *is-a* keyword in the ontology. This feature significantly reduces the overhead for grammar development and testing.

The developed JAPE rules combine ontology-based lookup information with noun phrase (NP) chunks to detect semantic units. NP chunking is performed

[5] JAPE is a regular-expression based language for writing grammars over annotations, from which finite-state transducers are generated by a GATE component.

using the MuNPEx chunker,[6] which relies mostly on part-of-speech (POS) tags, but can also take the lookup information into account. This way, it can prevent bad NP chunks caused by mis-tagged software entities (e.g., method names or program keywords tagged as verbs). Essentially, we combine two complementary approaches for entity detection: A *keyword*-based approach, relying on lexical information stored in the documentation ontology (see above). For example, the text segment *"the storeAttribute() method..."* will be annotated with a lookup information indicating that the word *method* belongs to the ontology class "Method." Likewise, the same segment will be annotated as a single noun phrase, showing determiner *("the")*, modifier *("storeAttribute()")*, and head noun *("method")*. Using an ontology-based grammar rule implemented in JAPE, we can now combine these two information and semantically mark the NP as a method. Similar rules are used to detect variables, class names, design patterns, or architectural descriptions. Note that this approach does not need to know about "storeAttribute()" being a method name; this fact is induced from a combination of grammatical (NP chunks) and lexical (ontology) information.

The second approach relies on source code analysis results stored in the initialized software ontology (see above). Every method, class, package, etc. name will be automatically represented by an instance in the source code sub-ontology and can thus be used by the OntoGazetteer for entity detection. This applies also in case when these instances appear outside a grammatical construct recognized by our hand-crafted rules. This is especially useful for analysing software documents in conjuction with their source code, the primary scenario our system was designed for.

Coreference Resolution. We use a fuzzy set theory-based coreference resolution system [16] for grouping detected entities into *coreference chains*. Each chain represents an equivalence class of textual descriptors occurring within or across documents. Not surprisingly, our fuzzy heuristics developed originally for the news domain (e.g., using *WordNet*) were particularly ineffective for detecting coreference in the software domain. Hence, we developed an extended set of heuristics dealing with both pronominal and nominal coreferences.

For *nominal coreferences*, we rely on three main heuristics. The first is based on simple string equality (ignoring case). The second heuristic establishes coreference between two entities if they become identical when their NPs' HEAD and MOD slots are inverted, as in *"the selectState() method"* and *"method selectState()"*. The third heuristic deals with a number of grammatical constructs often used in software documents that indicate synonymous entities. For example, in the text fragment *"... we have an action class called ViewContentAction, which is invoked."* we can identify the NPs *"an action class"* and *"ViewContentAction"* as being part of the same coreference chain. This heuristic only considers entities of the same ontology class, connected by a number of predefined relation words (e.g., "named", "called"), which are also stored in the ontology.

[6] MuNPEx, http://www.ipd.uka.de/~durm/tm/munpex/

Table 1. Lexical normalization rules for various ontology classes

Ontology Class	H	DH	MH(cM)	MH(cH)	DMH(cM)	DMH(cH)
Class	H	H	H	lastM	H	lastM
Method	H	H	H	lastM	H	lastM
LayeredArchitecture	H	H	MH	MH	MH	MH
AbstractFactory	H	H	MH	MH	MH	MH
OO_Interface	H	H	H	lastM	H	lastM

For *pronominal resolution*, we implemented a number of simple sub-heuristics dealing only with 3rd person singular and plural pronouns: *it, they, this, them,* and *that.* The last three can also appear in qualified form *(this method, that constructor).* We employ a simple resolution algorithm, searching for the closest anaphorical referent that matches the case and, if applicable, the semantic class.

Normalization. Normalization needs to decide on a canonical name for each entity, like a class or method name. This is important for ontology population, as an instance, like of the ontology class Method, should reflect only the method name, omitting any additional grammatical constructs like determiners or possessives. Thus, a named entity like *"the static TestClass() constructor"* has to be normalized to "TestClass" before it can become an instance (ABox) of the class Method (TBox) in the populated ontology.

This step is performed through a set of lexical normalization rules, which are stored with their corresponding classes in the software document sub-ontology, allowing us to inherit rules through subsumption. Table 1 shows a number of these rules for various ontology classes: D, M, H refer to determiner, modifier, and head, respectively, and c(x) denotes the ontology class of a particular slot; the table entry determines what part of a noun phrase is selected as the normalized form, which is then stored as a feature in the entity's annotation.

Relation Detection. The next major step is the detection of *relations* between entities, e.g., to find out which interface a class is implementing, or which method belongs to which class. Relation detection in our system is again done with two complementary approaches: a set of hand-crafted grammar rules implemented in JAPE, and a deep syntactic analysis using the SUPPLE parser. Afterwards, detected relations are filtered through the software ontology to erase semantically invalid results. We now describe these steps in detail.

Rule-Based Relation Detection. Similarly to entity recognition, rule-based relation detection is performed in a two-step process: first, a JAPE-based transducer is run to detect verb groups (VGs) based on POS tags. Then, tokens that are candidates for relation predicates (e.g., "implements," "extends") are marked by the OntoGazetteer. Combining these two information, we can create custom JAPE rules to detect relations between entities detected previously, for example, to find the classes creating a certain design pattern or to find relations between described classes and methods. Using the voice information (active/passive) provided by the VG chunker, we can then assign subject/object slots for the entities participating in a relation.

Deep Syntactic Analysis. For a deep syntactic analysis, we currently employ the SUPPLE parser [17], which is integrated into GATE through a wrapper component. SUPPLE is a general-purpose bottom-up chart parser for feature-based context-free phrase structure grammars, implemented in Prolog. It produces syntactic as well as semantic annotations to a given sentence. Grammars are applied in series allowing to choose the best parse for each step and continue to the next layer of grammatical analysis with only the selected best parse. The identification of verbal arguments and attachment of nominal and verbal post-modifiers, such as prepositional phrases and relative clauses, is done conservatively. Instead of producing all possible analyses or using probabilities to generate the most likely analysis, SUPPLE only offers a single analysis that spans the input sentence only if it can be relied on to be correct, so that in many cases only partial analyses are produced. SUPPLE outputs a logical form, which is then matched with the entities detected previously to obtain predicate-argument structures.

Result Integration. The results from both rule- and parser-based relation detection form the candidate set for ontology relation instances created for a text. As both approaches may result in false positives, e.g., through ambiguous syntactical structures or rule mismatches, we prune the set by checking each candidate relation for semantic correctness using our software ontology. As each entity participating in a relation has a corresponding ontology class, we can query the ontology to check whether the detected relation (or one of its supertypes) exists between these classes. This way, we can filter out semantically incorrect relations like a *variable* "implementing" an *interface* or a *design pattern* being "part-of" a *class*, thereby significantly improving precision (cf. Section 3).

Note that relation detection and filtering is one particular example where an ontology delivers additional benefit when compared with classical NLP techniques like plain gazetteering lists or statistical/rule-based systems [18].

Ontology Export. Finally, the instances found in the document and the relations between them are exported to an OWL-DL ontology. Note that entities provided by source code analysis are only exported in the document ontology if they have also been detected in a text (cf. Fig. 3).

In our implementation, ontology population is done by a custom GATE component, the *OwlExporter*, which is application domain-independent. It collects two special annotations, `OwlExportClass` and `OwlExportRelation`, which specify instances of classes and relations (i.e., object properties), respectively. These must in turn be created by application-specific components, since the decisions as to which annotations have to be exported, and what their OWL property values are, depend on the domain.

The class annotation carries the name of the class, a name for the instance (the normalized name created previously), and the GATE internal ID of an annotation representing the instance in the document. If there are several occurrences of the same entity in the document, the final representation annotation

Table 2. Evaluation results: Entity recognition and normalization performance

	Text Mining Only				**With Source Ontology**			
Corpus	*P*	*R*	*F*	*A*	*P*	*R*	*F*	*A*
Java Collections	0.89	0.67	0.69	75%	0.76	0.87	0.79	88%
uDig	0.91	0.57	0.59	82%	0.58	0.87	0.60	84%
Total	0.90	0.62	0.64	77%	0.67	0.87	0.70	87%

is chosen from the ones in the coreference chain by the component creating the `OwlExportClass` annotation. In case of the software text mining system, a single representative has to be chosen from each coreference chain. Remember that one chain corresponds to a single semantic unit, so the final, exported ontology must only contain one entry for, e.g., a method, not one instance for every occurrence of that method in a document set. We select the representative using a number of heuristics, basically assuming that the longest NP that has more slots (DET, MOD, HEAD) filled is also the most salient one.

From this representative annotation, all further information is gathered. After reading the class name, the OwlExporter queries the ontology via the Jena[7] framework for the class properties and then searches for equally named features in the representation annotation, using their values to set the OWL properties.

3 Evaluation

So far, we evaluated our text mining subsystem on two collections of texts: a set of 5 documents (7743 words) taken from the Java documentation for the *Collections* framework[8] and a set of 7 documents (3656 words) from the documentation of the uDig[9] geographic information system (GIS). The document sets were chosen because of the availability of the corresponding source code.

Both sets were manually annotated for named entities, including their ontology classes and normalized form, as well as relations between the entities. In what follows, we present results for the named entity recognition, entity normalization, and relation detection tasks.

Named Entity Recognition Evaluation. We computed the standard precision, recall, and F-measure results for NE detection. A named entity was only counted as correct if it matched both the textual description and ontology class. Table 2 shows the results for two experiments: first running only the text mining system over the corpora (left side) and second, performing the same evaluation after running the code analysis, using the populated source code ontology as an additional resource for NE detection as described above. As can be seen, the text mining system achieves a very high precision (90%) in the NE detection task,

[7] Jena Semantic Web Framework for Java, http://jena.sourceforge.net/

[8] Java Collections Framework Documentation, http://java.sun.com/j2se/1.5.0/docs/guide/collections/index.html

[9] uDig GIS Documentation, http://udig.refractions.net/

Table 3. Evaluation results: Relation detection performance

Corpus	**Before Filtering**			**After Filtering**			
	P	*R*	*F*	*P*	*R*	*F*	*ΔP*
Text Mining Only							
Java Collections	0.35	0.24	0.29	0.50	0.24	0.32	30%
uDig	0.46	0.34	0.39	0.55	0.34	0.42	16%
Total	0.41	0.29	0.34	0.53	0.29	0.37	23%
With Source Ontology							
Java Collections	0.14	0.36	0.20	0.20	0.36	0.25	30%
uDig	0.11	0.41	0.17	0.24	0.41	0.30	54%
Total	0.13	0.39	0.19	0.22	0.39	0.23	41%

with a recall of 62%. With the imported source code instances, these numbers become reversed: the system can now correctly detect 87% of all entities, but with a lower precision of 67%.

The drop in precision after code analysis is mainly due to two reasons. Since names in the software domain do not have to follow any naming conventions, simple nouns or verbs often used in a text will be mis-tagged after being identified as an entity appearing in a source code. For example, the Java method *sort* from the collections interface will cause all instances of the word "sort" in a text to be marked as a method name. Another precision hit is due to the current handling of class constructor methods, which are typically identical to the class name. Currently, the system cannot distinguish the class name from the constructor name, assigning both ontology classes (i.e., Constructor and OO_CLass) for a text segment, where one will always be counted as a false positive.

Both cases require additional strategies when importing entities from source code analysis, which are currently under development. However, the current results already underline the feasibility of our approach of integrating code analysis and NLP.

Entity Normalization Evaluation. We also evaluated the performance of our lexical normalization rules for entity normalization, since correctly normalized names are a prerequisite for the correct population of the result ontology. For each entity, we manually annotated the normalized form and computed the accuracy *A* as the percentage of correctly normalized entities over all correctly identified entities. Table 2 shows the results for both the system running in text mining mode alone and with additional source code analysis. As can be seen from the table, the normalization component performs rather well.

Relation Detection Evaluation. Not surprisingly, relation detection was the hardest subtask within the system. Like for entity detection, we performed two different experiments, with and without source code analysis results. Additionally, we evaluated the influence of the semantic relation filtering step using our ontology as described above. The results are summarized in Table 3. As can be seen, the current combination of rules with the SUPPLE parser does not achieve a high performance. However, the increase in precision (ΔP) when applying the filtering step using our ontology is significant: upto 54% better than without semantic filtering.

The overall low precision and recall values are mainly due to the unchanged SUPPLE parser rules, which have not yet been adapted to the software domain. Also, the conservative PP-attachement strategy of SUPPLE misses many predicate-argument structures. We currently experiment with different parsers (RASP and MiniPar) and are also adapting the SUPPLE grammar rules in order to improve the detection of predicate-argument structures.

4 Related Work and Discussion

Very little previous work exists on text mining software documents. Most of this research has focused on analysing texts at the specification level, e.g., in order to automatically convert use case descriptions into a formal representation [19,20] or detect inconsistent requirements [21]. In contrast, we aim to support the complete software documentation life-cycle, from white papers, design and implementation documents to in-line code texts (e.g., JavaDoc). To the best of our knowledge, there has been so far no attempt to automatically combine source code analysis with the text mining of software documents, which is an important contribution of our work.

There exists some research in recovering traceability links between source code and design documents using Information Retrieval techniques. The IR models used include traditional vector space and probabilistic models [4], as well as latent semantic indexing (LSI) [22]. In contrast with these IR approaches, our work also takes advantage of structural and semantic information in both the documentation and source code by means of text mining and source code parsing.

5 Conclusions and Future Work

We presented a text mining system for the software domain that is capable of extracting entities from software documents. The system's output is a populated OWL-DL ontology containing normalized instances and their relations. The system is novel in two important aspects: First, it employs a formal ontology, based on description logics, both as a processing resource for the various NLP components and the result export format. Second, as the system is part of a larger ontology-based program comprehension environment, it can incorporate results from automated source code analysis subsystems in its NLP processing pipeline.

The ontological foundation allows for important improvements in software engineering, as it supports *queries* and *reasoning* services on semantic knowledge automatically derived from large amounts of documentation in natural language form. We previously showed how automated reasoning can support a software maintainer when performing knowledge-intensive tasks, like architectural recovery or source code security analysis. We are also currently experimenting with ontology alignment strategies to automatically establish links between code and its corresponding documentation [13]. This will allow, for the first time, the automatic establishment and analysis of traceability links, which is of high importance for the software industry.

Besides improving the individual components as discussed in the evaluation section, we plan to extend our system to explicitly deal with documents associated with the different steps in the software life-cycle, from white papers and requirements over design and implementation documents to user's guides and source code comments. This will allow us to trace concepts and entities across the different states of software development and different levels of abstraction.

References

1. Sommerville, I.: Software Engineering, 6th edn. Addison-Wesley, Reading (2000)
2. Seacord, R., Plakosh, D., Lewis, G.: Modernizing Legacy Systems: Software Technologies, Engineering Processes, and Business Practices. SEI Series in SE. Addison-Wesley, Reading (2003)
3. Jin, D., Cordy, J.: Ontology-Based Software Analysis and Reengineering Tool Integration: The OASIS Service-Sharing Methodology. In: 21st IEEE International Conference on Software Maintenance (ICSM) (2005)
4. Antoniol, G., Canfora, G., Casazza, G., Lucia, A.D.: Information retrieval models for recovering traceability links between code and documentation. In: Proc. of IEEE Intl. Conf. on Software Maintenance, San Jose, CA, USA (2000)
5. IEEE: IEEE Standard for Software Maintenance. IEEE 1219 (1998)
6. Riva, C.: Reverse Architecting: An Industrial Experience Report. In: 7th IEEE Working Conference on Reverse Engineering (WCRE), pp. 42–52 (2000)
7. Storey, M.A., Sim, S.E., Wong, K.: A Collaborative Demonstration of Reverse Engineering tools. ACM SIGAPP Applied Computing Review 10(1), 18–25 (2002)
8. Welty, C.: Augmenting Abstract Syntax Trees for Program Understanding. In: Proc. of Int. Conf. on Automated Software Engineering, pp. 126–133. IEEE Computer Society Press, Los Alamitos (1997)
9. Lethbridge, T.C., Nicholas, A.: Architecture of a Source Code Exploration Tool: A Soft-ware Engineering Case Study. Technical Report TR-97-07, Department of Computer Science, University of Ottawa (1997)
10. Meng, W., Rilling, J., Zhang, Y., Witte, R., Charland, P.: An Ontological Software Comprehension Process Model. In: 3rd Int. Workshop on Metamodels, Schemas, Grammars, and Ontologies for Reverse Engineering (ATEM, Genoa, Italy (October 1st 2006), pp. 28–35 (2006)
11. Lindvall, M., Sandahl, K.: How well do experienced software developers predict software change? Journal of Systems and Software 43(1), 19–27 (1998)
12. Johnson-Laird, P.N.: Mental Models: Towards a Cognitive Science of Language, Inference and Consciousness. Harvard University, Cambridge, MA (1983)
13. Rilling, J., Witte, R., Zhang, Y.: Automatic Traceability Recovery: An Ontological Approach. In: International Symposium on Grand Challenges in Traceability (GCT'07), Lexington, Kentucky, USA (March 22–23, 2007)
14. Haarslev, V., Möller, R., RACER,: System Description. In: Goré, R.P., Leitsch, A., Nipkow, T. (eds.) IJCAR 2001. LNCS (LNAI), vol. 2083, pp. 701–705. Springer, Heidelberg (2001)

15. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: A framework and graphical development environment for robust NLP tools and applications. In: Proc. of the 40th Anniversary Meeting of the ACL (2002)
16. Witte, R., Bergler, S.: Fuzzy Coreference Resolution for Summarization. In: Proceedings of 2003 International Symposium on Reference Resolution and Its Applications to Question Answering and Summarization (ARQAS), Venice, Italy, Università Ca' Foscari (June 23–24 2003), pp. 43–50 http://rene-witte.net
17. Gaizauskas, R., Hepple, M., Saggion, H., Greenwood, M.A., Humphreys, K.: SUPPLE: A practical parser for natural language engineering applications. In: Proc. of the 9th Intl. Workshop on Parsing Technologies (IWPT2005), Vancouver (2005)
18. Witte, R., Kappler, T., Baker, C.J.O.: Ontology Design for Biomedical Text Mining. In: Semantic Web: Revolutionizing Knowledge Discovery in the Life Sciences, pp. 281–313. Springer, Heidelberg (2006)
19. Mencl, V.: Deriving behavior specifications from textual use cases. In: Proceedings of Workshop on Intelligent Technologies for Software Engineering, Linz, Austria, Oesterreichische Computer Gesellschaft, pp. 331–341 (2004)
20. Ilieva, M., Ormandjieva, O.: Automatic transition of natural language software requirements specification into formal presentation. In: Montoyo, A., Muńoz, R., Métais, E. (eds.) NLDB 2005. LNCS, vol. 3513, pp. 392–397. Springer, Heidelberg (2005)
21. Kof, L.: Natural language processing: Mature enough for requirements documents analysis? In: Montoyo, A., Muńoz, R., Métais, E. (eds.) NLDB 2005. LNCS, vol. 3513, pp. 91–102. Springer, Heidelberg (2005)
22. Marcus, A., Maletic, J.I.: Recovering Documentation-to-Source-Code Traceability Links using Latent Semantic Indexing. In: Proc. of 25th Intl. Conf. on Software Engineering (2002)

Treatment of Passive Voice and Conjunctions in Use Case Documents

Leonid Kof

Fakultät für Informatik, Technische Universität München,
Boltzmannstr. 3, D-85748 Garching bei München, Germany
kof@informatik.tu-muenchen.de

Abstract. Requirements engineering, the first phase of any software development project, is the Achilles' heel of the whole development process, as requirements documents are often inconsistent and incomplete. In industrial requirements documents natural language is the main presentation means. In such documents the system behavior is specified in the form of use cases and their scenarios, written as a sequence of sentences in natural language. For the authors of requirements documents some facts are so obvious that they forget to mention them. This surely causes problems for the requirements analyst.

Missing information manifests itself, for example, in sentences in passive voice: such sentences just say that some action is performed, but they do not say who performs the action. In the case of requirement analysis this poses a serious problem, as in every real system there is an actor for every performed action.

There already exists an approach able to guess missing actors and actions. However, the existing approach is able to handle sentences containing exactly one verb only. The approach presented in this paper extends the existing one by treatment of compound sentences and passive voice. Feasibility of the presented approach to the treatment of passive and conjunctions was confirmed in a case study.

1 Document Authors Are Not Aware That Some Information Is Missing

Some kind of requirements document is usually written at the beginning of every software project. The majority of these documents are written in natural language, as the survey by Mich et al. shows [1]. This results in the fact that the requirements documents are imprecise, incomplete, and inconsistent. The authors of requirements documents are not always aware of these document defects. From the linguistic point of view, document authors introduce three defect types, without perceiving them as defects (cf. Rupp [2]):[1]

Deletion: "...is the process of selective focusing of our attention on some dimensions of our experiences whereas excluding other dimensions. Deletion reduces the world to the extent that we can handle."

[1] The following definitions are translations of the definition from [2] (in German).

Z. Kedad et al.(Eds.): NLDB 2007, LNCS 4592, pp. 181–192, 2007.

Generalization: "...is the process of detachment of the elements of the personal model from the original experience and the transfer of the original exemplary experience to the whole category of objects."

Distortion: "...is the process of reorganization of our sensory experience."

It is one of the goals of requirements analysis, to find and to correct the defects of requirements documents.

In requirements documents the behavior of the prospective system is often specified as a set of *use cases*, each use case represented by one or several *scenarios* (cf. Rupp [2]). A scenario is a sequence of natural language sentences. Each sentence of this sequence represents either some input to the system or the reaction of the system to previous inputs.

The presented paper focuses on the "deletion"-defects in scenarios. Deletion manifests itself in scenarios in the form of missing action subjects or objects or even in whole missing actions. One of the reasons for the deletion may be the fact that some information is too obvious for the author of the requirements document, so that she finds it unnecessary to write down this information. One further reason for missing action subjects, manifesting itself in sentences in passive voice, can be the absence of an exact construction plan, typical in the early stages of the project. It is the goal of the approach presented in this paper, to identify missing parts of scenarios written in natural language and to produce message sequence charts (MSCs) containing the reconstructed information. (See Section 3 for an introduction to MSCs.)

For the remainder of the paper we use the following terminology: A *scenario* is a sequence of natural language sentences, each sentence representing some *action*. A *message sequence chart (MSC)* is a set of *communicating objects* and a sequence of *messages* sent/received by these objects.

The remainder of the paper is organized as follows: Section 2 introduces the case study used to evaluate the presented approach. Section 3 introduces message sequence charts (MSCs) and an existing approach transforming scenarios to MSCs. This approach works only for sentences in active voice, containing exactly one verb. Section 4 explains an extention of this approach, allowing both for passive voice and for several verbs in the same sentence. Section 5 presents the evaluation of the approach on a case study. Finally, Sections 6 and 7 present an overview of related work and the summary of the paper, respectively.

2 Case Study: The Instrument Cluster

Authors of requirements documents tend to forget to write down facts that seem obvious to them. Even in a relatively precise requirements document, as for example the instrument cluster specification [3], some missing facts can be identified. The instrument cluster specification describes the optical design of one part of the car dashboard (the instrument cluster), its hardware, and, most importantly, its behavior. The behavior is specified as a set of scenarios, like this:

1. The driver switches on the car (ignition key in position ignition on).
2. The instrument cluster is turned on and stays active.

3. After the trip the driver switches off the ignition.
4. The instrument cluster stays active for 30 seconds and then turns itself off.
5. The driver leaves the car.

There are apparent problems if we try to translate this scenario to a sequence of messages exchanged by communicating objects. Firstly, there is no one-to-one correspondence between sentences and messages. For example, sentences number 2 and 4 contain two potential messages each: Sentence 2 contains actions "The instrument cluster is turned on" and "The instrument cluster stays active" and sentence 4 contains actions "The instrument cluster stays active for 30 seconds" and "The instrument cluster turns itself off". Furthermore, for at least one of these actions ("The instrument cluster is turned on") the actor is not explicitly specified. It is the goal of the approach presented in this paper, to resolve such incomplete specifications and present the results to a human analyst for validation.

3 Scenarios and Message Sequence Charts

Message Sequence Charts (MSCs) are a convenient means for concise and precise representation of action sequences. An MSC consists of a set of communicating objects. These communicating objects exchange messages, whereas every message has a well defined sender and receiver. Graphically, communicating objects are represented as rectangles, and messages as arrows; the time line is directed top down (cf. Figure 1).

When translating scenarios (written in natural language) to MSCs, it is necessary to deal with typical deficiencies of natural language texts: It can happen that either the message sender or the receiver are not explicitly mentioned, or the whole action is just omitted. For example, if we directly translate the scenario introduced in Section 2 to an MSC, a possible translation is the MSC in Figure 1[2]. The problems of this translation are apparent: there are definitely missing messages from the car to the instrument cluster, otherwise the instrument cluster cannot know that it should be turned on or off. Furthermore, some sentences, like "The instrument cluster is turned on", do not specify the message receiver. Even if we rephrase this sentence to active voice ("The instrument cluster turns on"), the message receiver remains unspecified.

The problem of unspecified message senders/receivers and missing actions was solved in [4] by the organization of MSC messages in a stack. Organization of messages in a stack is motivated by the idea of situation stack by Grosz et al. [5]. Grosz et al. introduce a situation stack to explain how the human attention focuses on different objects during a discourse. The focus depends on the sequence of sentence heard so far. By default, a sentence defines some situation and is pushed onto the stack. If a sentence reverts the effect of some previous sentence, the corresponding stack element is popped:

```
John enters the shop               //push "enter"
— Some actions in the shop —
John leaves the shop               //pop "enter" and the above stack elements
```

The idea of the situation stack can be easily transferred to MSCs: It is possible to define an active object as an object that has sent a message but has not received an

[2] To make the figure compacter, "instrument cluster" is abbreviated as "ins. clust.".

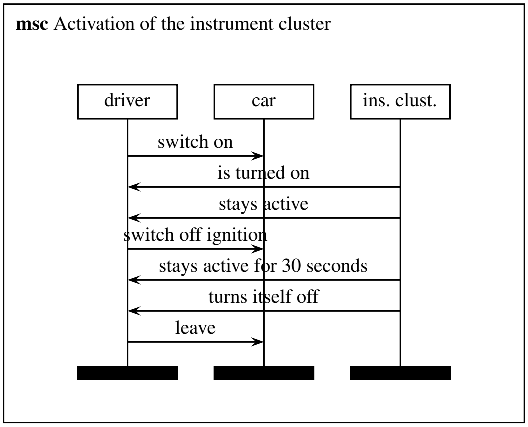


Fig. 1. Scenario “Activation of the instrument cluster”, manual translation to MSC

answer yet. If the receiver of the message under analysis (msg) is an active object, then it is possible to find the topmost message of the stack sent by this object (msg'). Then, msg' and the messages contained in the stack above it are popped. If the receiver is not an active object, the message under analysis is pushed onto the stack.

The organization of messages in a stack makes also the identification of missing messages possible: If the sender of the message under analysis ($sender_{new}$) differs from the receiver of the message on the top of the stack (rec_{top}), then the message from rec_{top} to $sender_{new}$ is missing. For example, for the MSC in Figure 1, missing messages from “car” to “instrument cluster” just after messages “switch on” and “switch off ignition” can be identified in this way. The message stack enables the identification of missing message senders and receivers as well: The default message sender/receiver equals to the receiver/sender of the message on the top of the stack. The details of the above algorithm can be found in [4].

The procedure of identification of senders and receivers apart from the analysis of the stack, as implemented in [4], is rather simple: It is assumed that every sentence contains exactly one verb. Furthermore, it is assumed that there exists a previously constructed list of potential communicating objects (glossary). Then, the longest word sequence before/after the verb that is contained in the glossary is identified as the message sender/receiver. If no such word sequence is found in the sentence, the sender/receiver remains unspecified for the concrete sentence. In this case the sender/receiver is augmented by the means of stack analysis.

The requirement that every sentence contain exactly one verb is obviously violated in passive and compound sentences. For example, the sentence “The instrument cluster is turned on and stays active” contains three verbs: “is”, “turned”, “stays”. In the case study performed in [4] such sentences were manually split and rewritten, so that in the resulting text every sentence contained exactly one verb. It is the goal of the approach presented in this paper, to extend the procedure of the identification of senders and receivers implemented in [4] onto sentences containing several verbs. This includes both passive voice and compound sentences.

4 Compound Sentences and Passive Voice: Translation to MSC Messages

The basic idea for the translation of compound sentences to MSC messages is fairly simple: We split every sentence into *elementary segments* and translate every segment to an MSC message. An *elementary segment* is defined as a sentence segment that does not contain any conjunctions or commas/colons. For example, the translation of the sentence “The instrument cluster is turned on and stays active” consists of two messages: Some unspecified object sends the command “turn on” to the instrument cluster and receives the answer “stays active” from the instrument cluster.

Generally, we want to take following issues into account when translating sentences to MSC messages:

- If we split the original sentence into elementary segments, it can happen that one of the segments lack the grammatical subject. For example, the sentence “The instrument cluster is turned on and stays active” would be split into “the instrument cluster is turned on” and “stays active”. The second segment lacks the subject. However, the subject is necessary to identify the message sender. This problem can be solved by propagation of the grammatical subject from the first segment of the sentence to the second one.
- Even when the grammatical subjects of the sentence segments coincide, the senders of the corresponding messages can differ. This is due to the fact that in passive sentences the grammatical subject corresponds to the message receiver, not to the sender. For example, in the translation of the segment “the instrument cluster is turned on”, “instrument cluster” is the receiver of the message “turn on”.
- If the sentence consists of several parts and some parts do not contain an own verb, the verbs should be accordingly propagated. For example, in the sentence “The driver drives more than 30 km/h and less than 50 km/h” the verb “drives” should be propagated to the segment “less than 50 km/h”.

As the technical means for splitting the sentences into segments and for identification of the verb we use a part-of-speech (POS) tagger in the presented approach. Such a tagger assigns a POS-tag (substantive, verb, adjective, . . .) to every word. Currently available taggers, as for example the tagger by Ratnaparkhi [6], have the precision of about 97%, which makes them unlikely to become an extra error source. The translation of tagged sentences to MSC messages goes in five steps:

1. The tagged sentences are split into elementary segments, not containing any conjunctions or commas/colons.
2. Every sentence segment is annotated as either active or passive or sentence segment without any verb.
3. For every sentence segment, the grammatical subjects, objects, and verbs are extracted, if possible.
4. The extracted grammatical subjects and objects are propagated to other sentence segments, if necessary.
5. Finally, for active segments the subjects are declared to message senders and objects to message receivers. For passive segments the assignment of senders and receivers is the opposite.

Every of these steps is explained below in detail.

Splitting of tagged sentences: The POS tagger by Ratnaparkhi appends a tag to every word using the underscore, so that the tagged sentence looks like this:

> The_DT instrument_NN cluster_NN is_VBZ turned_VBN on_RP and_CC stays_NNS active_JJ ._.

This form allows to split every sentence into elementary segments. As splitting marks we use the regular expression matching conjunctions: `" [^ ]*_CC "` (space, followed by a character sequence without spaces, followed by underscore, followed by the conjuction tag, followed by space)[3], and also regular expressions matching tagged punctuation: `" [^ ]*_, "` (matching coma), `" [^ ]*_\.[ ]*"` (matching period), and `" [^ ]*_:[ ]*"` (matching colon). The splitting mark matching conjunctions splits the sentence "The instrument cluster is turned on and stays active" into "The instrument cluster is turned on" and "stays active". The splitting mark matching punctuation would decompose constructions like "X, Y, and Z do something" into "X", "Y", and "Z do something".

Annotation of sentence segments: The annotation of sentence segments as either active or passive or sentence segment without verb is necessary for two reasons:

- For sentence segments without verbs the verbs have to be accordingly adopted from other segments.
- The mapping of grammatical subjects/objects to message senders/receivers is different for active and passive segments.

For the annotation of sentence segments it is possible to use regular expressions based on POS-tags, again. A tagged sentence segment is annotated as passive if and only if it matches the regular expression `".*` $\langle be - form \rangle$`.*VBN.*"` (any character sequence, followed by some form of the verb "to be", followed by a verb participle[4], followed by any character sequence). In this expression $\langle be - form \rangle$ can be equal to "be", "am", "are", "is", "was", "were", or "been". For example, the segment "the_DT instrument_NN cluster_NN is_VBZ turned_VBN on_RP" is annotated as passive because the verb "is" is followed by the participle "turned_VBN".

If the tagged segment does not match the regular expression `".*_VB.*"` (i.e., it does not contain any verb tag), it is annotated as "segment without verb". Otherwise, if the segment contains a verb but does not match any of the passive expressions, the segment is annotated as active, as for example the segment "the_DT driver_NN leaves_VBZ the_DT car_NN".

Extraction of subjects and objects: To extract subjects and objects, active sentence segments are spilt on the basis of the POS-tags into three parts: the verb, the word sequence before the verb, and the word sequence after the verb. Passive sentence segments are split into four parts: the auxiliary verb, the word sequence before the auxiliary verb, the

[3] Here the Java syntax for regular expressions is used. For details see http://java.sun.com/j2se/1.5.0/docs/api/java/util/regex/Pattern.html

[4] Verb participle is denoted by the VBN-tag.

participle, the word sequence after the participle. Then, a previously constructed glossary is used to identify subjects and objects, as in [4]: The grammatical subject is the longest word sequence before the (auxiliary) verb, contained in the glossary. In the case of active segments, the object is the longest word sequence after the verb, contained in the glossary. In the case of passive segments, the object is the longest word sequence after the participle, contained in the glossary.

Propagation of subjects, objects, and verbs: Propagation of subjects, objects and verbs is necessary due to the fact that some sentence segments do not explicitly contain them but share with other segments. The propagation algorithm can be most simply illustrated on the tagged sentence

> “The_DT driver_NN accelerates_VBZ and_CC drives_VBZ faster_JJR than_IN 30km/h_CD and_CC less_JJR than_IN 50km/h_CD ._.”,

taken from the instrument cluster specification [3]. This sentence contains three elementary segments:

1. The_DT driver_NN accelerates_VBZ
2. drives_VBZ faster_JJR than_IN 30km/h_CD
3. less_JJR than_IN 50km/h_CD

The first segment contains a subject (“driver”) and a verb (“accelerates”). The second segment does not contain a subject but contains a verb (“drives”). Thus, the second segment inherits the subject (“driver”) from the first one and results in the segment “driver drives faster than 30km/h”. The third segment, in turn, lacks both subject and verb. Thus, it inherits them from the modified second segment and turns into “driver drives less than 50km/h. In a similar way the objects can be propagated as well.

The segments without verb inherit the active/passive annotation together with the verb. When the verb propagation is completed, there are no segments annotated as “segment without verb” any more. This propagation algorithm can be easily generalized to the case where the first sentence segment lacks a verb and also to passive segments. (Generalization not presented here due to space limitations.)

Mapping of subjects and objects to message senders and receivers: When the grammatical subjects and objects have been extracted and the verbs have been propagated to the segments without verbs, it is easy to translate every segment to an MSC message:

active: Message sender equals to the grammatical subject, receiver equals to the object, message content is the word sequence between the verb and the receiver.

passive: Message sender equals to the grammatical object, receiver equals to the subject, message content is the word sequence between the verb participle and the sender.

If the message sender or receiver cannot be identified directly from the sentence segment, they identification is postponed. In this case they are identified by the means of the analysis of the message stack, as in [4].

5 Evaluation: Case Study

The approach presented in this paper was evaluated on the instrument cluster specification [3]. Although not used in an industrial development project, this specification was derived from real industrial documents. This specification was also intended to serve as the contract basis between the car manufacturer and the supplier of the instrument cluster. The glossary, necessary for the translation of scenarios to MSCs, was extracted in our previous work [7].

The case study presented in this paper considered the same set of use cases as the case study performed in our previous work [4]. The difference lies in the treatment of passive and compound sentences. In the case study in [4] passive and compound sentences were manually split and rewritten. Table 1 shows some examples of the performed changes: Sentences in bold font are the corrected versions of the corresponding sentences on the left hand side.

Table 1. Original scenario (left) and corrected scenario used in the previous work [4] (right)

Use Case: activation of the instrument cluster	*Use Case: activation of the instrument cluster*
The driver switches on the car (ignition key in position ignition on).	The driver switches on the car (ignition key in position ignition on).
The instrument cluster is turned on and stays active.	**The instrument cluster turns on.** **The instrument cluster stays active.**
After the trip the driver switches off the ignition.	After the trip the driver switches off the ignition.
The instrument cluster stays active for 30 seconds and then turns itself off.	**The instrument cluster stays active for 30 seconds.** **The instrument cluster turns itself off.**
The driver leaves the car.	The driver leaves the car.

Table 2. Case Study: Statistics of Usage of Conjunctions and Passive

	Matching regular expression	Number of matching sentences
conjunction, and/or	`" [^ ]*_CC "`	96
conjunction, comma	`" [^ ]*_, "`	34
conjunction, colon	`" [^ ]*_:[ ]*"`	23
passive with verb “are”	`".* are_.*VBN.*"`	17
passive with verb “is”	`".* is_.*VBN.*"`	52
other passive forms	modifications of the above expressions	0

The case study consisted of 42 scenarios, containing on the total 384 sentences. Out of these 42 scenarios, only 37 were translated to MSCs in [4]. For the remaining 5 scenarios the necessary rewriting was too extensive. In the original (not rewritten) scenarios, a significant number of sentences contained either passive or conjunctions (cf. Table 2). These sentences were manually rewritten in [4], but treated without any changes in the case study presented here. The matchings listed in Table 2 are not disjoint, i.e., there are sentences matching several regular expressions and correspondingly counted

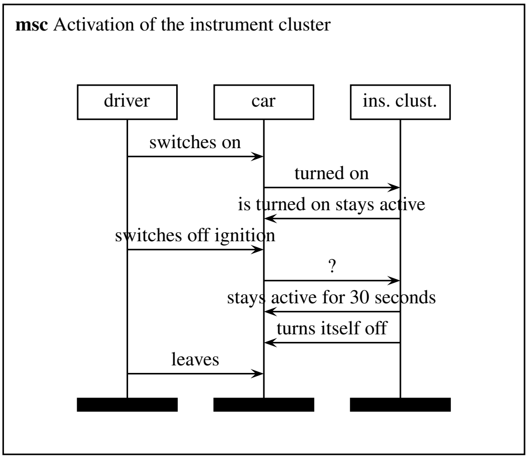

msc Activation of the instrument cluster
driver
car
ins. clust.
switches on
turned on
is turned on stays active
switches off ignition
?
stays active for 30 seconds
turns itself off
leaves

Fig. 2. MSC for the scenario "Activation of the instrument cluster", extracted from the original version of the scenario (left hand part of Table 1)

in several lines of Table 2. For example, the sentence "The instrument cluster is turned on and stays active" matches both the passive regular expression with the verb "is" and the regular expression for conjunctions with and/or.

To evaluate the correctness of MSCs extracted with the approach presented in this paper, these MSCs were manually compared with the MSCs extracted in [4]. Examples of the extraction results are shown in Figures 2 and 3: Figure 2 shows the MSC extracted from the original scenario (left hand part of Table 1), whereas Figure 3 shows the MSC extracted in [4] from the corrected scenario (right hand part of Table 1). These MSCs are obviously different. This difference results from two facts:

- In Figure 3 two missing messages (marked with "?") are identified, in Figure 2 the first "?"-message is not necessary any more because there is the explicit "turned on" message from the car to the instrument cluster[5]. This difference in MSCs is desirable, as the MSC in Figure 2 better identifies the message flow and makes less guessing of missing messages necessary.
- The MSC in Figure 2 contains a message "is turned on stays active" instead of "stays active". This message name is caused by a tagger error: "stays" is tagged as a noun, thus, the sentence segment "stays active" is considered as a sentence sequence without a verb and inherits its verb from the first sentence segment. Errors of this type can be corrected manually only by the requirements analyst.

Manual comparison of the MSCs extracted with automatic treatment of passive and conjunctions and the MSC extracted in [4] showed that they coincide in 26 cases out of 37 constructed in [4], modulo differences like between Figures 2 and 3. In 11 cases they were different due to the following effect: if all the segments of some sentence

[5] It is easy to use a stemmer, for example the Porter stemmer [8] to convert "turned on" to "turn on". In the presented work this issue was neglected because the grammatical form of the messages is irrelevant for the management of the message stack.

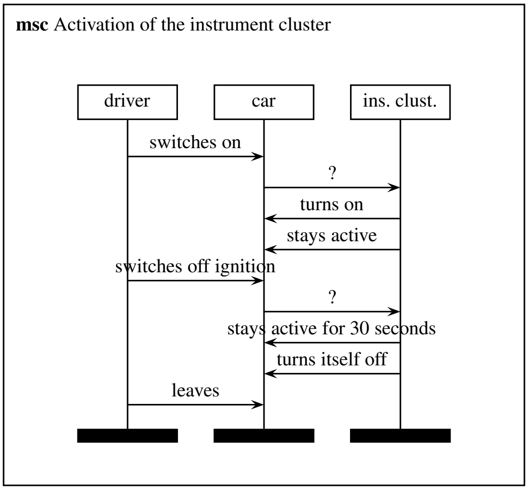


Fig. 3. MSC for the scenario "Activation of the instrument cluster", extracted from the corrected version of the scenario (right hand part of Table 1) in [4]

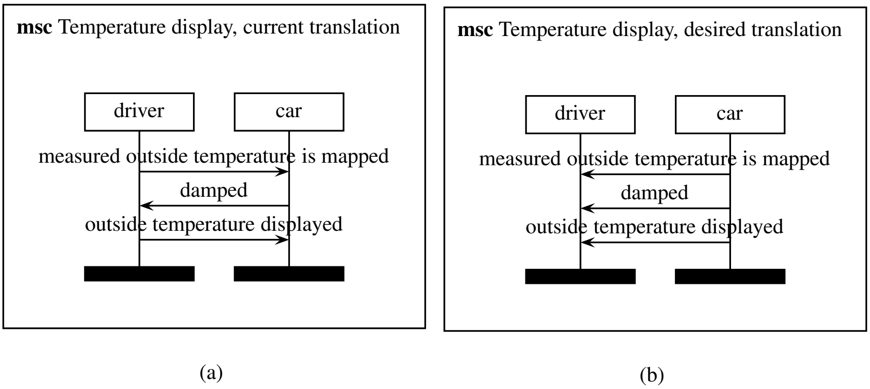
are passive, i.e., no message sender can be identified, the algorithm managing the message stack translates these segments to an interleaving sequence of requests and replies, instead of a sequence of messages going in the same direction. For example, the sentence "The measured outside temperature is mapped, damped and outside temperature displayed" is translated to the message sequence in Figure 4(a), whereas the desired translation is shown in Figure 4(b). Manual analysis of the program outputs showed that passive and conjunctions were treated correctly even for such sentences, but the stack management algorithm cannot take sequences of passive sentence segments into account yet. This was not necessary in [4] because the approach in [4] considers active sentences only.

6 Related Work

The idea to use computational linguistic to analyze requirements documents is surely not new. There was a lot of work in this area in recent years. There are three areas where natural language processing is applied to requirements engineering: assessment of document quality, identification and classification of application specific concepts, and analysis of system behavior.

Approaches to the analysis of document quality were introduced, for example, by Rupp [2], Fabbrini et al. [9], and Kamsties et al. [10]. All these approaches have in common that they define guidelines for document writing and measure document quality by analyzing the degree to which the document satisfies the guidelines. These approaches are barely comparable to the approach presented in this paper, as they do not perform any behavior analysis.

Other class of approaches, like for example those by Goldin and Berry [11] and Abbott [12], analyze the requirements documents, extract application specific concepts,


msc Temperature display, current translation
driver
car
measured outside temperature is mapped
damped
outside temperature displayed
msc Temperature display, desired translation
driver
car
measured outside temperature is mapped
damped
outside temperature displayed

(a)

(b)

Fig. 4. MSCs for the sentence "The measured outside temperature is mapped, damped and outside temperature displayed", translation with the current stack management algorithm (left) and desired translation (right)

and provide an initial model of the application domain. They do not perform any behavior analysis, either.

The approaches analyzing system behavior, as for example those by Ambriola and Gervasi [13], Rolland and Ben Achour [14], Díaz et al. [15], and Vadera and Meziane [16], translate the text to executable models by analyzing linguistic patterns. The approach presented in this paper differs from the approaches by Ambriola and Gervasi and by Vadera and Meziane in one extremely important feature: these approaches analyze only the information directly available in the document and do not reconstruct missing objects and actions. The approach by Rolland and Ben Achour defines rules for manual translation of sentences to messages, but does not perform any automatic analysis. Díaz et al. introduce a transformation technique producing UML sequence diagrams. However, the input to this transformation technique is semantical representation of the sentences and not plain text as in the presented paper.

To summarize, to the best of our knowledge, there is no approach to requirements documents analysis, able to identify missing pieces of behavior, especially by analyzing passive sentences and integrating them in a message stack, yet.

7 Conclusion

Requirements Engineering is a non-trivial task and the presented approach does not claim to solve all its problems. However, it solves several important problems of the early requirements analysis phase:

- It detects missing information in scenarios by guessing message senders/receivers in passive sentences.
- Compared to [4], it makes rewriting of passive and compound sentences unnecessary.
- It translates textual scenarios to MSCs, allowing for further validation.

When validated, the constructed MSCs can be used in further software development. Thus, the approach presented in this paper makes a contribution to behavior modeling. As shown in a case study, the approach is applicable to industrial documents.

References

1. Mich, L., Franch, M., Novi Inverardi, P.: Market research on requirements analysis using linguistic tools. Requirements Engineering 9, 40–56 (2004)
2. Rupp, C.: Requirements-Engineering und -Management. Professionelle, iterative Anforderungsanalyse für die Praxis. 2nd edn. Hanser–Verlag, ISBN 3-446-21960-9 (2002)
3. Buhr, K., Heumesser, N., Houdek, F., Omasreiter, H., Rothermehl, F., Tavakoli, R., Zink, T.: DaimlerChrysler demonstrator: System specification instrument cluster (2004), (accessed 11.01.2007), http://www.empress-itea.org/deliverables/D5.1_Appendix_B_v1.0_Public_Version.pdf
4. Kof, L.: Scenarios: Identifying missing objects and actions by means of computational linguistics, Contribution to the 15th IEEE International Requirements Engineering Conference (2007)
5. Grosz, B.J., Joshi, A.K., Weinstein, S.: Centering: A framework for modeling the local coherence of discourse. Computational Linguistics 21, 203–225 (1995)
6. Ratnaparkhi, A.: A maximum entropy model for part-of-speech tagging. In: Brill, E., Church, K., eds.: Proceedings of the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Somerset, New Jersey, pp. 133–142 (1996)
7. Kof, L.: Text Analysis for Requirements Engineering. PhD thesis, Technische Universitaet Muenchen (2005)
8. Porter, M.: An algorithm for suffix stripping. Program 14, 130–137 (1980) (accessed 14.07.2003), http://www.tartarus.org/~martin/PorterStemmer/
9. Fabbrini, F., Fusani, M., Gnesi, S., Lami, G.: The linguistic approach to the natural language requirements quality: benefit of the use of an automatic tool. In: 26th Annual NASA Goddard Software Engineering Workshop, pp. 97–105. IEEE Computer Society Press, Los Alamitos (2001)
10. Kamsties, E., Berry, D.M., Paech, B.: Detecting ambiguities in requirements documents using inspections. In: Workshop on Inspections in Software Engineering, Paris, France, pp. 68–80 (2001)
11. Goldin, L., Berry, D.M.: AbstFinder, a prototype natural language text abstraction finder for use in requirements elicitation. Automated Software Eng. 4, 375–412 (1997)
12. Abbott, R.J.: Program design by informal English descriptions. Communications of the ACM 26, 882–894 (1983)
13. Ambriola, V., Gervasi, V.: The Circe approach to the systematic analysis of NL requirements. Technical Report TR-03-05, University of Pisa, Dipartimento di Informatica (2003)
14. Rolland, C., Ben Achour, C.: Guiding the construction of textual use case specifications. Data. & Knowledge Engineering Journal 25, 125–160 (1998)
15. Díaz, I., Pastor, O., Matteo, A.: Modeling interactions using role-driven patterns. In: RE'05: Proceedings of the 13th IEEE International Conference on Requirements Engineering (RE'05), Washington, DC, USA, IEEE Computer Society, pp. 209–220. IEEE Computer Society Press, Los Alamitos (2005)
16. Vadera, S., Meziane, F.: From English to formal specifications. The. Computer Journal 37, 753–763 (1994)

Natural Language Processing and the Conceptual Model Self-organizing Map

Ricardas Laukaitis and Algirdas Laukaitis

Vilnius Management Academy
J.Basanaviciaus g. 29a, LT-2009,Vilnius, Lithuania
{laukaitis,alaukaitis}@vva.lt

Abstract. Self-organizing map can be an effective tool for the textual data classification. In this paper, we represent the methodology of an integration of the information system modeling and the development of the information system natural language interface. The main idea of the paper is to build the set of self-organising maps from information system documentation and then reuse it in human-machine communication as a semantic parsing component. The IBM's Information Framework (IFW) Financial Services Data Model has been used in an experiment where we tested how appropriate is presented methodology and what is classification accuracy of the received self-organizing maps. We compare classification accuracy with the IBM's WebSphere Voice Server NLU solution and demonstrate that self-organising maps can be a competitive components in the information systems natural language interfaces.

Keywords: Conceptual modeling, natural language processing, self-organizing maps, information systems natural language interfaces.

1 Introduction

Substantial part of the artificial intelligent research has been committed for the semantic parsing i.e. natural language sentences transformation to the formal language sentences. Nevertheless, natural language processing (NLP) has difficulties in finding its way into information systems development. For the proof of those statements we can look at the history of the development of the natural language database interfaces (NLDBI) (see [1] for the field review and [15] on what is state of the art in the field).

There are many reasons for the limited achievements of the NLDBI attempts. In this paper we attribute the two uppermost problems that we met when we tried to implement NLDBI for the database that has more than 100 tables. We used Microsoft English Query [15] due to its reliability and availability. Microsoft English Query has worked satisfactory when the number of entities were small (about 20 entities). But the work, ones needs to do by paraphrasing entities relationships when the number of entities increase beyond 20 seems quit substantial and then performance of the system deteriorated significantly.

Z. Kedad et al.(Eds.): NLDB 2007, LNCS 4592, pp. 193–203, 2007.

The problem is that symbol processing or rules based approaches require to much efforts for manually turning parameters when we are dealing with the hundreds of entities in the domain. On the other hand, over the past decade there have been developed a number of systems that map natural language sentences to the formal language sentences that used corpora based machine learning approaches [17]. Such factors like big manually classified corpora (Reuters-21578 text collection for example) and availability of natural language parsers for English language has boosted corpora based machine learning in the area of the NLP research as well. Then, in this paper, we state the following problems: *1. How can we combine two paradigms: symbol processing and corpora based to achieve better semantic parsing results?*

The second problem that we attribute from our experience with the Microsoft English Query product is that NLDBI is developed as a separate system or component when a business system is already build in place. This means that all the natural language based knowledge used to build the business system must be rediscovered and coded separately in the stage of NLDBI development. Then we state the second problem in this paper: *2. How can we preprocess information system textual documentation so that latter reuse it the NLDBI development?*

To work out on those two problems, the set of self-organizing maps (SOM) [10], [11] is proposed as a tool to analyze the documents and communication utterance. We suggest to use the set of maps where each map is associated with some subpart of conceptual model domain. In the paper [13] has been shown that the usage of one SOM can be useful but when the number of concepts increases the usage of the map becomes meaningless as well as the usage of other available tools. In this paper we demonstrate that the set of neural networks where each network specialises for the identification of the small number of concepts can improve the concepts identification accuracy.

The rest of the paper is organised as follows. First, we present the general framework of automated SOM generation from the information system documentation and engineers utterance. Next, we present the solution of the natural language processing (NLP) system which has been build from the open source, state-of-the-art NLP components. The idea of the conceptual model vector space and SOM associated with it is introduced and explained. Finally to prove the soundness of the proposed method we provide a numerical experiment in which the ability of the system to identify concepts from users utterance is tested. The IBM Voice Toolkit for WebSphere [8] (approach based on statistical machine learning) solution is compared with solution suggested in this paper.

2 General Framework

If we analyse the process of developing an information system then we can notice that most information analysts enquire by means of natural language. Such information is then used to get the better understanding between software engineers in producing formal statements about information system. If the natural language interface is of interest then it is produced as the separate product without referencing to software development documentation used to develop the system.

In the Figure 1 the Use Case diagram describes suggested system's behaviour from a software engineer viewpoint. Use Cases are shown where requirements for the natural language interface are concerned at the initial stage of business information system modeling and design.

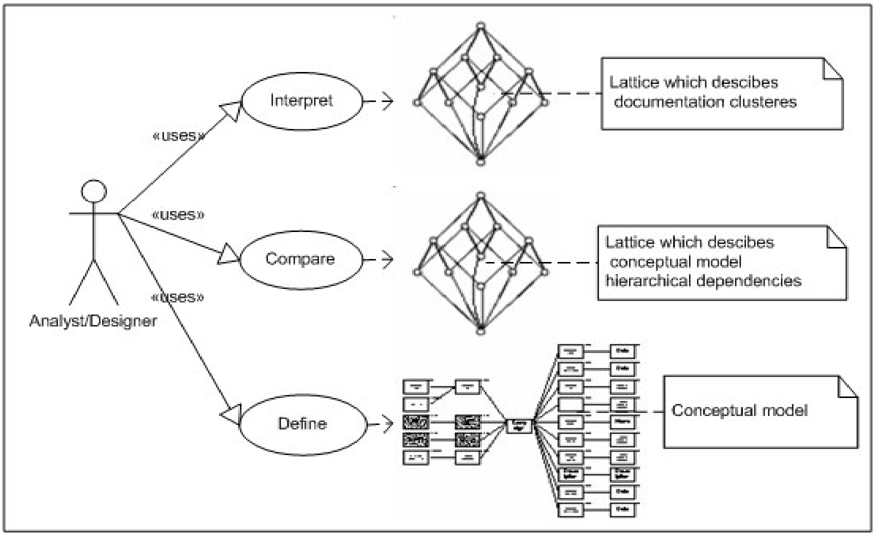

Interpret
«uses»
Lattice which descibes documentation clusteres
«uses»
Compare
Lattice which descibes conceptual model hierarchical dependencies
«uses»
Analyst/Designer
Define
Conceptual model

Fig. 1. Use Cases from software engineer viewpoint

There are two concept lattices one of which is received from information system documentation and another one from conceptual model themselves. Lattices are produced with Formal Concept Analysis (FCA) [3] and details on how they are produced from model and its documentation are described in [13]. The main idea is that by having two homogeneous structures we can control how good information system documentation is and interactively change documentation and see effect on the lattice structure.

The objects Collaboration diagram of the suggested system is presented in Figure 2. In the objects Collaboration diagram the document storage represents all information system documents. In our experiment it is the documents that describes the meaning of the concepts from the conceptual model.

We assume that there is a set of knowledge bases in the form of ontology. NLP Engine produces vector spaces based on ontology and documents corpus. Functionality of the NLP Engine is discused in section 3. The constraint that we put on the document base is the requirement for each document to have the concept name associated with it. Then, for example document:

> *An Employee is an Individual who is currently, potentially or previously employed by an Organization, commonly the Financial Institution itself. For example, Employee 123 fills a teller Employment Position ...*

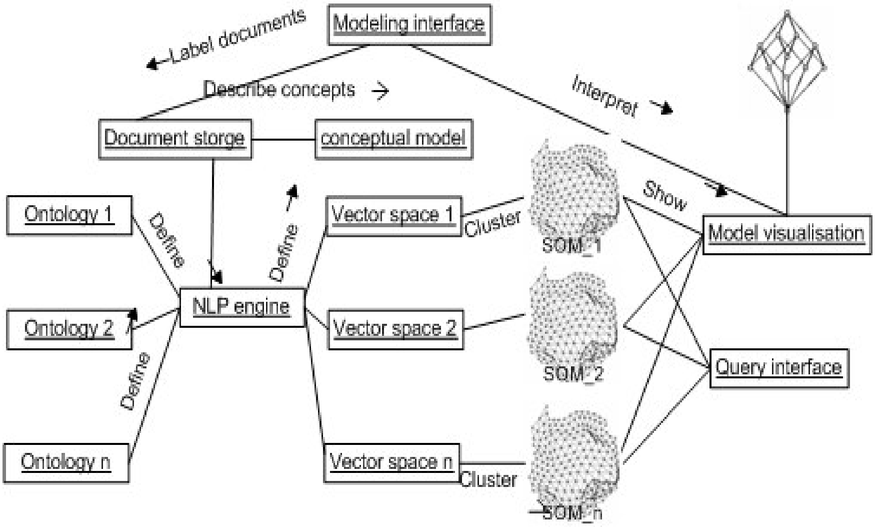


Fig. 2. Process of integration: Conceptual modeling, textual descriptions clusters detection and interpretation by use of formal concept analysis

will have concept name *Employee* that associates this document with the conceptual model and will be represented in the vectorial form by NLP engine as follows: "{... 0.45 0 0.145 0 0 ...}". Then, the self-organizing network [10] is build and used in the document base cluster analysis. Next, with conceptual context build from documents self-organizing map the documents concept lattice [3] is generated (section 4 for details). In parallel to this hierarchical clustering process, the conceptual context and concept lattice are build directly from the conceptual model. By constructing those two concept lattice (from model documentation and from the model itself) we can formally define what is model consistent documentation.

Definition. *We say that IS model documentation is consistent and well defined if concept lattice generated from documentations has the same structure as the concept lattice generated from model itself.*

There is a great amount of work done on how to compute the similarity between hierarchical structures. In suggested framework the similarity is measured by analyst who compare these lattice using Galicia software [19]. Recently an interesting work was presented by Maedche and Staab in which ontologies are compared along different levels: semiotic, syntactic and pragmatic [14]. But more research must be done to adopt those lines of thinking for the framework suggested in this paper.

The motivation to use the self-organizing maps for IS model generation and documentation verification follows from the fact that the self-organizing maps has been extensively studied in the field of textual analysis. Such projects like WEBSOM [9], [12] have shown that the self-organizing map algorithm can organize very large text collections and that SOM is suitable for visualization and intuitive exploration of the documents collection. The experiments with the Reuters corpus (a popular benchmark for text classification) have been investigated in the paper [6] and there were presented evidence that SOM can outperform other alternatives.

On the other hand, SOM gives cluster structures projected on the 2 or 3 dimensional surface and we need a technique that builds hierarchies from these clusters. Those arguments motivates integration of the formal concept analysis and other text clustering techniques.

3 Knowledge Base Vector Space Representation

English language is the one language where the most computational linguistic research has been done. Before beginning our research project we looked what products are available for English language parsing and semantic annotation. To our surprise the only product which was free, reliable and open source was GATE [2] NLP tool developed at University of Sheffield. In this section we describe what techniques from this product we used in our system. Additionally we explain how we integrate modules from GATE with our solution to produce vector space of the knowledge base.

The vector space model for text transformation to the vectors is a known conceptualization that transforms a document to a weight vector. Then received vector can be used in various tasks of textual information analysis. The method is based on the bag-of-words approach, which ignores the ordering of words within the sentence and use basic words occurrence information [18]. But recently this naive methods is used quit rarely and most researches produce some additional transformations to reduce either the dimensionality or to achieve better representation for the task at hand.

Dimensionality reduction is an important issuer. In our experiment we used the corpus with more than 3000 unique words and as the experiment in the section **??** demonstrates without the adequate dimension classification accuracy is too low for any reasonable use.

The processes of NLP that we used in this research for dimensionality reduction and knowledge base SOM production is depicted in Figure 3. Additionally at the right side of the Figure 3 we present the main idea on how we used information systems self organising map in natural language interface.

3.1 Learning

Document storage and *Conceptual model* are the same objects as in the Figure 2. From them we extract *concept triplet*: concept name, the concept parent name, and document that relates to the concept.

The Unicode tokeniser splits the text into simple tokens and is used for the next steps of the NLP. Additionally at this step we annotate all word by using WordNet dictionary [16] with the labels that show if the word have the noun or verb meanings. It has been done due to the fact that in some cases the

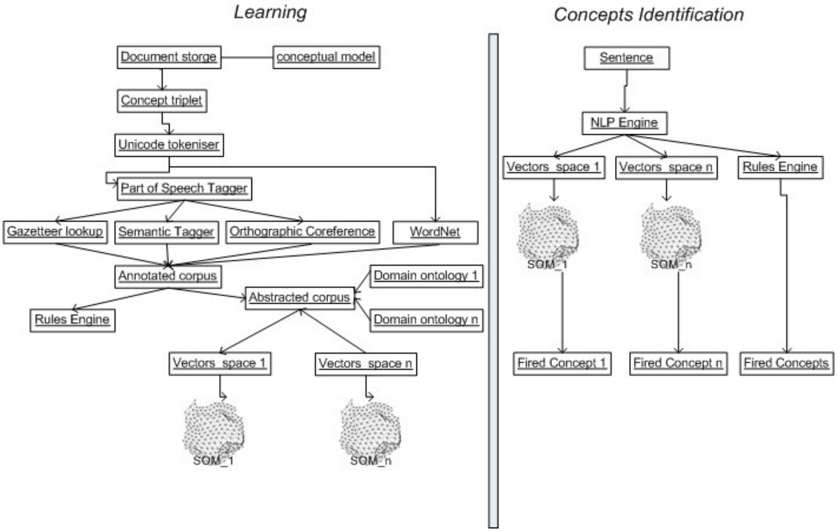


Fig. 3. NLP processes for conceptual model self-organizing map design

GATE system that we mentioned above produces wrong POS labeling and we detect those situations by comparing labels for consistency with the WordNet dictionary. This step indicated that more part of speech taggers we have the more accurate POS anotation can be done.

The part of speech tagger is a modified version of the Brill tagger, which produces a tags as an annotation on each word or symbol. In presented framework the tagger is used to extract nouns and verbs and remove all other words from the documents. This tagger is part from the GATE system and as we mentioned we additionally check its information with the WordNet dictionary.

The gazetteer reduces dimensionality of the documents corpus prior to classification. It uses the lists of named entities and annotates text with class labels such as cities, organisations, days of the week, etc. In presented framework each named entity is replaced by the name of the class. The information from the Gazetteer module is used directly in natural language interfaces as well. The classification results from this module has the highest priority in natural language interface concepts identification task. But as we mentioned above it surfers form the same problem as Microsoft English Query [15] product i.e. we are not able to produce symbolic based robust solution for big scale NLP task with the limited amount of human annotators resources.

Semantic tagger provides finite state transduction over annotations based on regular expressions. It produced additional set of named entities and we replaced each named entity with the class label.

Orthographic Coreference module adds identity relations between named entities found by the semantic tagger. Reduction of the state space dimensionality is achieved by replacing marked tokens with named entities class labels found by the semantic tagger.

Abstraction. The basic idea of the abstraction process is to replace the terms by more abstract concepts as defined in a given thesaurus, in order to capture similarities at various levels of generalization. For this purpose we used WordNet [16] and annotated GATE corpus as the background knowledge base. WordNet consists of so-called synsets, together with a hypernym/hyponym hierarchy [5]. To modify the word vector representations, all nouns have been replaced by WordNet corresponding concept ('synset'). Some words have several semantic classes ('synsets') and in that case we used a disambiguation method provided by WordNet - the 'most common' meaning for a word in English was our choice.

TFIDF Vectors space. In our experiments we used vector space of the terms vectors weighted by *tfidf* (term frequency inverse document frequency)[18], which is defined as follows:

$$tfidf(c,t) = tf(c,t) \times \log \frac{|C|}{|C_t|}.$$

where $tf(c,t)$ is the frequency of term t in concept description c, and C is total number of terms and C_t is the number of concepts descriptions containing this term. $tfidf(c,t)$ weighs the frequency of a term in a concept description with a factor that discounts its importance when it appears in almost all concepts descriptions.

3.2 Concepts Identification

The right side of the Figure 3 shows what steps are produced after we taught SOM neural networks. *NLP Engine* represents all those steps we described above to produce vector space of document collections. Then the usage of the SOM neural network is straightforward. We produce the document or sentences representation vector and SOM fires one neuron which is associated with some particular concept.

In parallel as we already mentioned we produce semantic annotations directly from the GATE system. If adequate annotation found then it is given the highest priority in decision making process that is driven by the *Rules engine*.

4 Self-organizing Maps of the Conceptual Model

Neurally inspired systems also known as connectionist approach supplement the use of symbols in problem solving by using simple arithmetic units through the process of adaptation. The winner-take-all algorithms also known as self-organizing network [10],[11] selects the single neuron in a layer of neurons that responds most strongly to the input pattern.

That feature is very attractive due to its simplicity. Every time input space can activate only one neuron. Most researches in the area of NLP pointed that this future can bring its own limitations due to the fact that quite often the documents can be labeled by several classes. As has been pointed by [6] the solution to deal with this problem can be by adding additional labels that codes two or more classes together. For example if we ask the system: "Show me recent history of company XXX credit rate?" - then, we will have activate four from nine the most abstract concepts from our conceptual model: *Involved Party, Arrangement, Event, Classification.* Now if we try to enter class that joints those two concepts and if we remember that in our conceptual model there is more than 1000 Concepts/Entities then we can see billions of classes. In such circumstances no flat structure will be reasonable to code all those classes.

Those arguments suggest the use of several smaller neural networks instead of the big one. Each neural network can take a small part of conceptual model and code its concepts. Some concepts can be used in several networks. If in one network we will use more abstract concepts then we can have hierarchically arranged.

In suggested architecture each self-organising map consist of a regular grid of neurons. Each neuron i is represented by prototype vector i.e. concept, $m_i = [m_{i1}...m_{in}]$ where n is input vector dimension. Input units take the input in terms of a feature vector and propagate the input onto the output neurons. The number of neurons and topological structure of the grid determines the accuracy and generalization capabilities of the SOM.

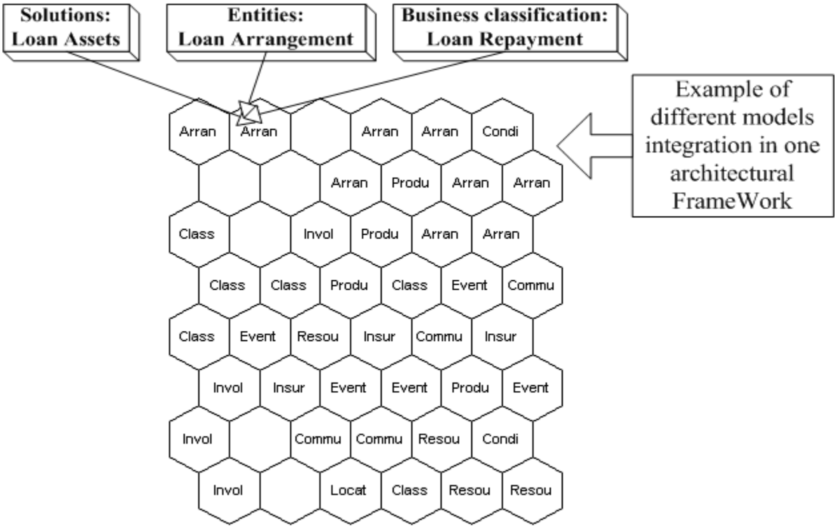


Fig. 4. The most general SOM for the conceptual model. Labels: invol, accou, locat, arran, event, produ, resou, condi represents concepts: involved party, accounting, location, event, product, resource, condition.

During learning the unit with the highest activation, i.e. the best matching unit, with respect to a randomly selected input vector is adapted in a way that it will exhibit even higher activation with respect to this input in future. Additionally, the units in the neighborhood of the best matching unit are also adapted to exhibit higher activation with respect to the given input.

We decided to test the framework presented in this paper on the one of the most successful conceptual models used by financial institutions - the IBM IFW financial services data model (FSDM) [7]. The model is divided into a number of levels with a different degree of abstraction: the 'A' level with nine data concepts that define the scope of the enterprise model (involved party, products, arrangement, event, location, resource items, condition, classification, business), the 'B' level with business concepts hierarchies (more than 3000 concepts), the 'A/B' level with business solutions (integrates business solutions with more than 6000 concepts) and 'C' level - entity relationship ER diagram with about 6000 entities, relationships and attributes.

From this model we extracted triplets: *concept, parent name and document which describes the meaning of the concept.* Each document has been transformed to the numerical representation as described in the previous section. As a result of training the self organizing map of the IBM IFW financial conceptual model has been obtained an it is shown in the Figure 4.

It has been expected that if the conceptual model vector space has some clusters that resembles conceptual model itself, then we can expect that the model will be easer understood compared with the model of more random structure. On a closer look at the map we can find regions containing semantically related concepts. For example, the right side top of the final map represents a cluster of concepts "Arrangement" and bottom right side "Resource items". Such map can be used as an interface to the underlying conceptual model. To obtain information from the collection of documents the users may formulate queries describing their information needs in terms of the features of the required concept.

5 Experiment

In the previous sections we have shown how to build hierarchical knowledge bases from IS documentation and how formally verify business information system conceptual model. As it was mentioned in the introduction, one of the objectives in this research project was to find the techniques and tools of IS modeling that brings an opportunity to reuse some components from modeling system in information systems natural language interfaces. In the presented framework such components are self-organizing maps. After the modeling stage the self-organizing map can be directly applied for concept identification from users utterance. The usage of SOM is simple: one of the maps neuron is firing when the new sentence is presented. As we have the set of concepts associated with each neuron then we can get the concept names for every sentence presented to the system.

To evaluate conceptual model SOM text classification performance we conducted the following experiment.

A group consisting of 9 students has been instructed about the database model. They queried the system with about 20 questions and tried to identify the "Involved Party" concept. After each experiment we increased the number of concepts that we put into the model. At the beginning only 9 top 'A' level concepts were considered. Next the number of concepts was increased to 50, 200, 400 and finally 500 concepts.

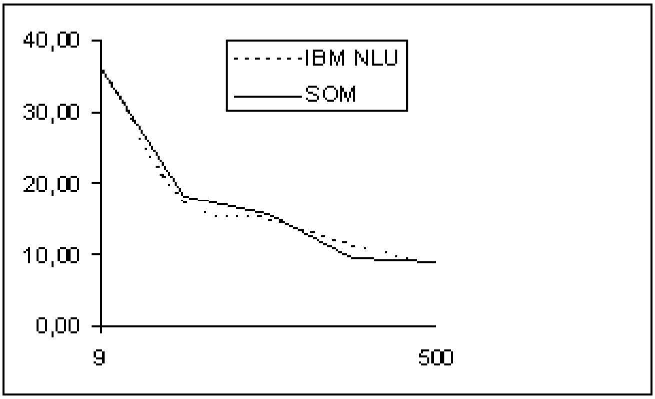


Fig. 5. Concept identification comparison between IBM NLU toolbox and conceptual model self-organizing map

As an alternative to the conceptual model self-organizing map we used the IBM WebSphere Voice Server NLU solution. We have taken the black box approach for both solutions: put the training data, compile and test the system response for new data set. The set of pairs {*textual description:concept name*} were constructed to train the IBM NLU model. The same set has been used to get business model self-organizing map. To detect the classification error the proportion of the correct identified concepts has been used. Figure 5 shows the results of the experiment. As we can see from the table SOM performance is similar to the IBM WebSphere Voice Server NLU solution.

6 Conclusion

Conceptual models and other forms of knowledge bases can be viewed as the products emerged from human natural language processing. The self-organization is the key property of humans mental activity and the present research investigated what self-organization properties can be found in the knowledge bases. We have shown that with the self-organizing map and formal concept analysis we can indicate inadequateness of the concept descriptions and improve the process of knowledge base development. Presented methodology can serve as the tool for maintaining and improving enterprise-wide knowledge bases. Additionally, we provided evidence that high quality documentation can be reused as the separate module in the IS natural language interfaces.

References

1. Androutsopoulos, I., Ritchie, G.D., Thanisch, P.: Time, Tense and Aspect in Natural Language Database Interfaces. Natural Language Engineering 4, 229–276 (1998)
2. Cunningham, H.: GATE: a General Architecture for Text Engineering. Computers and the Humanities 36, 223–254 (2002)

3. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer, Berlin (1999)
4. Hertzum, M., Pejtersen, A.M.: The information-seeking practices of engineers: searching for documents as well as for people. Journal of Information Processing and Management 36, 761–778 (2000)
5. Hofmann, T.: Probabilistic latent semantic indexing. In: Research and Development in Information Retrieval, pp. 50–57 (1999)
6. Hung, C., Wermter, S., Smith, P.: Hybrid Neural Document Clustering Using Guided Self-organisation and WordNet, pp. 68–77. IEEE Computer Society Press, Los Alamitos (2004)
7. IBM: IBM Banking Data Warehouse General Information Manual (accessed July 2006), Available from on the IBM corporate site http://www.ibm.com
8. IBM Voice Toolkit V5.1 for WebSphere Studio (accessed July 2006) http://www-306.ibm.com/software/
9. Kaski, S., Honkela, T., Lagus, K., Kohonen, T., WEBSOM,: self-organizing maps of document collections. Neurocomputing 21, 101–117 (1998)
10. Kohonen, T.: Self-organized formation of topologically correct feature maps. Biological Cybernetics 43, 59–69 (1982)
11. Kohonen, T.: Self-Organizing Maps. Springer, Heidelberg (2001)
12. Lagus, K., Honkela, T., Kaski, S., Kohonen, T., WEBSOM,: WEBSOM for textual datamining. Articial Intelligence Review 13(5/6), 345–364 (1999)
13. Laukaitis, A., Vasilecas, O.: Integrating all stages of software development by means of natural language processing. In: Proc. of International Working Conference on Requirements Engineering: Foundation for Software Quality (2007)
14. Maedche, A., Staab, S.: Measuring similarity between ontologies. In: Proceedings of the European Conference on Knowledge Engineering and Knowledge Management (EKAW), pp. 251–263 (2002)
15. Microsoft corporation. SQL Server and English Query (accessed January 2007) http://msdn.microsoft.com/
16. Miller, G.A.: WordNet: A Dictionary Browser. In: Proc. 1st Int'l Conf. Information in Data, pp. 25–28 (1985)
17. Mooney, R.J.: Learning for semantic parsing. In Proc. of Eighth International Conference on Intelligent Text Processing and Computational Linguistics (CICLing), pp. 311–324 (2007)
18. Salton, G.: Automatic Text Processing: The Transformation, Analysis and Retrieval of Information by Computer. Addison-Wesley, Reading (1989)
19. Valtchev, P., Grosser, D., Roume, C., Rouane, H.M.: GALICIA: an open platform for lattices. In: de Moor, A., Ganter, B. (ed.) Using Conceptual Structures: Contributions to 11th Intl. Conference on Conceptual Structures, pp. 241–254 (2003)

Automatic Issue Extraction from a Focused Dialogue

Koen V. Hindriks[1], Stijn Hoppenbrouwers[2], Catholijn M. Jonker[1],
and Dmytro Tykhonov[1]

[1] Man-Machine Interagtion Group, Delft University of Technology, Mekelweg 4,
2628 CD Delft, The Netherlands
[2] Faculty of Science, Radboud University Nijmegen, Toernooiveld 1, 6500 GL Nijmegen,
The Netherlands
{k.v.hindriks,c.m.jonker,d.tykhonov}@tudelft.nl, stijn@cs.ru.nl

Abstract. Various methodologies for structuring the process of domain modeling have been proposed, but there are few software tools that provide automatic support for the process of constructing a domain model. The problem is that it is hard to extract the relevant concepts from natural language texts since these typically include many irrelevant details that are hard to discern from relevant concepts. In this paper, we propose an alternative approach to extract domain models from natural language input. The idea is that more effective, automatic extraction is possible from a natural language text that is produced in a focused dialogue game. We present an application of this idea in the area of pre-negotiation, in combination with sophisticated parsing and transduction techniques for natural language and fairly simple pattern matching rules. Furthermore, a prototype is presented of a conversation-oriented experimentation environment for cooperative conceptualization. Several experiments have been performed to evaluate the approach and environment, and a technique for measuring the quality of extraction has been defined. The experiments indicate that even with a simple implementation of the proposed approach reasonably acceptable results can be obtained.

Keywords: natural language processing, domain modeling, grammar parsing.

1 Introduction

Domain models (including domain ontologies) are now a common asset created and used in many contexts, perhaps most prominently in Knowledge Engineering and Information System Development (two increasingly related disciplines). The groups involved in the research reported in this paper are concerned with domain modeling from different perspectives ranging from supporting system development to supporting negotiators. For the moment, the chief context to which we apply our ideas and setup is that of *conceptual modeling in small, communication-oriented, volatile domains*. The main characteristic of modeling in such domains is that it cannot be solidly based on existing data (corpus, documents, reference models) since the concepts involved reflect knowledge of only a small number of individuals, which in addition may crystallize only in the course of the interaction between those involved (consensus-based modeling). A typical example of such a context would be

Z. Kedad et al.(Eds.): NLDB 2007, LNCS 4592, pp. 204–216, 2007.

prenegotiation, a process that among others involves establishing a conceptual common ground on the basis of which negotiations can take place, and *specification* of information system requirements and models in fast evolving environments [12] .

Only limited research has yet been done concerning the *process* of domain modeling (for example [1], [2], [6], [16]), and only some of it has an experimental character. In order to study and, in the longer run, support and improve domain modeling in general, we believe it is important to create controlled environments that enable an experimental approach to modeling *processes* and *strategies*. We believe that such environments can evolve into actual modeling environments that take modeling beyond mere "ad hoc model creation" (graphical or otherwise). Such environments will take the shape of cooperative software tools that actively support the participants in the domain description process and allow them to discuss the target domain in a focused and structured manner, and, consecutively, can present them with a clear domain model they can then validate and refine.

The research presented here concerns the design, deployment, and evaluation of a prototype of a conversation-oriented experimentation environment for cooperative conceptualization. Our focus is on the detailed succession of expressive actions taken by people involved in a conversation for domain description/modeling, and (crucially) on the patterns, rationale, and strategies underlying such actions ([6]).

Our approach involves two key steps: *focused elicitation* of a domain description in the form of a structured natural language dialogue (captured in written textual form), and *automated extraction* of the core domain concepts from that dialogue. In our approach, we assume that a predefined meta-model is available and the aim of extraction is to populate this predefined meta-model. The meta-model for the experiment was designed by us and is presented in this paper.

We also present data and results from an experiment that has been carried out in order to *evaluate* the combined focused elicitation and automated extraction approach. As part of this evaluation, we use a manually constructed domain model as a benchmark (see section 5) and apply a metric to calculate the success rate of the automatic model extractor.

We believe our approach is promising for a number of reasons. If one takes a complex text or document, not specifically created to render core concepts, as a basis for automated domain analysis, then there are two main problems:

- The Natural Language Processing (NLP) involved (parsing, semantic analysis) is highly complex, very likely beyond the point of realistic application;
- Text analysis usually renders a large number of concepts with strongly varying degrees of relevance. Separating relevant concepts from irrelevant concepts is a daunting task that cannot be automated (not without substantial material to "learn about the domain from", that is).

So, if we cannot rely on high quality bulk input that can be effectively analyzed (indeed we assume we cannot), then instead we prefer to start with the creation of a *simple* text that is *purpose created* to contain core domain concepts and show that such texts can be *analyzed* using *simple, robust NLP techniques*. In order to obtain such natural language input, we use focused dialogue games. Formal dialogue games are interactions between two or more players, where each player acts by making utterances, according to a set of rules (cf. [14]). A dialogue game has a clear goal shared by the participants in the dialogue. As a consequence, it is reasonable to expect that the task of “filtering out” relevant concepts happens *as the text is created*, based on human intelligence in description/production rather than reading/interpretation afterwards. This “filtering” effect may be enhanced by structured/guided elicitation, i.e. by introducing additional rules in the dialogue setting that should be adhered to by the participants. This approach thus is based on an alternative *method* for domain modeling: a guided elicitation process, which aims at the production of focused texts including primarily relevant, core domain concepts in a structured environment, to which the automated, and therefore repeatable, extraction procedure is then applied.

An additional advantage of our approach lies in our use of a basic meta-model that requires minimal categorization effort on behalf of the extractor. This reduces the sensitivity to errors in the extraction process. The structure we use matches the basic structures in many comparable but more elaborate meta-models (ontological meta-models), suggesting that, for example, extending this approach to more negotiation-specific and complex meta-models (such as a negotiation description language [3], or to more generic widely-used ontology specification languages such as OWL [18], should not be too challenging. Later refinement of the domain model is possible if required (both of the meta-model and of the elicitation procedure).

In many applications, including prenegotiation, extraction of a domain model instance with relations exclusively between specific objects defined in the meta-model is required (bound variables). The main bulk of the domain independent knowledge can be pre-defined in the meta-model, by knowledge engineers. Thus, language constructions such as quantification or complex anaphoric references, which are particularly difficult in view of NLP, can be omitted in the *automated extraction* stage of our approach.

We propose a method to automatically extract a (partial) domain model from a focused dialogue of natural language. The effectiveness of the extraction method has been empirically validated by means of a series of experiments. The results of the experiments were validated against manually built models using a validation metric. The metric calculates the distance between the “ideal” model extracted manually by a human domain modeling expert and the atomically extracted model.

In the next section, we present our domain extraction model. Section 3 briefly introduces the NLP techniques used in the extraction tool. The extraction approach itself is introduced in Section 4. The results of the experiments with human dialogues are used to validate the extraction approach in Section 5. Our conclusions are presented in Section 6.

2 The Domain Extraction Approach

The extraction approach proposed here consists of two phases: (i) Focused Elicitation and (ii) Automated Extraction. The goal of the first phase is to organize collaboration of the domain experts on model elicitation with a specific focus on the domain: the natural language input for the domain extraction system should have a reasonable fit with the meta-model that is used. To ensure such a fit we propose to use variants of a *dialogue game*. The main advantage of dialogue games is that the users can be manipulated to keep their sentences simple.

The second phase automatically extracts a model from the elicited domain description in terms of a given domain meta-model. The method that is proposed here for extracting a domain model instance from natural language is a combination of *robust, wide coverage parsing techniques* and what we call *concept extraction rules*, which are used by a pattern matching algorithm to process the parser results. In two steps, the automatic domain extraction system transforms the natural language input into a domain model, an instance of the given meta-model. The effectiveness of this method relies on the assumption that the natural language utterances have a reasonable "fit" with a predefined, given meta-model. Effective concept extraction rules can then be derived from this meta-model and the output format of the parser.

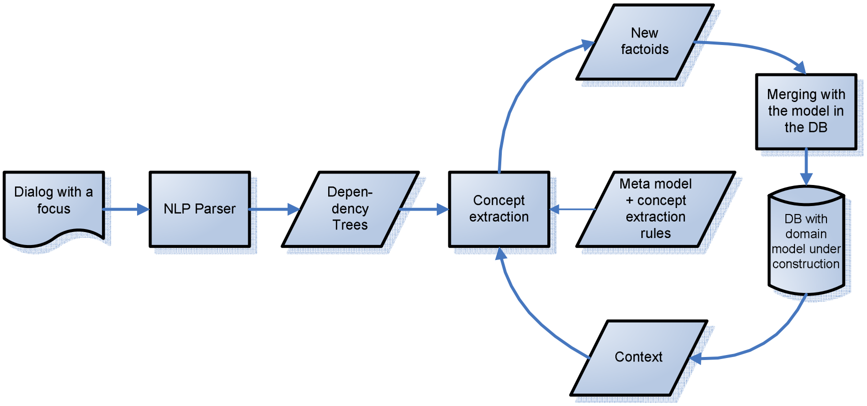

New factoids
Merging with the model in the DB
Dialog with a focus
NLP Parser
Depen-dency Trees
Concept extraction
Meta model + concept extraction rules
DB with domain model under construction
Context

Fig. 1. Method for Automatic Model Extraction

The method for automatic domain model extraction is shown in Figure 1. A transcript of a dialogue is provided as input to the system. A robust *dependency parser* is used to transform the utterances into so-called *dependency trees* (see below for an explanation). The dependency trees are input to a pattern matching module which is able to take the context of a tree (representing one or more factoids) into account, e.g. for resolving pronoun references. Finally, so-called *concept extraction rules* are used to extract a concrete instance of a domain model. These rules are fairly simple pattern matching rules derived from the generated parser output and the meta-model.

3 Dependency Trees and Dependency Triplets

All utterances are parsed using the EP4IR grammar of English [7], [8], normalized, transduced to dependency trees, and unnested to dependency triplets. By a dependency tree (DTree) we mean a graph (a tree with possibly some confluent arcs) whose nodes are marked with words and whose arcs are marked with certain syntactic relations. A dependency tree obtained from an utterance represents the most important syntactic relation in the utterance: SVOC (Subject/Verb/Object/Complement) trees and NP (Noun Phrase) trees. The SVOC trees correspond to the *factoids* (who is said to do what to whom under what circumstances) expressed by the utterance. The following dependency tree shows the typical structure of the attributed noun and of the SVOC-sentence.

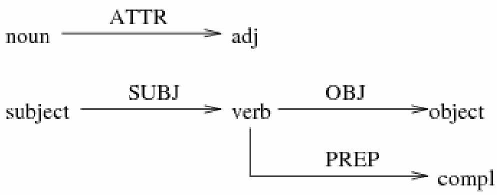


Fig. 2. Example of a dependency tree

By a dependency triple (DT) we mean a triple (word, relation, word), which forms part of a dependency tree, from which it can be obtained by *unnesting* the tree. DT's are the building-stones that constitute factoids. There is a long history of the use of DT's and the related *head/modifier pairs* [9] in Information Retrieval.

A dependency tree gives an abstract view of the structure of a sentence in terms of well defined syntactic word relations from which semantic relations can be derived relatively easily. A dependency tree is much more compact and abstract than a constituent tree (parse tree).

The parsing process takes into account the subcategorization frames of verbs, nouns and adjectives, as well as the verb valences. The words occurring in the DTs are lemmatized. The following table shows the most important dependency relations, together with their concrete notation as a DT and an example:

Table 1. Dependency relations

subject relation	[noun,SUBJ verb]	[picture,SUBJ show]
object relation	[verb,OBJ noun]	[show,OBJ view]
attrib relation	[noun,ATTR noun]	[theatre,ATTR movie]
attrib relation	[noun,ATTR adje]	[monument,ATTR large]
predicative relation	[noun,PRED noun]	[Louvre,PRED museum]
prepos relation	[noun,PREP noun]	[sword,IN hand]
prepos relation	[verb,PREP noun]	[sit,ON chair]
prepos relation	[adje,PREP noun]	[full,OF arrows]
modification	[adje,MOD advb]	[green,MOD intensely]
modification	[verb,MOD advb]	[cause,MOD not]
quantification	[noun,QUANT number]	[horse man,QUANT three]
determination	[noun,DET determiner]	[scene,DET whole]

As an example, the sentence 'the picture shows a view of Ravenna from the air' corresponds to the following dependency tree:

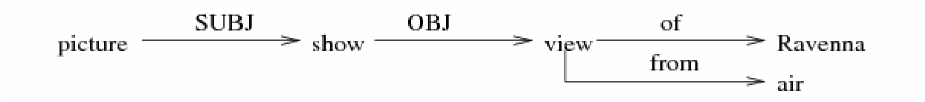


Fig. 3. Dependency tree for 'the picture shows a view of Ravenna from the air'

The example 'the picture shows a view of Ravenna taken from the air' is transduced to two (connected) Dependency Trees [10]:

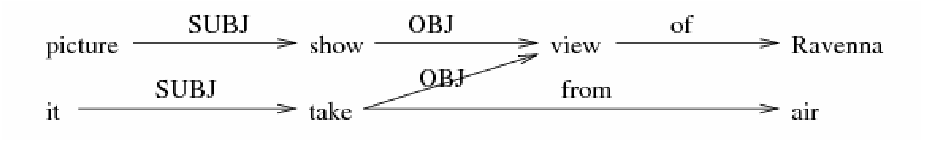


Fig. 4. Dependency tree for 'the picture shows a view of Ravenna taken from the air'

The subject 'it' in the second DTree is just a handle for anaphora resolution. During the transduction, extensive normalizations are performed in order to map equivalent phrases onto a common representative: variations in word order, time and modality are eliminated, questions and passive sentences are translated to active form (see [9], [10]). Finally, the words in the DT's are lemmatized. The EP4IR parser/transducer was developed for application in Information Retrieval [11]. Our paper shows that it can also be used successfully for Domain Modeling.

4 Extracting a Domain Model

In general, it will not be possible to match the dependency tree output of the parser one-on-one with a given meta-model. The natural language parser, however, does provide a well-structured and well-defined output that can be used in a final domain extraction phase. The key idea of this final phase is to match parts of a dependency tree with parts of the desired domain model.

The meta-model determines the structure of the desired domain model as well as that of the extraction rules that are used in the extraction phase. The meta-model consists of the key concepts that need to be extracted from the natural language text. Of course, the meta-model should have a reasonable fit with the natural language text. As discussed above, a reasonable fit can be obtained by using structured dialogue games to produce the text.

In the prenegotiation domain, which provides the running example of this paper, a meta-model of the domain of negotiation needs to be instantiated in order to fix the negotiation issues. As Raiffa discusses in [16], parties are advised to prepare a negotiation template in this prenegotiation phase. Such a template has a simple structure. It consists of a list of issues that need to be resolved, and, for each issue, an agreed-upon set of possible resolutions.

In a negotiation about multiple issues, the result of the domain extraction method should be an instance of the meta-model depicted in Fig. 5. Basically, objects and their properties need to be extracted from the dependency trees.

The rules for extracting domain elements have to capture those patterns present in a dependency tree that with a high probability indicate that the text is about an object or a property (or both). By inspection of the relations listed in Table 1, and dependency trees (cf. Fig. 3 and Fig. 4) that result from typical dialogue games, various patterns are readily suggested.

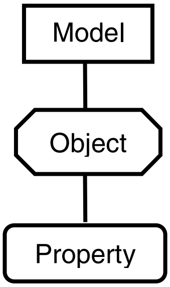


Fig. 5. Structure of the Negotiation Meta-Model

In the dialogue games that we have used in our experiments, typical patterns are, for example:

1. [pro: I, SUBJ, verb: have, OBJ, noun: *x*],
2. [noun: *x*, SUBJ, verb: have, OBJ, noun: *y*],
3. [noun: *x*, ATTR, adje: *y*].

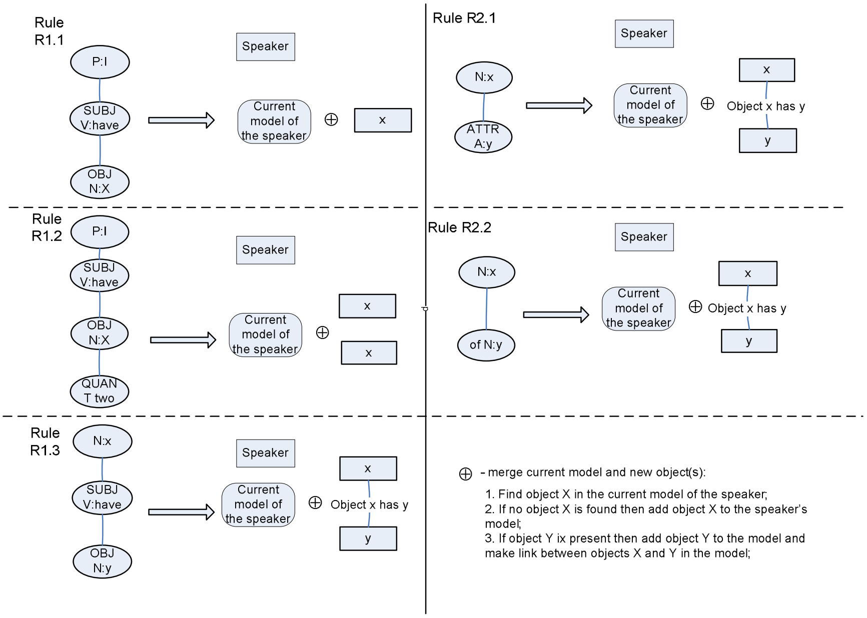
An instance of the first pattern is, for example, a sentence such as *I have a daisy*. It is clear that such a pattern requires the addition of the object named *daisy* to the domain model. The first pattern is also a sub-pattern of the slightly more complicated sentence *I probably have a daisy*, which is an instance of the pattern: [pro: I, SUBJ, verb: have, OBJ, noun: daisy, MOD, advb: probably]. Even though this sentence indicates that there is a chance the object is *not* a daisy, the pattern is processed by adding the object named *daisy* to the domain model.[1] An instance of the second pattern is e.g. *The cup has a handle.* Finally, an instance of the third pattern is *The cup is blue*. In the latter case, a property of being *blue* needs to be added to the model.

The conception extraction rules should map such patterns onto domain elements, where the domain structure is given by the meta-model. The basic structure of a conception extraction rule therefore is defined as:

<subpattern of dependency tree> → <update instruction(s) for domain model>.

The rules code instructions for extracting domain elements from a dependency tree in case the left-hand side of a rule matches with a sub pattern of the tree.

[1] Depending on the application area such rules can be changed to not allow this.



⊕ - merge current model and new object(s):

1. Find object X in the current model of the speaker;
2. If no object X is found then add object X to the speaker's model;
3. If object Y ix present then add object Y to the model and make link between objects X and Y in the model;

Fig. 6. Domain Extraction Rules

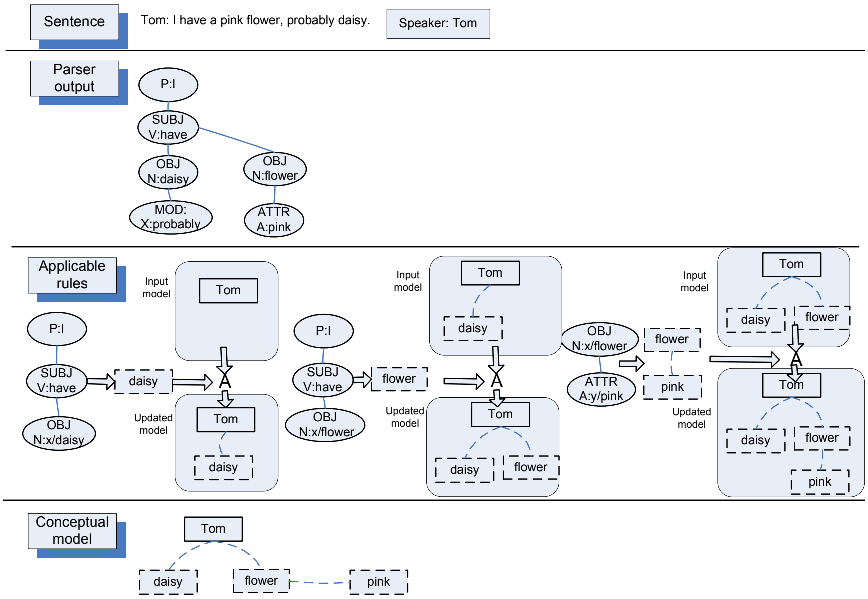


Fig. 7. Example of Domain Extraction with the Rules

The process of domain extraction can be summarized as follows (cf. also Fig. 5, and 6). The pattern matching module of the domain extraction system tries to match the left-hand side of each concept extraction rule. For each match, the resulting bindings of the matching process are retrieved and the instructions (properly instantiated) on the right-hand side of the rule are executed. These instructions consist of *adding a new node to the domain model*, *adding a property together with the related object to the domain model*, and *merging the extracted information with the domain model* (in case a property of an object needs to be added but the object is already present in the model). The primitive operations that are performed on a domain model are *add_node* and *add_edge* operations. The domain extraction module thus also performs merging of overlapping models that are extracted from different sentences of a single dialogue.

5 Experimental Validation

The proposed domain extraction method has been designed in order to facilitate humans in the construction of a domain model. For the running example, a specific meta-model was used to illustrate the extraction method. In order to validate the method proposed in the previous section, a series of experiments with human subjects was performed to measure the effectiveness of the method. For negotiation and its corresponding meta-model, a dialogue game is needed that results in a descriptive natural language text that is focused on the naming of objects and the identification of properties of these objects. Such a game can be viewed as a model of a domain modeling task in which a knowledge engineer and a domain expert are trying to construct a domain model.

In line with a general view on domain modeling as expressed in [5], the experiment was organized as a dialogue game (taking the form of a chatbox) played by two participants seated in different rooms, who were each presented with a set of pictures on a screen (some identical, some different). Figure 8 presents a screenshot of the chat

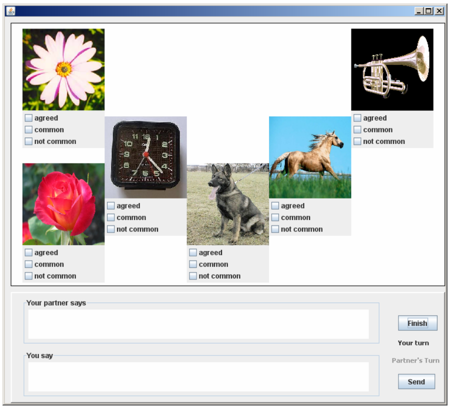


Fig. 8. Screenshot of the Chatbox Software used in the Experiment

box software used to organize the experiment. The participants were asked to discuss the objects displayed in the pictures (each participant could only see his/her own set of pictures). The participants were given the task to find out which of the objects are present on both sets of pictures (i.e., they had to identify the objects that are common, meaning that both participants see exactly the same pictures of those objects on their screens). This setup requires the participants to go through an elicitation phase as defined earlier. For the purpose of validation, the resulting dialogues were processed in two ways: by the automatic domain extraction tool and independently, by a knowledge engineer who manually created a domain model.

For manual domain modeling, the knowledge engineer was given a particular dialogue as a domain description, but the engineer had no access to the pictures that were presented to the participants in the dialogue. In this way, the engineer was limited to basing the domain model on the content of the dialogue. As a result, he added an object or its property to the model only if it was explicitly mentioned in the dialogue. For example, if a dialogue included a statement such as “Participant A: I have a pink flower” the knowledge engineer would add an object “flower” to the domain model and a property “pink” linked to the object “flower”. The domain model obtained in this way has been used as the standard (or “ideal”) domain model against which the results from automatic extraction were then compared.

To compare the ideal domain model and the automatically extracted model, the A* Algorithm for Error-Correcting Subgraph Isomorphism Detection [15] was used. Observe that domain models are graphs and thus can be provided as input to the algorithm. The algorithm calculates the similarity distance between two graphs and is based on the idea of compensating the distortions in one graph by means of edit operations that are applied to the second graph.

The edit operations include vertex deletion and insertion, edge deletion and insertion, and attributes and labels substitution. All edit operations have equal cost. The total cost of the transformation of the graph is the sum of the costs of each individual edit operation. The A* algorithm looks for a sequence of edit operations that would have the minimal total costs of the transformation.

The following formula determines the correctness of the extracted model:

$$c = \left(1 - \frac{d\left(g_{\mathrm{exp}ert}, g_{extracted}\right)}{d\left(g_{\mathrm{exp}ert}, \phi\right)}\right) \cdot 100\% \qquad (1)$$

where $d(g_{expert}, g_{extracted})$ is the distance between the domain model extracted by expert and the domain model automatically extracted by the tool, and

$d(g_{expert}, \varnothing)$ is the distance between the domain model extracted by the expert and the empty graph.

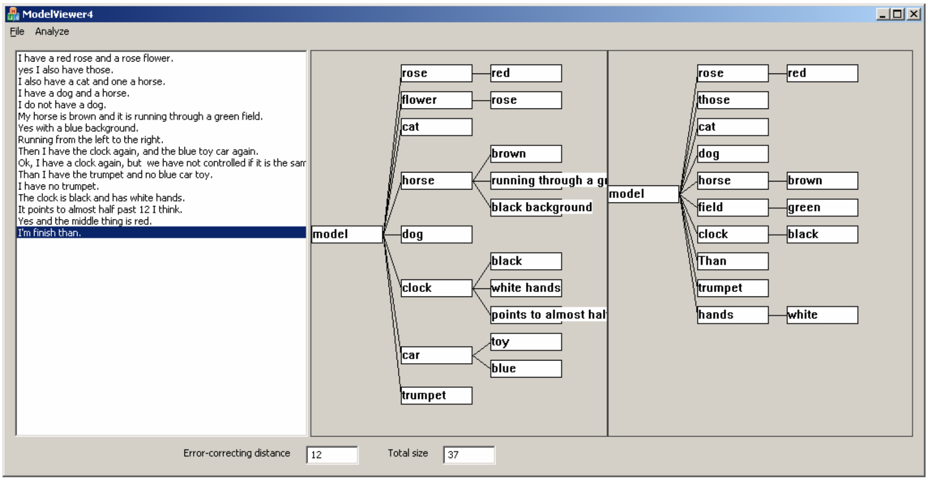
Table 2 presents the results of the validation of the series of experiment. Each of the eight pairs of the participants performed eight trials. Each trial has a set of six pictures. Two pictures out of six are common for the participants. We varied the sets of the pictures among the trials through the pairs of the participants to avoid any possible side-ways effects of the trials sequence.

Table 2. Experimental results – correctnes of the automaticaly extracted domain models

Experiment	Sets of pictures							
	1	2	3	4	5	6	7	8
Pair 1	40%	49%	46%	45%	51%	68%	69%	57%
Pair 2	50%	63%	58%	65%	77%	68%	67%	55%
Pair 3	68%	41%	43%	51%	68%	72%	49%	65%
Pair 4	56%	54%	41%	43%	63%	61%	65%	54%

The average percentage of the correctness of the extracted models is 57%.

The experimental results show that the precision of the model extraction still needs significant improvement. However, note that the models were extracted without any use of semantics. One way of improving the accuracy of the models is to use domain knowledge available, e.g., WordNet [4], CYC [13]. Accuracy might be improved by adding more rules to the dialogue game to structure the natural language produced. Another way is to make the modeling process interactive by presenting the updated instance of the domain model while the user continues his/her descriptions in natural language. Since the user immediately sees the interpretation of his words s/he can re-formulate his/her sentence if necessary.



I have a red rose and a rose flower.
yes I also have those.
I also have a cat and one a horse.
I have a dog and a horse.
I do not have a dog.
My horse is brown and it is running through a green field.
Yes with a blue background.
Running from the left to the right.
Then I have the clock again, and the blue toy car again.
Ok, I have a clock again, but we have not controlled if it is the sam
Than I have the trumpet and no blue car toy.
I have no trumpet.
The clock is black and has white hands.
It points to almost half past 12 I think.
Yes and the middle thing is red.
I'm finish than.

Error-correcting distance 12 Total size 37

Fig. 9. Screenshot of the automatic domain extraction tool (from left to right: source dialogue, manually extracted reference model, automatically extracted model)

6 Conclusions

This article presents an automatic domain model extraction method based on a predefined meta-model. Our method involves two basic steps: focused elicitation where domain experts describe the domain in a natural language dialogue and automated extraction based on an existing NLP parser and a set of pattern-matching rules to extract the basic concepts of the domain. The output of the proposed method has been validated against ideal models build manually by a domain expert using the dialogues received from the experimental setup.

Validation results show a big deviation in the accuracy of the domain model extraction. The accuracy metric varies from 40% to 77% throughout the experiments, generally in correspondence to the "neatness" (complexity) of the sentences produced by the participants. In future work a sentence complexity evaluation algorithm will be developed using the parser output to assess quality of the domain elicitation. The accuracy of the approach will be improved: by involving the domain experts in a more direct way and by presenting them continuously with the models extracted. This allows the human to directly correct the system if necessary. Furthermore, the humans will be asked to reformulate if the parser has difficulties with the sentences produced. Finally, advanced pattern matching rules will be used, that are based on semantic knowledge obtained from the Internet, a specialized database or existing ontologies.

Acknowledgement. The authors thank Kees Koster (Radboud University Nijmegen) for his help in the production of this paper and in particular for his effort in making the EP4IR parser available for the reported research.

References

1. Anthony, S., Batra, D., Santhanam, R.: The use of a knowledge-based system in conceptual data modeling. Decision Support Systems 41, 176–190 (2005)
2. Batra, D., Antony, S.: Consulting support during conceptual database design in the presence of redundancy in requirements specifications: an empirical study. IBM Journal of Research and Development 54, 25–51 (2006)
3. Elfatatry, A., Layzell, P.: A negotiation description language. Software—Practice & Experience 35(4), 323–343 (2005)
4. Fellbaum, C.: WordNet: an Eleectronic lexical database. MIT Press, Cambridge (1998)
5. Hoppenbrouwers, S.J.B.A., Proper, H.A., van der Weide, T.P.: A Fundamental View on the Process of Conceptual Modeling. In: Delcambre, L.M.L., Kop, C., Mayr, H.C., Mylopoulos, J., Pastor, Ó. (eds.) ER 2005. LNCS, vol. 3716, Springer, Heidelberg (2005)
6. Hoppenbrouwers, S.J.B.A., Proper, H.A., van der Weide, Th.P.: Towards explicit strategies for modeling. In: Halpin, T.A., Siau, K. and Krogstie, J. (eds.) Proc. of the Workshop on Evaluating Modeling Methods for Systems Analysis and Design (EMMSAD'05), p485-492. FEUP, Porto, Portugal, EU (2005) ISBN 9727520774
7. Koster, C.H.A.: Affix Grammars for Natural Languages. In: Alblas, H., Melichar, B. (eds.) Attribute Grammars, Applications and Systems. LNCS, vol. 545, pp. 469–484. Springer, Heidelberg (1991)
8. Koster, C.H.A., Verbruggen, E.: The AGFL Grammar Work Lab. In: Proc. FREENIX/ Usenix, pp. 13–18 (2002)
9. Koster, C.H.A.: Head/Modifier Frames for Information Retrieval. In: Gelbukh, A. (ed.) CICLing 2004. LNCS, vol. 2945, pp. 420–432. Springer, Heidelberg (2004)
10. Koster, C.H.A.: Transducing Text to Multiword Units, Workshop on MultiWord Units MEMURA at the fourth International Conference on Language Resources and Evaluation, LREC-2004. Lisbon, Portugal, p. 8 (2004)

11. Koster, C.H.A., Seibert, O., Seutter, M.: The PHASAR Search Engine. In: Kop, C., Fliedl, G., Mayr, H.C., Métais, E. (eds.) NLDB 2006. LNCS, vol. 3999, pp. 141–152. Springer, Heidelberg (2006)
12. Krogstie, J., Jorgensen, H.D.: Quality of Interactive Models. In: Spaccapietra, S., March, S.T., Kambayashi, Y. (eds.) ER 2002. LNCS, vol. 2503, pp. 351–363. Springer, Heidelberg (2002)
13. Lenat, D., Guha, R.V.: Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project. Addison-Wesley, Reading (1990)
14. McBurney, P., Parsons, S.: Dialogue Games in Multi-Agent Systems. Informal Logic. Special Issue on Applications of Argumentation in Computer Science 22(3), 257–274 (2002)
15. Messmer, B.T.: Efficient Graph Matching Algorithms for Preprocessed Model Graphs, PhD thesis, Institut für Informatik und angewandte Mathematik, Universität Bern, Switzerland (1995)
16. Raiffa, H.: Lecture Notes on Negotiation Analysis, Harward University, PON Books (1996)
17. Shoval, P., Danoch, R., Balaban, M.: Hierarchical ER Diagrams (HERD) - The Method and Experimental Evaluation. In: Olivé, À., Yoshikawa, M., Yu, E.S.K. (eds.) Advanced Conceptual Modeling Techniques. LNCS, vol. 2784, pp. 264–274. Springer, Heidelberg (2003)
18. Web Ontology Language (OWL), http://www.w3.org/2004/OWL/

Character N-Grams Translation in Cross-Language Information Retrieval

Jesús Vilares[1], Michael P. Oakes[2], and Manuel Vilares[3]

[1] Department of Computer Science, University of A Coruña
Campus de Elviña s/n, 15071 – A Coruña (Spain)
jvilares@udc.es

[2] School of Computing and Technology, University of Sunderland
St. Peter's Campus, St. Peter's Way, Sunderland – SR6 0DD (United Kingdom)
Michael.Oakes@sunderland.ac.uk

[3] Department of Computer Science, University of Vigo
Campus As Lagoas s/n, 32004 – Ourense (Spain)
vilares@uvigo.es

Abstract. This paper describes a new technique for the direct translation of character n-grams for use in Cross-Language Information Retrieval systems. This solution avoids the need for word normalization during indexing or translation, and it can also deal with out-of-vocabulary words. This knowledge-light approach does not rely on language-specific processing, and it can be used with languages of very different natures even when linguistic information and resources are scarce or unavailable. Our proposal also tries to achieve a higher speed during the n-gram alignment process with respect to previous approaches.

Keywords: Cross-Language Information Retrieval, character n-grams, translation algorithms, alignment algorithms, association measures.

1 Introduction

The interest in using character n-grams for text conflation in Information Retrieval (IR) comes from the possibilities they offer, particularly in the case of non-English languages [6,7]. Since it provides a surrogate means to normalize word forms and it does not rely on language-specific processing, it can be applied to very different languages, even when linguistic information and resources are scarce or unavailable.

Its use is quite simple, since both queries and documents are just tokenized into their compounding overlapping n-grams instead of words: the word `potato`, for example, is split into: `-pot-`, `-ota-`, `-tat-` and `-ato-`. The resulting n-grams are then processed by the retrieval engine.

Nevertheless, when extending its use to Cross-Language Information Retrieval (CLIR), an extra translation phase is needed. A simple solution consists of, firstly, using any of the standard machine translation methods used in CLIR for translating the query and, next, splitting the resulting query into n-grams [6].

Z. Kedad et al.(Eds.): NLDB 2007, LNCS 4592, pp. 217–228, 2007.

Our approach is based on the previous work of the Johns Hopkins University Applied Physics Lab (JHU/APL), which went one step further and proposed a direct n-gram translation algorithm which allowed translation not at the word level but at the n-gram level [7]. This solution avoids some of the limitations of classic dictionary-based translation methods, such as the need for word normalization or the inability to handle out-of-vocabulary words. Nevertheless, the initial proposal resulted to be very slow. For example, it could take several days in the case of working with 5-grams.

This paper describes a new proposal for direct n-gram translation we have developed and which tries to speed up the process in order to make the testing of new developments easier. The article is structured as follows. Firstly, Sect. 2 describes our system. Next, Sect. 3 evaluates our approach. Finally, Sect. 4 presents our conclusions and future work.

2 The Character *N*-Gram Alignment Algorithm

In contrast with the original system developed by JHU/APL, which relies mainly on ad-hoc resources, our system has been built using freely available resources when possible in order to minimize effort and to make it more transparent. This way, our system employes the open-source retrieval platform Terrier [1]. This decision was supported by the satisfactory results obtained with n-grams using different indexing engines [11]. The well-known Europarl parallel corpus [3] is also used. This corpus was extracted from the proceedings of the European Parliament, containing up to 28 million words per language. It includes versions in 11 European languages: Romance (French, Italian, Spanish, Portuguese), Germanic (English, Dutch, German, Danish, Swedish), Greek and Finnish.

Our n-gram alignment algorithm consists of two phases. In the first phase, the slowest one, the input parallel corpus is aligned at the word-level using the well-known statistical tool GIZA++ [9], obtaining as output the translation probabilities between the different source and target language words. Next, in the second phase, n-gram translation scores are computed employing statistical association measures [5]. Our approach increases the speed of the process by concentrating most of the complexity in the word-level alignment phase. This first step acts as a filter, since only those n-gram pairs corresponding to aligned words will be considered, whereas in the original JHU/APL approach all n-gram pairs corresponding to aligned paragraphs were considered.

2.1 Word-Level Alignment Using Association Measures

Our n-gram alignment algorithm is an extension of the way association measures can be used for creating bilingual word dictionaries taking as input parallel collections aligned at the paragraph level [12]. In this context, given a word pair $(word_s, word_t)$ —$word_s$ standing for the source language word, and $word_t$ for its candidate target language translation—, their cooccurrence frequency can be organized in a *contingency table* resulting from a cross-classification of their cooccurrences in the aligned corpus:

	$T = word_t$	$T \neq word_t$	
$S = word_s$	O_{11}	O_{12}	$= R_1$
$S \neq word_s$	O_{21}	O_{22}	$= R_2$
	$= C_1$	$= C_2$	$= N$

As shown, the first row accounts for those instances where the source language paragraph contains $word_s$, while the first column accounts for those instances where the target language paragraph contains $word_t$. The cell counts are called the *observed frequencies*: O_{11}, for example, stands for the number of aligned paragraphs where the source language paragraph contains $word_s$ and the target language paragraph contains $word_t$. The total number of word pairs considered —or *sample size* N— is the sum of the observed frequencies. The row totals, R_1 and R_2, and the column totals, C_1 and C_2, are also called *marginal frequencies*, and O_{11} is called the *joint frequency*.

Once the contingency table has been built, different association measures can be easily calculated for each word pair. The most promising pairs, those with the highest association measures, are stored in the bilingual dictionary.

2.2 Adaptations for *N*-Gram-Level Alignment

We have described how to compute and use association measures for generating bilingual word dictionaries from parallel corpora. However, our context is different, since we do not have aligned paragraphs composed of words, but aligned words —previously aligned through GIZA++— composed of n-grams. A first choice could be just to adapt the contingency table to this context, by considering that we are managing n-gram pairs $(n\text{-}gram_s, n\text{-}gram_t)$ cooccurring in aligned words instead of word pairs $(word_s, word_t)$ cooccurring in aligned paragraphs. So, contingency tables should be adapted accordingly: O_{11}, for example, should be re-formulated as the number of aligned word pairs where the source language word contains $n\text{-}gram_s$ and the target language word contains $n\text{-}gram_t$.

This solution seems logical, but is not completely accurate. In the case of aligned paragraphs, we had *real* instances of word cooccurrences at the paragraphs aligned. However, now we do not have *real* instances of n-gram cooccurrences at aligned words, but just *probable* ones, since GIZA++ uses a statistical alignment model which computes a translation probability for each cooccurring word pair [9]. So, the same word may be aligned with several translation candidates, each one with a given probability. Taking as example the case of the English words `milk` and `milky`, and the Spanish words `leche` *(milk)*, `lechoso` *(milky)* and `tomate` *(tomato)*, a possible output word-level alignment would be:

source word	candidate translation	probability
milk	leche	0.98
milky	lechoso	0.92
milk	tomate	0.15

This way, it may be considered that the source 4-gram `-milk-` does not *really* cooccur with the target 4-gram `-lech-`, since the alignment between its containing words `milk` and `leche`, and `milky` and `lechoso` is not certain. Nevertheless, it seems much more probable that the "*translation*" of -milk- is -lech- rather than -toma-, since the probability of the alignment of their containing words —milk and tomate— is much smaller than that of the words containing -milk- and -lech- —the pairs milk and leche and milky and lechoso. Taking this idea as a basis, our proposal consists of weighting the likelihood of a cooccurrence according to the probability of its containing alignments.

So, the resulting contingency tables corresponding to the n-gram pairs *(-milk-, -lech-)* and *(-milk-, -toma-)* are as follows:

	$T =$ -lech-	$T \neq$ -lech-	
$S =$ -milk-	$O_{11} = 0.98 + 0.92 =$**1.90**	$O_{12} = 0.98 + 3 * 0.92 + 3 * 0.15 =$**4.19**	$R_1 =$**6.09**
$S \neq$ -milk-	$O_{21} =$**0.92**	$O_{22} = 3 * 0.92 =$**2.76**	$R_2 =$**3.68**
	$C_1 =$**2.82**	$C_2 =$**6.95**	$N =$**9.77**

	$T =$ -toma-	$T \neq$ -toma-	
$S =$ -milk-	$O_{11} =$**0.15**	$O_{12} = 2 * 0.98 + 4 * 0.92 + 2 * 0.15 =$**5.94**	$R_1 =$**6.09**
$S \neq$ -milk-	$O_{21} =$**0**	$O_{22} = 4 * 0.92 =$**3.68**	$R_2 =$**3.68**
	$C_1 =$**0.15**	$C_2 =$**9.62**	$N =$**9.77**

Notice that, for example, the O_{11} frequency corresponding to *(-milk-, -lech-)* is not 2 as might be expected, but 1.90. This is because the pair appears in two alignments, milk with leche and milky with lechoso, but each cooccurrence in an alignment has been weighted according to its translation probability:

$$O_{11} = 0.98 \text{ (for milk with leche)} + 0.92 \text{ (for milky with lechoso)} = 1.90 \ .$$

Once the contingency tables have been generated, the association measures can be computed. Our system employes two classic measures: the *Dice coefficient* (*Dice*) and *mutual information* (*MI*), defined by the following equations [5]:

$$Dice(n\text{-}gram_s, n\text{-}gram_t) = \frac{2O_{11}}{R_1 + C_1}. \quad (1) \quad MI(n\text{-}gram_s, n\text{-}gram_t) = log\frac{NO_{11}}{R_1 C_1}. \quad (2)$$

If using the Dice coefficient, for example, we find that the association measure of the pair *(-milk-, -lech-)* —the correct one— is much higher than that of the pair *(-milk-, -toma-)* —the wrong one:

$$Dice(\texttt{-milk-, -lech-}) = \frac{2*1.90}{6.09+2.82} = \mathbf{0.43} \ . \ Dice(\texttt{-milk-, -toma-}) = \frac{2*0.15}{6.09+0.15} = \mathbf{0.05} \ .$$

3 Evaluation

Before trying with less well-known languages with a greater lack of resources —which are the aim of this approach—, our system has to be tuned and studied more in depth. For this purpose, our approach has been initially tested in English-to-Spanish bilingual runs using the English topics and the Spanish document collections of the CLEF 2006 *robust task* [8]. The Spanish data collection is formed by 454,045 news reports (1.06 GB), while the test set consists of the

60 topics (C050–C059, C070–C079, C100–C109, C120–C129, C150–159, C180–189) of the *training topics* subset established for that task. Topics are formed by three fields: a brief *title* statement, a one-sentence *description*, and a more complex *narrative* specifying the relevance assessment criteria. Nevertheless, only *title* and *description* fields have been used, simulating in this way the case of "*short*" queries as those used in commercial engines [8].

Regarding the indexing process, documents were lowercased and punctuation marks —but not diacritics— were removed. Finally, the texts were split into n-grams and indexed, using 4-grams as a compromise n-gram size after studying the previous results of the JHU/APL group [7]. The open-source Terrier platform [1] has been employed as the retrieval engine, using a InL2[1] ranking model [2]. No stopword removal or query expansion were applied at this point.

For querying, the source language topic is firstly split into n-grams. Next, these n-grams are replaced by their candidate translations according to a selection algorithm, and the resulting translated topics are then submitted to the retrieval system. Two selection algorithms are currently available: a *top-rank-based* algorithm, that takes the N highest ranked n-gram alignments according to their association measure, and a *threshold-based* algorithm, that takes those alignments whose association measure is greater or equal than a threshold T.

Next, we present the results obtained with the association measures currently implemented in our system: the Dice coefficient and mutual information.[2]

3.1 Results Using the Dice Coefficient

Results Using Unidirectional Word-Level Alignment. Our first tests with the Dice coefficient used the top-rank-based selection algorithm, that is, by taking the target n-grams from the N top n-gram-level alignments with the highest association measures.[3] The best results were obtained when using a limited number of translations, those with N=1 being the best ones. Such results are displayed in the left-hand Precision vs. Recall graph of Fig. 1, labeled as 'W=0.00 N=1' —notice that mean average precision (MAP) values are also given.

The next tests used the threshold-based selection algorithm, that is, by fixing a minimal association measure threshold T.[4] The best run, using T=0.30, is shown in the left-hand graph of Fig. 1 labeled as 'W=0.00 T=0.30'. As can be seen, the results obtained were significantly less good as the previous ones.[5]

Next, trying to reduce the noise introduced in the system by word-level translation ambiguities and, in this way, to improve the n-gram alignment, we removed from the input those least-probable word alignments. After studying the distribution of the input aligned word pairs across their translation probabilities, we decided to dismiss those pairs with a probability less than a threshold W=0.15.

[1] Inverse Document Frequency model with Laplace after-effect and normalization 2.

[2] These experiments must be considered as *unofficial* experiments, since the results obtained have not been checked by the CLEF organization.

[3] With $N \in \{1, 2, 3, 5, 10, 20, 30, 40, 50, 75, 100\}$.

[4] With $T \in \{0.00, 0.001, 0.01, 0.05, 0.10, 0.20, 0.30, 0.40, 0.50, 0.60, 0.70, 0.80, 0.90, 1.00\}$.

[5] Two-tailed T-tests over MAPs with α=0.05 have been used along this work.

Table 1. General distribution of input aligned word pairs across their translation probabilities

	unidir. alignment		*bidir. alignment*	
	W=0.00	*W=0.15*	*W=0.00*	*W=0.15*
#pairs	2,155,482	66,610	672,502	32,011
μ	0.0233	0.2936	0.0287	0.3489
σ	0.0644	0.1845	0.0887	0.2116

Table 2. General distribution of output aligned n-gram pairs across their association measures: the Dice coefficient and mutual information

		unidir. alignment		*bidir. alignment*	
		W=0.00	*W=0.15*	*W=0.00*	*W=0.15*
#pairs		18,463,772	1,166,930	6,828,044	600,120
Dice	μ	0.0036	0.0644	0.0133	0.1439
	σ	0.0261	0.1355	0.0721	0.2252
MI	μ	-0.6672	4.3056	-0.1476	5.2094
	σ	3.8994	2.3019	4.0581	2.4206

This way we reduced the number of input pairs processed by 97%, from 2,155,482 to 66,610 —see Table 1—, and by 94% the number of output n-gram pairs generated, from 18,463,772 to 1,166,930 —see Table 2. This resulted in a considerable reduction of processing and storage resources, processing time included.

On the other hand, according to Tables 1 and 3, the level of ambiguity was reduced in both the input and output. In the case of the input, the mean number of possible translations per source word in the input word-level alignment was reduced from 41.1477 translations per source word with a mean probability of 0.0233, to 2.0049 translations with a mean probability of 0.2936. This implies a reduction of 95% in the number of possible translations and a parallel increase of 1160% in their mean translation probability.

Table 3. General distribution of source-language terms across their number of possible translations: in the input aligned word pairs (*left*), and in the output aligned n-gram pairs (*right*)

	input aligned word pairs				*output aligned n-gram pairs*			
	unidir. alignment		*bidir. alignment*		*unidir. alignment*		*bidir. alignment*	
	W=0.00	*W=0.15*	*W=0.00*	*W=0.15*	*W=0.00*	*W=0.15*	*W=0.00*	*W=0.15*
#terms	52,384	33,223	48,935	28,238	35,728	30,880	33,818	27,932
μ	41.1477	2.0049	13.7427	1.1336	516.7871	37.7892	201.9056	21.4850
σ	76.1284	1.4717	43.1740	0.3858	949.8868	82.6615	502.7873	50.0478

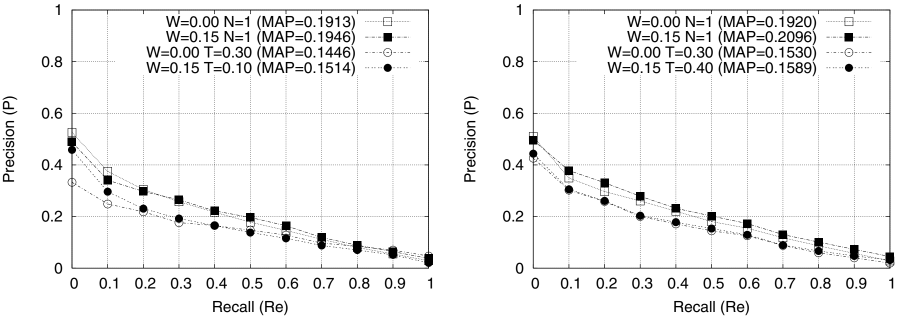


Fig. 1. Precision vs. Recall graphs of the test runs performed using the Dice coefficient and taking as input a unidirectional (*left*) or bidirectional (*right*) word-level alignment

In the case of the output, according to Tables 2 and 3, the mean number of possible translations per source n-gram in the output was reduced from 516.7871 translations with a mean association measure of 0.0036, to 37.7892 translations with a mean measure of 0.0644. This implies a reduction of 93% in the number of translations and a increase of 1689% in their association measure.

The results obtained introducing this refinement are no significantly different, in general, from those obtained without pruning, whatever the selection algorithm used. Those best results obtained for each selection approach —with N=1 and T=0.10— are shown in the left-hand graph of Fig. 1. As can be seen, the top-rank-based selection algorithm keeps performing significantly better.

So, we can conclude that although this refinement does not really improve the results, it reduces considerably those computing and storage resources required by the system, justifying its application. On the other hand, the system showed to be robust against the noise introduced by the high percentage of low-probability alignments of the input.

Results Using Bidirectional Word-Level Alignment. Once more, we tried to reduce the noise introduced in the system, this time by refining the initial word-level alignment by using a bidirectional alignment [4]. That is, we considered a $(word_{English}, word_{Spanish})$ English-to-Spanish alignment only if there also existed a corresponding $(word_{Spanish}, word_{English})$ Spanish-to-English alignment. This way we focus the processing on those words whose translation seems less ambiguous. The best results obtained for this approach are presented in the right-hand graph of Fig. 1. We will discuss now the impact of this refinement, taking as the baseline those runs obtained using a unidirectional algorithm where no minimal word-level translation probability threshold was fixed —i.e., W=0.

By examining Table 1 we can see that the bidirectional alignment reduced the number of input word pairs by 69% —from 2,155,482 to 672,502 pairs— and, according to Table 2, it reduced the number of output n-gram pairs by 63% —from 18,463,772 to 6,828,044 pairs. This reductions allows us to reduce both computing and storage resources —including processing time.

Regarding the level of ambiguity, in the case of the input, Tables 1 and 3 show a reduction from 41.1477 translations per input source word with a mean probability of 0.0233, to 13.7427 translations with a probability of 0.0287. This means a reduction of 67% in the number of translations and a increase of 23% in the translation probability of the input. In the case of the output, Tables 2 and 3 show a reduction from a mean of 516.7871 translations per source n-gram with a mean association measure of 0.0036, to 201.9056 translations with a measure of 0.0133; a reduction of 61% and a increment of 269%, respectively.

With respect to the results, the best ones, obtained again with N=1 and T=0.30, are shown in the right-hand graph of Fig. 1, being not significantly different from those obtained with the original unidirectional alignment, with the top-rank-based selection algorithm performing significantly better than the threshold-based approach. So, we can conclude that the use of bilingual alignment does not damage the performance of the system, and also reduces computing and storage resources —including processing time. The system was also demonstrated to be robust against inaccurate or ambiguous input alignments.

We have also considered combining the word-level bilingual alignment with the use of the word-level translation probability threshold W, looking for an extra reduction of both the level of ambiguity and the computing and storage resources needed. Taking as the baseline the results obtained when applying such a probability threshold W=0.15 over the original unidirectional alignment, Table 1 shows an extra 52% reduction —from 66,610 to 32,011 pairs— in the number of input word alignments, and an extra 49% reduction —from 1,166,930 to 600,120 pairs— in the output n-gram alignments.

With respect to the level of ambiguity, there is an extra 43% reduction — from 2.0049 to 1.1336 pairs— in the mean number of input word translations, with an 19% increment —from 0.2936 to 0.3489— of the mean word translation probability. In the case of the output n-gram translations, their mean number of translations was reduced from 37.7892 to 21.4850 pairs (43%), with a increase of the mean association measure from 0.0644 to 0.1439 (123%).

The results obtained, shown in the right-hand graph of Fig. 1, continue being not significantly different from the initial ones, with the top-rank-based selection algorithm performing significantly better. On the other hand, they show no apparent damage to the performance, allowing us to conclude that the combined use of both refinements minimizes the resources required by the system without harming its performance.

3.2 Results Using Mutual Information

Our second main set of experiments used mutual information (MI) as the association measure. The main difference with respect to the Dice coefficient is that the Dice coefficient takes values within the range $[0 \,..\, 1]$, while MI can take any value within $(-\infty \,..\, +\infty)$. Moreover, negative MI values correspond to pairs of terms avoiding each other, while positive values point out cooccurring terms. Finally, MI also tends to overestimate low-frequency data.

These features had to be taken into account in order to adapt our testing methodology. In the case of the top-rank-based selection algorithm, we continued taking the N top-ranked n-gram alignments, even if their MI value was negative. However, in the case of the threshold-based algorithm, since the range of MI values for each test run may vary considerably, the threshold values were fixed according to the following formula in order to homogenize the tests:

$$T_i = \mu + 0.5\, i\, \sigma\ . \tag{3}$$

where T_i represents the i-th threshold, with $i \in \mathbb{N}_0$, μ represents the *mean* of the MI values obtained for the present configuration, and σ its *standard deviation*. This way, the first threshold was fixed at $T_0 = \mu$, the following threshold at $T_1 = \mu + 0.5\ \sigma$, next at $T_1 = \mu + \sigma$, and so on, until reaching the highest possible threshold without overpassing the maximal MI value for the present configuration.

Results Using a Unidirectional Word-Level Alignment. This first test run corresponds to a unidirectional alignment using the top-rank-based selection algorithm with no word-level pruning —i.e., W=0.00. Results were not as good as those obtained using the Dice coefficient. The best run, that one using N=30, is presented in the left-hand graph of Fig. 2.

When introducing the word-level translation probability threshold W=0.15, the gains were the same as with the Dice coefficient, except for the mean association measure. This is because word-level gains —reduction of input word pairs and increment of the mean translation probability— only depend on the value of W, and are not affected by the association measure. At the n-gram level, the reduction in the number of output n-gram pairs only depends on the input word pairs —and, consequently, on W. Nevertheless, the mean association measures will vary, since we are now using MI. Mean values are shown in Table 2, and we can see how they increased from -0.6672 to +4.3056 (745%).

The results obtained were not significantly different from those obtained with W=0.00. The best ones, those for N=20, are shown in the left-hand graph of Fig. 2. As in the case of the Dice coefficient, the introduction of the threshold W does not damage the performance of the system, but reduces the computing and storage resources required. On the other hand, the system demonstrated again its robustness against the distortion introduced by low-probability inputs.

When using the threshold-based algorithm, results were slightly better than those with the top-rank-based algorithm —except at the lowest recall levels—, although this difference was not significant. Results improved when raising the threshold, but continued being not as good as those obtained with the Dice coefficient. The results for the best run, with $T = \mu + 2.5\ \sigma$, are shown in the left-hand graph of Fig. 2.

When pruning the input data by applying the word-level probability threshold W=0.15, the results seemed to approach even more those obtained with the top-rank-based algorithm. As before, no significant difference was found with respect to the results obtained without pruning. In this case the best threshold was $T = \mu + 0.5\ \sigma$, as shown in the left-hand graph of Fig. 2.

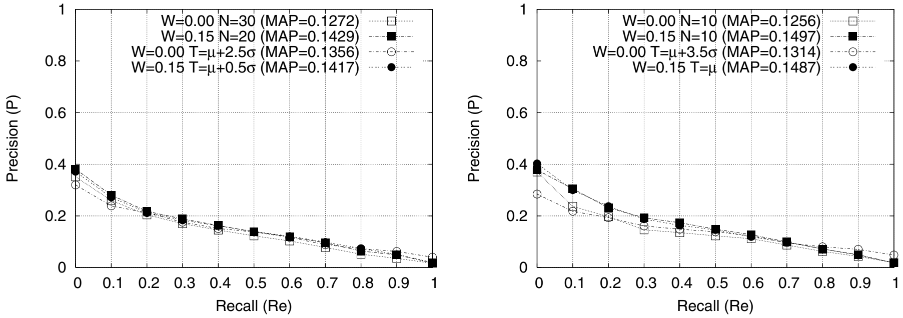


Fig. 2. Precision vs. Recall graphs of the test runs performed using mutual information and taking as input a unidirectional (*left*) or bidirectional (*right*) word-level alignment

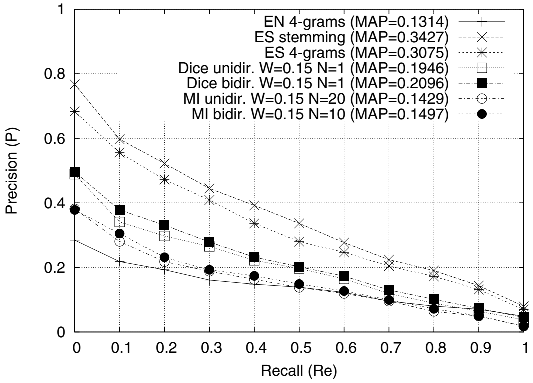


Fig. 3. Final summary Precision vs. Recall graph

Results Using a Bidirectional Word-Level Alignment. Our last set of test runs introduced again a word-level bidirectional alignment. The results obtained when using the top-rank-based selection algorithm were not significantly different from those obtained when employing a unidirectional alignment, whether we use W=0.00 or W=0.15 —see right-hand graph of Fig. 2.

As before, the gains obtained with the word-level threshold W were the same as with the Dice coefficient, except for the mean association measure. When taking as the baseline the unidirectional run with W=0.15, Table 2 shows an 21% increment of the mean MI value, from 4.3056 to 5.2094.

In the case of using a threshold-based selection algorithm, the results obtained were again not significantly different from those obtained with an unidirectional alignment, as shown in the right-hand graph of Fig. 2.

So, we can conclude that, as with the Dice coefficient, the introduction of a bidirectional alignment does not damage the performance of the system, but reduces the resources required. On the other hand, the system showed again its robustness against inaccurate or ambiguous input word alignments.

Finally, to complete this evaluation section, Fig. 3 shows the best results obtained for each combination of association measure and word-level alignment approach, with respect to several baselines: by querying the Spanish index with the English topics split into 4-grams (`EN 4-grams`) —allowing us to measure the impact of casual matches—, by querying the Spanish index using the stemmed Spanish topics[6] (`ES stemming`), and by querying the Spanish index using the Spanish topics split into 4-grams (`ES 4-grams`) —our ideal performance goal. As can be seen, the Dice coefficient in combination with the top-rank-based selection algorithm obtained the best results, performing significantly better than mutual information.

Although we still need to improve our results in order to reach our ideal performance goal, our current results are encouraging, since it must be taken into account that these are our very first experiments, so the margin for improvement is still great.

4 Conclusions and Future Work

This paper describes a system for character n-gram-level alignment in a parallel corpus and its use for direct translation of character n-grams in Cross-Language Information Retrieval. The algorithm proposed consists of two phases. In the first phase, the slowest one, the input parallel corpus is statistically aligned at word-level. In the second phase, n-gram association measures are computed — currently, the Dice coefficient and mutual information—, taking as input the translation probabilities calculated in the previous phase. This solution speeds up the training process, concentrating most of the complexity in the word-level alignment phase, making the testing of new association measures for n-gram alignment easier. On the other hand, two algorithms for the selection of candidate translations have been tested: a top-rank-based algorithm, which takes the N highest ranked n-gram alignments; and a threshold-based algorithm which takes those alignments according to a minimal threshold T.

Our experiments have shown that the Dice coefficient outperforms mutual information. In the case of using the Dice coefficient, the top-rank-based selection algorithm performs better. However, in the case of using mutual information, there is no apparent difference between the two selection algorithms available.

The use of a bidirectional alignment during the input word-level alignment and the introduction of a minimal word-level translation probability threshold have allowed us to reduce drastically both the number of input word alignments to be processed and the number of output n-gram alignments, but without damaging the performance of the system. This way, we can reduce considerably the computing and storage resources required, including processing time. Moreover, these experiments have demonstrated the robustness of the system against noisy or ambiguous input alignments.

With respect to our future work, new tests with other languages of different characteristics are being prepared in order to complete the tune of the system. We

[6] We have used the Snowball stemmer (`http://snowball.tartarus.org`), based on Porter's algorithm [10] and one of the most popular stemmers in IR research.

will also focus our effort on the development of new algorithms for the selection of candidate translations, and the application of new association measures.

Acknowledgment

This research has been partially funded by Ministerio de Educación y Ciencia and FEDER (TIN2004-07246-C03), Xunta de Galicia (PGIDIT05PXIC30501PN, PGIDIT05SIN044E, *Rede Galega de Procesamento da Linguaxe e Recuperación de Información*, and *Programa de Recursos Humanos* grants), and Universidade da Coruña. The authors desire to thank Prof. John Tait (Univ. of Sunderland, UK) for his support.

References

1. http://ir.dcs.gla.ac.uk/terrier/
2. Amati, G., van Rijsbergen, C.J.: Probabilistic models of Information Retrieval based on measuring divergence from randomness. ACM Transactions on Information Systems 20(4), 357–389 (2002)
3. Koehn, P.: Europarl: A Parallel Corpus for Statistical Machine Translation. In: Proc. of the 10th Machine Translation Summit, pp. 79–86 (2005) Corpus available in http://www.iccs.inf.ed.ac.uk/~pkoehn/publications/europarl/
4. Koehn, P., Och, F.J., Marcu, D.: Statistical phrase-based translation. In: Proc. of the 2003 Conf. of the North American Chapter of the ACL, pp. 48–54 (2003)
5. Manning, C.D., Schütze, H.: Foundations of Statistical Natural Language Processing. The MIT Press, Cambridge (1999)
6. McNamee, P., Mayfield, J.: Character N-gram Tokenization for European Language Text Retrieval. Information Retrieval 7(1-2), 73–97 (2004)
7. McNamee, P., Mayfield, J.: JHU/APL experiments in tokenization and non-word translation. In: LNCS, vol. 3237, pp. 85–97. Springer, Heidelberg (2004)
8. Nardi, A., Peters, C., Vicedo, J.L. (eds.): Working Notes of the CLEF 2006, Workshop (2006) Available at http://www.clef-campaign.org
9. Och, F.J., Ney, H.: A systematic comparison of various statistical alignment models, Source code (2003) available at http://www.fjoch.com/GIZA++.html
10. Porter, M.F.: An algorithm for suffix stripping. Program 14(3), 130–137 (1980)
11. Savoy, J.: Cross-Language Information Retrieval: experiments based on CLEF 2000 corpora. Information Processing and Management 39, 75–115 (2003)
12. Vilares, J., Oakes, M.P., Tait, J.I.: CoLesIR at CLEF 2000 rapid prototyping of a N-gram-based CLIR system. In: Nardi, et al. (2006) [8]

Cross-Lingual Information Retrieval by Feature Vectors

Jeanine Lilleng and Stein L. Tomassen

Department of Computer and Information Science,
Norwegian University of Science and Technology,
Sem Saelandsvei 7-9, NO-7491 Trondheim, Norway
{jeanine.lilleng,stein.l.tomassen}@idi.ntnu.no

Abstract. This paper investigates query translation in cross-lingual information retrieval, especially the challenges caused by ambiguity and polysemi. We base our ideas on feature vectors and our method uses context during the translation of queries. Achieving good query translation can be difficult, due to short queries lacking context information. We argue that by using information external to the query, like ontologies and document collections, the effect of ambiguity and polysemi can be reduced. Different approaches for translation of these feature vectors are proposed and discussed.

Keywords: cross-lingual information retrieval, query expansion, feature vector.

1 Introduction

Cross-lingual information retrieval (CLIR) has been a research area for many years and will be increasingly important. In 2001 Google had more than 2 billion Web pages in their index [1], where approximately half a billion of these was in non-English. In 2005 it was estimated that Google had indexed more than 8.1 billion Web pages [2], while the number of non-English pages was unknown. Additional, in January 2007 it was assumed that approximately 29% of the Internet users was speaking English [3] compared to while only 17% of the world's population was speaking English. Consequently, when more people start using the Web most of these will be non-English speakers [4]. Considering these figures it will be increasingly important to focus on high-quality CLIR techniques to make the Web truly available for all. In this paper we propose a flexible CLIR approach based on translation of feature vectors (*fv*s).

Monolingual information retrieval, where the language of the query and the document collection are the same, is obviously proven successful since searching is the most used tool on the Web. However, when it comes to cross-lingual information retrieval, where the language of the query and the documents are not necessarily the equal, the situation is quite different. To our knowledge, there are few CLIR systems available for the Web being of satisfactory quality, but for restricted domains (e.g. medicine) CLIR approaches has shown to be more lucrative.

As mentioned, there does exist some CLIR approaches on the Web showing potentials, where probably Babelplex [5] is the most prominent of them. Sadly enough there is little detailed information available for how Babelplex works.

Z. Kedad et al.(Eds.): NLDB 2007, LNCS 4592, pp. 229–239, 2007.

Nevertheless, it seems to be using a standard query translation approach where it translates the query terms by using Google Translate [6]. Next, both the original and the translated terms are submitted as two distinct queries to Google and finally the results of each query are presented side by side. However, Babelplex do suffer of the same typical limitations that are common for most CLIR approaches, and that is not being able to disambiguate the terms correctly and hence the translation is often of low quality.

Query interpretation is the first phase of an information retrieval session and the only part of the session that receives clear inputs from the user. Users tend to use very few terms, 3 or less, in their search queries [7, 8]. As a result, the system cannot disambiguate the terms correctly. By adding more relevant terms to the query the domain of interest can to some extend be identified. However, adding the correct terms is not always trivial, since the user needs knowledge about the terminology used in that particular domain to find those correct terms. Consequently, the users uses few terms that makes it equally difficult for the systems to correctly disambiguate the terms.

For closed or restricted domains CLIR approaches does traditionally produce better result compared to CLIR used in open domains. Typically a domain specific dictionary and thesaurus are used, as a result it is easier for a system to disambiguate the terms of a query and hence produce a better translation. Despite these promising results, they are highly depended on a fairly common terminology being used. Within the oil and gas industry, many companies usually have their own terminology (e.g., all the equipment available). Inconsistent usage of terminology causes problems in documents exchange among the industrial partners. The Integrated Information Platform for reservoir and subsea production systems (IIP) project [9], that partly funds this work, is creating an ontology for all subsea equipment used by the oil and gas industry. A goal of this project is to define an unambiguous terminology of the domain and build an ontology that will ease integration of systems between disciplines.

Ontologies can define concepts and the relationships among them [10] from any domain of interest. Considering multi-disciplinary domains and the big variation of terminology used one of the challenges is adoption of the created ontology to the document space. In our approach [11, 12], we use ontologies to define concepts in a particular domain. We use a query enrichment approach that uses contextually enriched ontologies to bring the queries closer to the user's preferences and the characteristics of the document collection. The idea is to associate every concept of the ontology with a feature vector to tailor these concepts to the specific terminology used in the document collection. Synonyms and conjugations would naturally go into such a vector, but we would also like to include related terms that tend to be used in connection with the concept and to provide a contextual definition of it. Afterward, the *fv*s are used to enrich the query provided by the user.

Since a feature vector includes only those terms found highly related to a concept we believe it can be automatically translated. Based on the semantic relations between the terms in a *fv* it is possible to automatically find a correct translation of each individual term. A correct translation is found and verified by finding an equal semantic relation between the set of translated candidate terms and the original terms of a *fv*. Those candidate terms found to have a similar semantic relation to the original *fv* are selected. The result of this will be a new translated *fv* with equally semantically related terms as the original *fv*.

This paper is organized as follows. In section 2, related work is discussed. In section 3, we describe the proposed approach for translation of feature vectors. Finally, in section 4 we discuss the potentials of this approach and conclude this paper.

2 Related Work

The related work to our approach comes from three main areas. Ontology based IR and cross-lingual information retrieval, in general, and approaches to query expansion, in particular. First, we will present some related work on ontology-based IR and query expansion and then on cross-lingual IR.

Some approaches combine both ontology based IR and the vector space model. For instance, some start with semantic querying using ontology query languages and then use resulting instances to retrieve relevant documents [13]. Nagypal [14] combines ontology usage with vector-space model by extending a non-ontological query. There, ontology is used to disambiguate queries. Paralic et al. [15] describes a similar approach where documents are associated with the concepts in an ontology. The concepts in the query are matched to the concepts of the ontology in order to retrieve terms and then used for calculation of document similarity.

Most query enrichment approaches are not using ontologies like [16, 17, 18, 19, 20]. Typically, query expansion is done by extending the provided query terms with synonyms or hyponyms (cf. [21]). Some approaches are focusing on using ontologies in the process of enriching queries [22, 23, 24]. However, an ontology in such a case typically serve as a thesaurus containing synonyms and hypernyms/hyponyms, and do not consider the context of each term (i.e. every term is equally weighted).

Qiu et al. [18] is using query expansion based on similarity thesaurus. Weighting of terms is used to reflect the domain knowledge. The query expansion is done by similarity measures. Similarly, Grootjen et al. [17] describes a conceptual query expansion. There, the query concepts are created from a result set. Both approaches show an improvement compared to simple term based queries, especially for short queries.

Adi describes in [20] a commercial search engine that provides three basic search strategies; word, concept, and super-concept search respectively. A concept is represented as a set of words, while a super-concept is a combination of several closely related concepts. The user can mix strategies when searching. Unfortunately, there are not enough details provided by Adi [20] to state how this work.

The approach presented by Ozcan et al. [24] is using ontologies for the representation of concepts. The concepts are extended with similar words using a combination pf Latent Semantic Analysis (LSA) and WordNet [25]. The approach gets promising results for short or poorly formulated queries.

Cross-lingual information retrieval is information retrieval with the added challenge of at least two different languages. The early approach to this challenge was to translate the query before the translated query was submitted to the IR system in the same language as the documents to be searched, an example of this is by Quilt [26]. However, ambiguity and polysemy causes significant problems when the query is translated [27]. The challenges are similar to the experienced difficulties in query expansion [28].

Techniques used by Lui et al. [29] to achieve word sense disambiguation in queries might be considered similar to our technique. However, their technique is based on WordNet [25]. This will give good results in general queries, but the WordNet coverage is not very good for more narrow domains (e.g., oil and gas).

3 Approach

In a cross-lingual information retrieval system the query and the documents to be searched are written in different languages. This challenge has one principal solution; translation. The question then becomes what to translate, the query, the documents, or both. Translating the query can be done in runtime, but due to the fact that queries often are very short, it might be difficult to disambiguate the terms. If the documents are translated, more information to disambiguate during the translation is available, but both the required processing time and disk space needed, will be substantial at best. The disk space requirement for n number of supported languages will be n times the original space. The final alternative is to use a common interlingua and translate both the queries and the documents to this language. Obviously this has all the same disadvantages regarding disambiguation as with query translation, but with interlingua only one translation of the queries have to be done and the documents will be independent of the number of languages.

In this paper we will investigate a situation with two languages, and will not investigate an interlingua approach. In addition, we focus on translation done on the

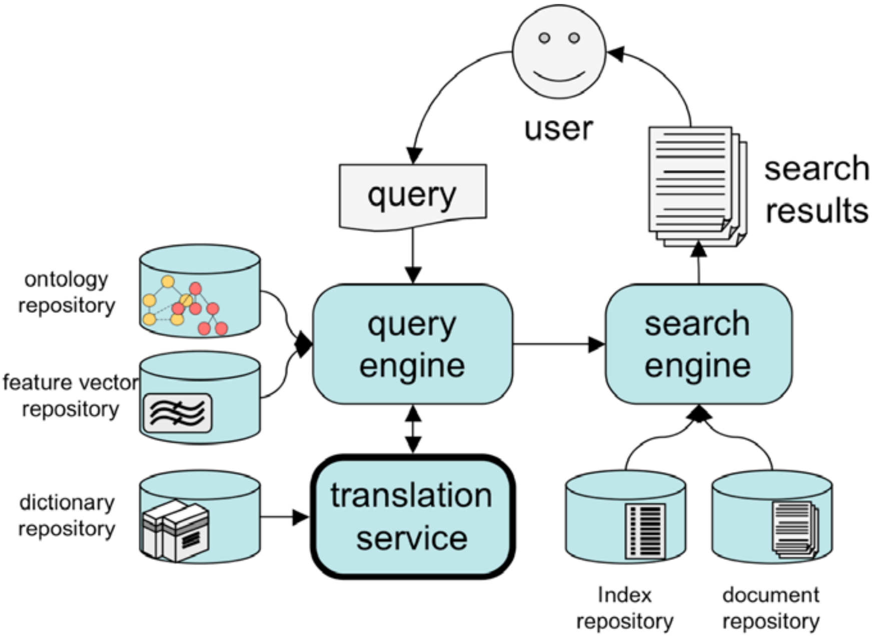


Fig. 1. The overall architecture of the approach. The *translation service* component is an extension to an existing ontology-driven information retrieval system under development, and is the focus of this paper.

query side in order to combine with the existing monolingual IR system. Therefore, this approach will be an extension of an earlier developed ontology-driven information retrieval (OdIR) system [11, 12] that uses ontologies tailored to the document collection by feature vectors (see Figure 1). The *fv*s are used to enhance the user queries before they are submitted to the IR system.

The expected improvements in query translation caused by the *fv* approach are caused by the information added to the *fv*s from the ontologies and the incorporated document collection (see [11] for further information of the process of creating *fv*s). However, the language resources added to a translation solution are always a limiting factor.

3.1 Query Translation

Having chosen to translate on the query side reduces the possible solutions somewhat. However, the query passes through three different forms or phases before it is submitted to the IR system; *user query*, *feature vector*, and *enriched query* respectively (see Figure 2). Any of the three forms can be used for translation from the source language to a target language. The chosen phase will affect both the quality of the translation and the number of resources required in the system. Next, the various alternatives will be discussed.

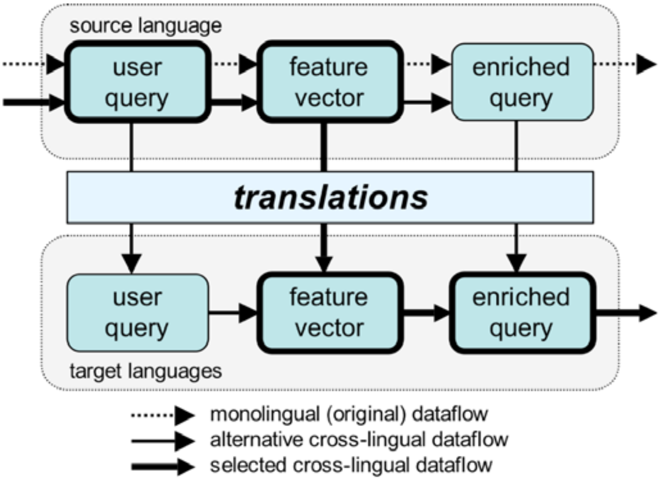


Fig. 2. The various translation approach alternatives. The selected translation approach is shown with bold lines. Note that the original query, depicted as dotted lines, is also sent to the search engine.

User query. If we choose to translate the user query, a full set of resources is needed for every supported language. This means either a comparable ontology in the target language must be available or a translation of one must be done. Using machine translation will cause reduced quality of both the feature vectors and the final enriched query. One could imagine that translating the ontology and using the target language could create better *fv*s than by translating the *fv*s directly. However, according to Fung [30] semantically similar terms occur in similar context and similar frequency across languages within the same timeframe and domain.

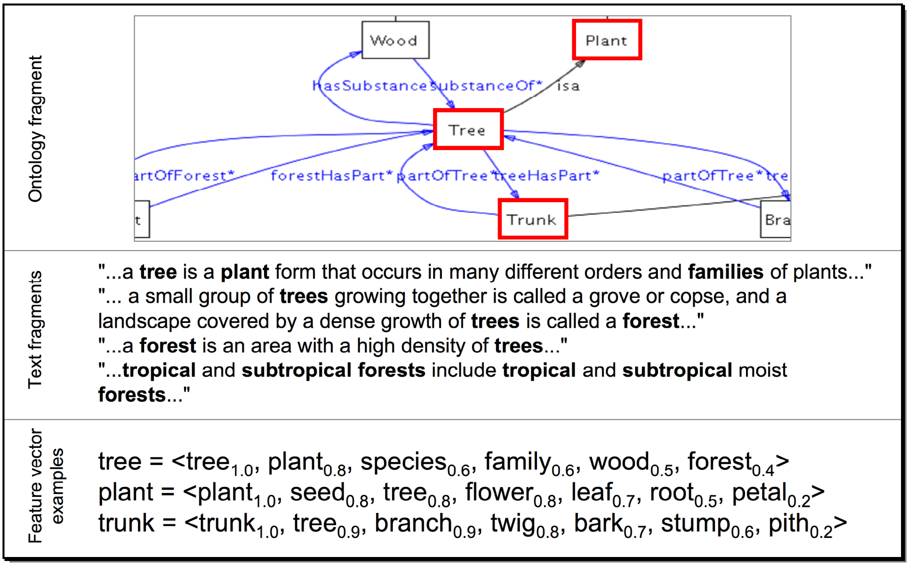
Feature vector. There exists a feature vector for every term in the query. To create these feature vectors both an ontology and the information from statistical analysis of the documents are used. Differences in coverage, granularity, and focus are reduced. Hence, the *fv*s are both domain specific due to the ontology used and adjusted to fit the document collection where the query is to be used. Since the terms of a *fv* are semantically related the possibility for good automatic disambiguation and hence a good translation will be more probable than when translating a few words (e.g. the original query).

Enriched query. The enriched query is a union of all the *fv*s of all the terms found in the original query. This is the last possible resource for translation. However, since the enriched query is the union of all the *fv*s, and consequently lacks the distinct *fv*s used for disambiguation during translation, it is difficult to see how this would be a good alternative.

Based on the pros and cons of the various alternatives discussed above we have chosen to translate form two, feature vectors (see Figure 2). The translation of *fv*s approach will be discussed next.

3.2 Translation of Feature Vectors

Before the enriched query can be created, the feature vectors corresponding to the submitted query must be translated to a selected target language. In this section, we will describe two approached for how these *fv*s can be translated, but first a method for how we can check for applicability.



Text fragments

"...a **tree** is a **plant** form that occurs in many different orders and **families** of plants..."
"... a small group of **trees** growing together is called a grove or copse, and a landscape covered by a dense growth of **trees** is called a **forest**..."
"...a **forest** is an area with a high density of **trees**..."
"...**tropical** and **subtropical forests** include **tropical** and **subtropical** moist **forests**..."

Feature vector examples

tree = <tree$_{1.0}$, plant$_{0.8}$, species$_{0.6}$, family$_{0.6}$, wood$_{0.5}$, forest$_{0.4}$>
plant = <plant$_{1.0}$, seed$_{0.8}$, tree$_{0.8}$, flower$_{0.8}$, leaf$_{0.7}$, root$_{0.5}$, petal$_{0.2}$>
trunk = <trunk$_{1.0}$, tree$_{0.9}$, branch$_{0.9}$, twig$_{0.8}$, bark$_{0.7}$, stump$_{0.6}$, pith$_{0.2}$>

Fig. 3. Explanatory example of three concept feature vectors, including the ontology being used and some related text fragments. These *fv*s are also used to exemplify the translation approaches.

A good method to check for applicability seems to first translate the feature vectors to a target language then back to the source language again. If they are equal, it seems reasonable to assume that the translation chosen conserves the semantic content of the feature vectors. Therefore, the hypothesis is that the more equal the content of the translated *fv* are with the initial *fv*, the more successful is the translation approach.

Figure 3 depicts an explanatory example of an ontology describing trees, some related text fragments from a document collection (Wikipedia [31] is used in this example), and three corresponding examples of feature vectors. These *fv*s are considered to be of average difficulty, regarding translation. These *fv*s will also be used to exemplify the translations to German described next.

Translation of every term

The intuitive solution is to choose the first suggested translation in a dictionary. This is comparable to submitting one term to a machine translation system and directly use the translated term returned. This approach will provide the translation shown in Figure 4.

plant$_{\text{(original)}}$ = <plant$_{1.0}$, seed$_{0.8}$, tree$_{0.8}$, flower$_{0.8}$, leaf$_{0.7}$, root$_{0.5}$, petal$_{0.2}$>
plant$_{\text{(German)}}$ = <Pflanze$_{1.0}$, säen$_{0.8}$, Baum$_{0.8}$, Blume$_{0.8}$, Blatt$_{0.7}$, Fuss$_{0.5}$, Blumenblatt$_{0.2}$>
plant$_{\text{(English)}}$ = <plant$_{1.0}$, **sow**$_{0.8}$, tree$_{0.8}$, flower$_{0.8}$, **sheet**$_{0.7}$, **feet**$_{0.5}$, petal$_{0.2}$>

trunk$_{\text{(original)}}$ = <trunk$_{1.0}$, tree$_{0.9}$, branch$_{0.9}$, twig$_{0.8}$, bark$_{0.7}$, stump$_{0.6}$, pith$_{0.2}$>
trunk$_{\text{(German)}}$ = <Kabel$_{1.0}$, Baum$_{0.9}$, Zweig$_{0.9}$, Zweig$_{0.8}$, Bark$_{0.7}$, Stummel$_{0.6}$, Mark$_{0.2}$>
trunk$_{\text{(English)}}$ = <**cable**$_{1.0}$, tree$_{0.9}$, **arm**$_{0.9}$, **arm**$_{0.8}$, **barque**$_{0.7}$, **snag**$_{0.6}$, pith$_{0.2}$>

Fig. 4. Translation of every term of the feature vectors, first for `plant` then for `trunk`. For each concept a feature vector being the original (being in English), the German, and finally the one translated back to English again.

If the method retained the semantics of the feature vectors 100%, then the twice-translated *fv*s should be identical to the original *fv*s. Even though both these examples are considered to be of average difficulty only half of the original terms can be found in the twice-translated feature vectors. The results could have been better if the terms found were synonyms with the original words. Unfortunately, in these two examples, they were not. Hence it seems reasonable to conclude that this translation technique is not adequate.

Context dependent translation

Recall that a feature vector is representing a concept and includes only those terms that tend to be used in connection with that concept. We believe the quality of these translated *fv*s can be improved if the semantic information contained in the feature vectors also is used.

Table 1 shows two of the 23 possible direct translations found by the LEO's dictionary [32] for the term `root`. Typically, a term will often have several alternative translations. However, in this example we have selected only two for the term `root`; the one found to be most correct and the one chose by the direct translation approach. For each translation corresponding synonyms are found. The synonyms shown here was found in online dictionaries [33, 34].

Table 1. Two translation matches by the LEO's dictionary for the term `root` and corresponding synonyms for each translation

Source language	Target language	
Term	Suggested translation	Synonyms
root	fuss	*[Fundament, Sockel, Unterbau]*
	wurzel	[Wurzelgeflecht, Radix, Wurz] [Anlass, Ansatzpunkt, Ausgangspunkt, Auslöser, Basis, Entstehung, Entstehungsort, Grundlage, Herkunft, Keimzelle, Kristallisationspunkt, Quelle, Ursache, Ursprung, Wiege]

The same process is repeated for all possible translations of the feature vector terms, which gives a large number of alternative final feature vectors. To identify the best translation, the synonym vectors for all the translated terms are compared. Since a lot of additional inaccuracies typically are introduced during translation, we have chosen to do all the comparison in the target language. The combinations of synonym vectors that are most similar are considered correct. Similarity is measured by number of similar words, words that have similar root, or word parts. We expect this to give a better and more context dependent translation.

$\text{plant}_{(\text{original})} = <\text{plant}_{1.0}, \text{seed}_{0.8}, \text{tree}_{0.8}, \text{flower}_{0.8}, \text{leaf}_{0.7}, \text{root}_{0.5}, \text{petal}_{0.2}>$
$\text{plant}_{(\text{German})} = <\text{Pflanze}_{1.0}, \text{Korn}_{0.8}, \text{Baum}_{0.8}, \text{Blume}_{0.8}, \text{Blatt}_{0.7}, \text{Wurzel}_{0.5}, \text{Blumenblatt}_{0.2}>$
$\text{plant}_{(\text{English})} = <\text{plant}_{1.0}, \text{seed}_{0.8}, \text{tree}_{0.8}, \text{flower}_{0.8}, \text{leaf}_{0.7}, \text{root}_{0.5}, \text{petal}_{0.2}>$

Fig. 5. The improved translation approach after including contextual information in the translation process

The result of this translation approach is shown in Figure 5 for the concept `plant`. Translating `Blatt` back to English can be a challenge, but becomes correct when using the technique described above for the German to English translation as well. In this example the approach retained the semantics of the *fv* 100%, that is, the twice-translated *fv* was identical to the original *fv*. For this reason it seems reasonable to conclude that this translation technique is feasible, but more thorough testing must be done to assess the utility of the approach.

4 Discussion and Conclusion

In this paper we have proposed a novel approach to cross-lingual information retrieval based on feature vectors. We have argued that directly translation of feature vectors can be sufficient for IR applications. However, as the research reported here is still in progress we have not been able to fully implement and evaluate this approach. Even so, we believe the method shows potential because of the quality and the semantic information that these feature vectors possess, which is important and used in the translation process.

To automatically find the correct translation of a term is typically very difficult. The main reason for this is that a term can have many different meanings being highly dependent on the context. Since a typical user tends to use three or less terms in a search query it is difficult, and in most cases impossible, to identify the correct context and hence the correct translation of the query. Consequently, the translation can be totally wrong or all possible translations of the terms must be included. The latter solution will include a lot of noise when searching and is therefore not satisfying. However, for narrow domains the system has some knowledge of the context and consequently the translation can be done more correctly. The terms of a *fv*, on the other hand, are semantically related which provide the system with contextual information that can provide better translation of a query.

The characteristic of a *fv* is dependent on the quality of both the ontology and the document collection being used. However, both the ontology and the document collection are somewhat independent of the approach described in this paper. For instance, there does not exist only one approach for how to create an ontology. One of the reasons for this is that there are many different views of what is considered to be a good ontology. Consequently, the quality of these ontologies will vary a lot depending on the creator. The quality of the documents in a corpus can also vary a lot (e.g., documents found on the Internet). Another important issue is that a good ontology can be applied on a mismatched document collection (e.g., a medical ontology used within the oil and gas domain). All these issues mentioned do have an impact on the final quality of the feature vectors and consequently influence of the translation of these as well, but they are considered all to be external aspects to this approach. In this paper it is assumed that the *fv*s are adequate.

Since we consider the quality of these *fv*s acceptable then we also believe that automatic translation of these can provide satisfying results. Given that a *fv* of a concept only include terms in the document collection that tend to be used in connection with that particular concept, then all those terms are assumed to be semantically related. Based on these semantic relations we believe that it will be possible to find a correct translation of each individual term. To find the likely correct translation of each term we compare with the set of possible translations of the other semantically related terms of the *fv*. Those possible translations that are semantically related are also assumed to be the correctly translated. The result of this will be a new translated *fv* with equally semantically related terms as the original *fv*.

In this paper we have presented two different approaches for how the feature vectors can be translated. The first, *translation of every term*, described a direct translation approach where each term was independently translated of each other. The first translation that the dictionary provided was selected. This approach did not give adequate results, which was not surprising. In the next approach, *context dependent translation*, the semantic relation between the terms was also used in the translation process. In the exemplified result, the twice-translation gave 100% match with the original *fv*. That was only one example and consequently more thorough testing needs to be done before we can conclude how successful this approach is.

As the research reported here is still in progress we need to fully implement the approach for more thorough testing and evaluation. We believe an advantage with this approach is the adaptability to several languages, which can be done by adding other dictionaries and thesauruses. However, that has to be fully tested before we can conclude. We will also have to investigate alternative methods for the translation of the feature vectors. For example, the *context dependent translation* technique described has a major shortcoming; a rather marginal term, with low weighting, has the same influence as more important terms. Therefore, we will investigate methods where the weighting of the terms can be taken into consideration as well.

Acknowledgements. This research work is partly funded by the Integrated Information Platform for reservoir and subsea production systems (IIP) project, which is supported by the Norwegian Research Council (NFR). NFR project number 163457/S30.

References

1. Google: 3 Billion Document Index (22.02.2007) http://www.google.com/3.html
2. Search Engine Size Wars V Erupts (22.02.2007) http://blog.searchenginewatch.com/blog/041111-084221
3. Internet world users by language (22.02.2007) http://www.internetworldstats.com/stats7.htm
4. Allan, J. et al.: Challenges in Information Retrieval and Language Modelling: report of a workshop held at the centre for intelligent information retrieval (2002)
5. Babelplex (22.02.2007) http://babelplex.com/
6. Google Translate (22.02.2007) http://www.google.com/translate
7. Gulla, J.A., Auran, P.G., Risvik, K.M.: Linguistic Techniques in Large-Scale Search Engines. Fast Search & Transfer, p. 15 (2002)
8. Spink, A., Wolfram, D., Jansen, M.B.J., Saracevic, T.: Searching the Web: the public and their queries. J. Am. Soc. Inf. Sci. Technol. 52, 226–234 (2001)
9. Gulla, J.A., Tomassen, S.L., Strasunskas, D.: Semantic Interoperability in the Norwegian Petroleum Industry. In: Karagiannis, D., Mayer, H.C. (eds.) Proceedings of the 5th International Conference on Information Systems Technology and its Applications (ISTA 2006) vol. P-84. Köllen Druck+Verlag GmbH, Bonn, Klagenfurt, Austria pp. 81–94 (2006)
10. Gruber, T.R.: A translation approach to portable ontology specifications. Knowledge Acquisition 5, 199–220 (1993)
11. Tomassen, S.L., Strasunskas, D.: Query Terms Abstraction Layers. In: Meersman, R., Tari, Z., Herrero, P. (eds.) On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops. LNCS, vol. 4278, pp. 1786–1795. Springer, Heidelberg (2006)
12. Tomassen, S.L., Gulla, J.A., Strasunskas, D.: Document Space Adapted Ontology: Application in Query Enrichment. In: Kop, C., Fliedl, G., Mayr, H.C., Métais, E. (eds.) NLDB 2006. LNCS, vol. 3999, pp. 46–57. Springer, Heidelberg (2006)
13. Kiryakov, A., Popov, B., Terziev, I., Manov, D., Ognyanoff, D.: Semantic Annotation, Indexing, and Retrieval. Journal of Web Semantics 2(1) (2005)
14. Nagypal, G.: Improving Information Retrieval Effectiveness by Using Domain Knowledge Stored in Ontologies. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM Workshops 2005. LNCS, vol. 3762, pp. 780–789. Springer, Heidelberg (2005)

15. Paralic, J., Kostial, I.: Ontology-based Information Retrieval. Information and Intelligent Systems, Croatia, pp. 23–28 (2003)
16. Rajapakse, R.K., Denham, M.: Text retrieval with more realistic concept matching and reinforcement learning. Information Processing & Management 42, 1260–1275 (2006)
17. Grootjen, F.A., van der Weide, T.P.: Conceptual query expansion. Data. & Knowledge Engineering 56, 174–193 (2006)
18. Qiu, Y., Frei, H.-P.: Concept based query expansion. In: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 160–169. ACM Press, Pittsburgh, Pennsylvania, USA (1993)
19. Chang, Y., Ounis, I., Kim, M.: Query reformulation using automatically generated query concepts from a document space. Information Processing and Management 42, 453–468 (2006)
20. Adi, T., Ewell, O.K., Adi, P.: High Selectivity and Accuracy with READWARE's Automated System of Knowledge Organization. Management Information Technologies, Inc. (MITi) (1999)
21. Chenggang, W., Wenpin, J., Qijia, T., et al.: An information retrieval server based on ontology and multiagent. Journal of computer research & development 38(6), 641–647 (2001)
22. Ciorăscu, C., Ciorăscu, I., Stoffel, K.: knOWLer - Ontological Support for Information Retrieval Systems. In: Proceedings of Sigir 2003 Conference, Workshop on Semantic Web, Toronto, Canada (2003)
23. Braga, R.M.M., Werner, C.M.L., Mattoso, M.: Using Ontologies for Domain Information Retrieval. In: Proceedings of the 11th International Workshop on Database and Expert Systems
24. Ozcan, R., Aslangdogan, Y.A.: Concept Based Information Access Using Ontologies and Latent Semantic Analysis. Technical Report CSE-2004-8. University of Texas at Arlington 16 (2004)
25. WordNet (22.02.2007) http://wordnet.princeton.edu/
26. Davis, M.W., Ogden, W.C.: QUILT: implementing a large-scale cross-language text retrieval system. In: 20th annual international ACM SIGIR conference on Research and development in information retrieval, Philadelphia, Pennsylvania, United States (1997)
27. Chen, H.-H. et al.: Resolving translation ambiguity and target polysemy in cross-language information retrieval. In: Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics, College Park, Maryland (1999)
28. Stokoe, C. et al.: Word sense disambiguation in information retrieval revisited. In: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, Toronto, Canada (2003)
29. Liu, S. et al.: Word sense disambiguation in queries. In: Proceedings of the 14th ACM international conference on Information and knowledge management, Bremen, Germany (2005)
30. Fung, P.: A Statistical View on Bilingual Lexicon Extraction: From Parallel Corpora to Non-parallel Corpora (1998)
31. Wikipedia (22.02.2007) http://en.wikipedia.org
32. English Wörterbuch (14.02.2007) http://dict.leo.org/
33. Duden (14.02.2007) http://www.duden-suche.de/
34. Das digitale Wörterbuch der deutchen Sprache des 20. Jh (14.02.2007) http://www.dwds.de/

Incomplete and Fuzzy Conceptual Graphs to Automatically Index Medical Reports

Loic Maisonnasse, Jean Pierre Chevallet, and Catherine Berrut

LIG - 38041 Grenoble Cedex 9

Abstract. Most of Information Retrieval (IR) systems are still based on bag of word paradigm. This is a strong limitation if one needs high precision answers. For example, in restricted domain, like medicine, user builds short and precise query, like "Show me chest CT images with emphysema.", and expects from the system precise answers. In such a case, the use of natural language processing to model document content is the only way to improve IR precision. This paper presents a model for text IR that index documents with Fuzzy Conceptual Graphs (FCG). Building automatically a complete and relevant conceptual structure is known to be a difficult task. To overcome this problem and keeping automatic graph building, we promote the use of incomplete FCG. We show how to deal with this incompleteness by using confidence. This confidence is attached to concepts and conceptual relations. As we use FCG as index, the matching process is based on a fuzzy graph matching. Finally, our experiments show that this outperforms classical word based indexing.

1 Introduction

Conceptual Indexing, i.e. the use of concepts instead of words or terms in an Information Retrieval (IR) system, seems a nice idea to cross the language barrier and also to come out with a much meaning full document index. Unfortunately, even if this idea is quite old [1], it is still difficult to come out with a conceptual automatic texts indexing, because it requires a knowledge base and some strong Natural Language Processing (NLP). Some conceptual models and structures has already being proposed in IR, like terminological logic [2] or conceptual dependencies [3]. Unfortunately such languages are still difficult to build automatically from texts and complex to use for matching. So, how to overcome this difficulty?

We promote the use of conceptual graphs [4] as document index, extended to Fuzzy Conceptual Graph (FCG) model and experiment automatic building of *incomplete graphs* from texts. Concepts and relations used to build a conceptual graph can be detected using a knowledge base with the help of syntactic information. Producing an accurate conceptual graph is difficult due to ambiguity of natural language and analysis errors. For this reason using an FCG that includes a detection confidence score for both query and document index, is a good solution for IR because IR is already based on fuzzy matching. In this work, we

Z. Kedad et al.(Eds.): NLDB 2007, LNCS 4592, pp. 240–251, 2007.

propose a variation of fuzzy graph matching. We compare this method to a bag of concept and a bag of word indexing. Results show that using FCG is the best solution when domain knowledge is available.

In this paper, we first present in 2 previous works on syntactic or semantic indexing structure. Then in 3 we present our FCG model. Finally we evaluate this model on the Medical CLEF 2005 collection.

2 Related Work

It is surprising that most of current IR researches still use the Bag of Word (BoW) paradigms, and some sophisticated probabilistic matching, that are more or less related to a weighted term intersection. Beside this fact, some research has been performed using NLP applied to IR. These approaches can be split in two types: works using syntactic information, and those using knowledge base to build the index. We briefly review these two on the following.

2.1 Using Syntactic Structures

A syntactic structure based index is a graph of terms where relationships express *syntactic links* between words. For example, a noun `subject` of a verb. Shallow parsing is the incomplete grammatical analysis of text and propose minimal syntactic structure using limited linguistic resources [5]. Shallow parsers are usually robust, i.e. always propose a possible parsing for every sentences. They also require less computing power. This is probably the reason why they are used in IR. Also as IR is concerned about discovering document theme, only noun phrase basic structure (head and dependent) may be useful for this task.

Among many syntactic shallow parsers available, some (mainly dependency parsers) produce a tree structure for sentence representation. Theses dependency structures are used to extract phrases (ex: noun phrase). In [6, 7] authors produce a dependency tree for all documents sentences. They extract phrases by apply patterns on the dependency tree. Their goal is to enhance BoW model by adding selected phrases, with a simple tf.idf weighting scheme, adjusted to enhance idf on phrases. They claim improvement of IR results. To our opinion, this gain cannot be directly linked to dependency structures which are only used to detect phrases: they do not keep in the index the original phrase structure as index are vector of set of word. For example ‘education by research’ is represented by {*education, research*}, and can also be related to ‘education by research’, ‘research on education’ or ‘research for education’. Syntactic structure drastically changes the meaning.

To solve this problem, some researchers tried to directly exploit syntactic dependency structure for indexing [8, 9]. They directly exploit dependencies trees, extracted from sentences and match query and documents by a *partial projection* of the query tree on documents trees. As preceding works provided only one unambiguous structure by sentence, in [10] Smeaton incorporates syntactic ambiguity in structure. His model is applied on phrases and similarity is provided by tree matching. But IR results obtained by this method are lower than results obtained by considering only phrases represented by trees.

In fact, even if some improvement has been shown, the use of syntactic structure for IR is rather limited, because of the lake of explicit semantic.

2.2 Using Semantic Structures

Using semantic structures for indexing should be more powerful than only syntax. There are already some attempts toward this direction, like [11] that uses semantic relation, COREL system [12] that uses frames-based indexing, or MIRTL [2] that uses terminological logic. In this model, a document is represented by individual constants and a query by a concept; a concept is defined as a group of individuals determined by a set of constraints (roles). The retrieval task is modelled as a logic subsumtion. The RIME [3] system is based on Schank dependency tree, extracted automatically from medical report and logic deduction for matching. Conceptual graphs have also already been used in IR. Work [13] has shown that matching can be done by applying a projection algorithm that maps query graph to document graph and that projection is equivalent to first order logic implication. But projection is not completely satisfying; ordering documents on relevance is impossible and documents have to completely satisfy queries to be retrieved. Different researchers have tried to tackle these problems [14, 13].

The main problem for semantic indexing remains in the difficulty to produce automatically such a structure. In fact it is mandatory to have access to important amount of formalized knowledge and to tackle difficult linguistic phenomenon like ambiguity. Moreover, matching functions used for such representations are most of time not enough efficient. Even so, semantic index seems more suitable than syntactic index to represent document content as it better captures sentence meaning.

As extracting precise and complete semantic representation is difficult, we propose incomplete FCG index and test the effectiveness of such approach on a test collection in a restricted domain.

3 Fuzzy Conceptual Graph Indexing

Why the use of concept instead of word or term, should lead to a better indexing ? Indexing using terms (e.g. ‘chest CT’) improves precision as a term is less ambiguous than a single words. But it can lead to a recall problem due to term variation and synonymy (e.g. ‘Computed tomography of chest’). Indexing at the conceptual level solves this problem because concepts are abstraction of terms. Also a conceptual indexing is naturally multilingual, as concepts can be defined as human understandable unique abstract notions independent from any direct material support, independent from any language or information representation.

Moreover relations between concepts are also important, because they place concept at a given role, for example in ‘blood smears that include polymorphonuclear neutrophils’.[1]

[1] Query from 2006 CLEF medical test collection.

It is a fact that NLP is mandatory to produce conceptual representation of document. To produce precise and linguistically correct structure we need to disambiguate terms and relations. In non IR application (ex: translation), NLP is evaluated through out precision and completeness. In IR application, NLP does not necessarily need to reach a top precision to have a positive impact in the overall IR system. Words stemming, for example, is neither linguistically correct nor complete but sufficient for good average performance: reducing both 'admiral' and 'admire' to 'admir' still works when searching with 'admiral' because of the context of other words. Disambiguation is usually an important step in text conceptualization. In IR it may be not that problematic because a super set of concepts in indexes may not interfere with the matching process if query is correctly disambiguated. Hence entire document coverage seems more important than complete ambiguity resolution. On the other hand, erroneous solved ambiguity may lead to recall problem.

In our work, we experiment simple disambiguation methods on document side that enables us the building of an incomplete graph with ambiguous concept or relation. We represent ambiguity and other NLP errors into the FCG by a confidence score.

3.1 Fuzzy Conceptual Graph IR Model

We choose to represent document indexes using FCG based on Wuwongse [15] conceptual graph theory extension. This extension includes fuzzy referents, and fuzzy conceptual relations. The fuzzification appears between the referent and the type of concept: the system has detected an object he assign the referent $\#n$ but its concept type is not certain. The FCG model of Mulhem [16] is a little different: his model is centered on the notion of referent.

We recall here some element of this model where we divide fuzzyness in two score. In a support $S = (Tc, Tr)$ where Tc is a partial order on concepts and Tr a partial order on relations, a fuzzy conceptual graph $G(C, R)$ is composed of two sets: the fuzzy concepts set C and the fuzzy relations set R.

- A *fuzzy concept* is represented by $[t : x|v, w_c]$ where t is the concept type, x is the referent and v denotes the confidence of assigning the concept type t to a referent x in the document, and w_c is the relevance of this concept to the document.
- A *fuzzy conceptual relation* is represented by $(\mathit{type}(r)|u_r, w_r)$ where r is a tupple of fuzzy concepts of C and $\mathit{type}(r)$ the relation type from Tr. This tupple r is ordered and numbered so that each position in the tupple corresponds to a role whom meaning depends of relation type $\mathit{type}(r)$. u_r denotes the confidence degree to which concepts of tupple r satisfy $\mathit{type}(r)$ in the document and w_r is the relevance of this relation in the document.

For matching process, we compute a matching degree Δ between concepts and relations like in [16], but we instantiate this degree in a different way. We define Δ the degree of match between a concept $c = [t : x|v, w_c]$ of query graph and a concept $c' = [t' : x'|v', w'_c]$ of the document graph as:

$$\Delta\left(c,c'\right)=\begin{cases}v\times v'\times w_c\times w'_c \text{ if } t'\leq t \text{ in } Tc\\ 0 \text{ otherwise}\end{cases}$$

The degree of match between a conceptual relation $\alpha = (type(r)|u_r, w_r)$ of query graph and a relation $\alpha' = (type(r')|u_{r'}, w_{r'})$ of the document graph is:

$$\Delta\left(\alpha,\alpha'\right)=\begin{cases}u_r\times u_{r'}\times w_r\times w_{r'} \text{ if } type(r')\leq type(r) \text{ in } Tr\\ 0 \text{ otherwise}\end{cases}$$

As a result, the projection of a graph $G(C,R)$ on a graph $G'(C',R')$ is a mapping π such that:

- for each concept c in C, $\pi(c)$ is a concept in C' and $\Delta(c,\pi(c))>0$,
- for each relation α in R, $\pi(\alpha)$ is a relation in R', $\Delta\left(\alpha,\pi(\alpha)\right)>0$ and if the i-th arc of α is linked to a concept c in G the i-th arc of $\pi(\alpha)$ is linked to $\pi(c)$ in G'.

Most of time the complete projection does not exist, as in [16], we consider that we can calculate the matching degree of a query subgraph on the document. For a complete or a subgraph projection we calculate the degree of match between a graph $G(C,R)$ and a graph $G'(C',R')$ by maximising $\Delta(G,G')$:

$$\Delta(G,G')=max_{\pi(G,G')}(\sum_{c\in C}\Delta(c,\pi(c))+\sum_{\alpha\in R}\Delta\left(\alpha,\pi(\alpha)\right))$$

4 Model Implementation

Computing a FCG from text implies the use of a domain knowledge resource. This resource is built by domain specialist and incorporate all useful terms and terms variations of this domain (terminology) so that each term is be properly associated to at least one concept. Links between concepts describe possible semantic relations (e.g. ‘localization of (emphysema ,CT images)’). For automatically building a FCG index (like on Fig. 4), we simplify the problem using two main hypotheses:

1. Document is indexed by a unique FCG.
2. A concept type can appears only once in a document index. As a consequence, if a concept is detected in two sentences of a document, we consider that these two instances have the same referent. As consequence, the generic referent * is not used.

Extracting and using a partial order on concepts and relations from resources is computationally time consuming. For this reason, currently we do not use such order. The graph projection is therefore reduces to the detection of the intersection between the query and the document graph.

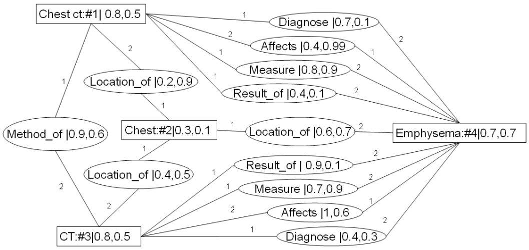


Fig. 1. FCG produced for *Show me chest CT images with emphysema*

4.1 Relation Confidence Score

Syntactic paths has already been used to discover variation of semantic relation for question answering for example in [17]. Compared with our work, we do not infer possible conceptual relation, but we rather compute a confidence score on only possible conceptual relations according to the knowledge base. In the following, we detail this computation.

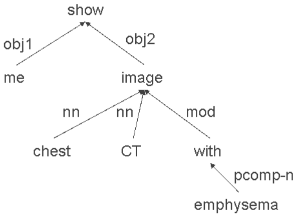
In order to automatically detect conceptual relations we need to set-up a leaning process based on a corpus where conceptual relations are known.

At first, concepts are detected from text and the concept extraction tool associates one concept to a *sequence* of syntactic elements. The shallow parsing produces a tree (see fig 4.1). So concepts detected are associated to a subset of the nodes of the syntactic tree. For example a concept could be associated to the three nodes {chest, CT, emphysema}.

Our idea is to detect semantic relations between too concepts by using the actual syntactic relations provided by the parser. We call this relation a *syntactic path*. First we have to assume that two concepts are associates to syntactic elements of the *same* syntactic tree.[2] Then the syntactic path is the unique graph path between the two heads of the associated terms. We do not take in consideration the orientation of the arcs, and as we are dealing with a tree this path is unique.

Let's finally define how we select the head from the subset of nodes: as we trust the parser, we simply take the node of the set that is the closest to the tree root. In the example of fig. 4.1, if a concept c is associated to the set {*chest, CT, image*}, after concept detection from the text "chest CT images", then the head is the node "image". In case of equal distant nodes, we select the far right node, because in English, head of noun phrases tends to be at the end (i.e. right) of a noun phrase. For example for a concept c_1 associated to {*chest,CT*}, the head is "CT". For a concept c_2 associated to "emphysema", the syntactic path between c_1 and c_2 is from the node "CT" to "emphysema", and is "*(nn)image(mod)with(pcomp-n)*". In the case of two concepts having the same head, we use a the pseudo-relation type (*ident*).

[2] In other case we force a constant value, see 5.



relation	path
(T135, C0817096, C0034067) *(location_of, chest, emphysema)*	(nn)image(mod) with(pcomp-n)
(T135,C0202823,C0040405) *(location_of, chest, CT)*	(nn)image(nn)

Fig. 2. Minipar[18] Syntaxtic tree for *Show me chest CT images with emphysema.* and corresponding syntactic path for two relations

We assume that the probability of a conceptual relation e between two concepts anchored on texts syntactic elements s_0 and s_{n+1} can be estimated considering the syntactic path $p = s_1, ..., s_n$ between the two anchors. Our idea is to use a *path model* for each possible semantic relation. A path $p = s_1, .., s_n$ is possibly associated to the semantic relation e if its probability to be generated by a relation model M_e is superior to its probability to be generated by a global relation model M_{Tr}. We propose the use of a unigram language model, so we do not take into account path item order. So, we model each relation type e with a unigram language model M_e on its path. Let's consider two concepts c, c' anchored to a path p, and a possible conceptual relation r between these two concepts: the probability of the concept type of r to be e can be compute as:

$$P(type(r) = e|p) = P(p|M_e)$$

Where M_e denotes the model of relation type e. Unigram model make the hypothesis of probability independence of model items. Hence, this probability is computed by:

$$P(p|M_e) = \prod_{s \in p} P(s|M_e)$$

We estimate each probability of syntax item s knowing the model of the conceptual type e, by computing the ratio of occurrence of $N(s, e)$, the frequency s appearing in a relation r' with $e = type(r')$, and $N(*, e)$ the total frequency:

$$P(p|M_e) = \prod_{s \in p} \frac{N(s, e)}{N(*, e)}$$

Finally, we compute the confidence degree of a relation as follow:

$$\text{conf}(p, e) = \frac{P(p|M_e)}{P(p|M_{Tr})}$$

where $e \in Tr$ and $P(p|M_{Tr}) = \prod_{s \in p} \frac{N(s)}{N(*)}$ is the probability that p is generated by any relation. Finally, it is this confidence degree that is used for the fuzzy value of a conceptual relation $(type(r) = e|u_r, w_r)$ among two concepts in-between the path p: $u_r = \text{conf}(p, e)$.

5 Experimental Results

We need to know if using a FCG index is better than using concept or relation indexes. We also need to demonstrate the usefulness of using a confidence score with a relevance score. We apply our model on medical text, since wide knowledge bases are available for this domain. We use UMLS, a large[3] medical meta-thesaurus result of the fusion of many resources. It includes concepts hierarchies and also a semantic network that defines high level relations between concepts.

5.1 FCG Generation

Graphs are produced in two steps: concept detection and then relation detection. For concept detection, we detect in UML all concepts that have a textual instance in the document. We show in [19] that such a strategy is better than extracting only precise concepts and can challenge or improve text based IR. Concept detection uses a syntactic analysis of sentences provided by MiniPar (figure 4.1) and a term mapping. To improve term mapping, we carried out some filtering on word and/or on UMLS. After concept detection, we add the conceptual relations. The relations used are those defined in the semantic network. We made the hypothesis that a relation exists in a semantic graph if two concepts are detected in the same sentence and if a relation between these concepts is defined in the semantic network.

At last, relation confidence scores are computed. As we do not have a learning corpus, we use all relations assigned on the collection. Thus, the best scores for a relation type will be assigned to relations that have a path composed with the most frequent words and the most frequent syntactic relations on the collection for this relation type. We compute the confidence of a relation on a document as the sum of confidence of each relation detected in a document. In our current experiments we do not use confidence score for concepts and for relation that have no syntactic path (the score is set to 1).

Detail on concepts and relations extraction (table 1) shows that an average of 4 concepts is detected by sentence, this number is quite low probably because some annotations are not in a well-formed English, for exemple a sentence can be very short (e.g. one word).

5.2 Protocol

We assess our model on ImageCLEFmed 2005[4] a multilingual images retrieval test collection. Most of textual descriptions are in English but some are in French and in German. Due to the parser, we can only use the textual English part of it (see ex: fig. 5.2).

The English part of the collection contains 40708 textual descriptions that describe more than 69% of the collection. We assess our results with two parallel

[3] More than 1 million concepts corresponding to more than 5 million terms.

[4] Part of "Cross Language Evaluation Forum" (CLEF) http://ir.shef.ac.uk/imageclef/

Table 1. Corpus informations

	CLEF	TXT_ENG
Detect concepts	388842	262983
Distinct concepts	7352	7352
average concept by sentence	4.4	4.2
Detect relations	3098306	2168794
Distinct relations	342592	342592
average relations by sentence	35	34.9

```
<GlobalID>1112</GlobalID> <Title>RESPIRATORY</Title>
<Description>RESPIRATORY: Lung: Arteriosclerosis Grade 3:
 Micro low mag H&E same as in slide  grade 3 lesion with dilated
 appearing arteriole distal to the small artery lesion </Description>
...
```

Fig. 3. Annotation example

evaluations. On the one hand, we evaluate our results as done in official CLEF track (named 'CLEF'), on the other hand, we only use English descriptions (named here 'TXT_ENG'). In that case, we obtain 33513 textual descriptions with ground truth for the 25 queries.

After presenting the baseline results, we compare three indexes; one that contains only concepts, a second that contains relations and a last one that contains FCG. We assess FCG indexes with or without confidence and with different IR weighting (w_r) variations:

- (occ) is concept (relation) frequency in a document,
- (tf) is log term frequency define by $log\,(n) - 1$,
- (idf) is inverse document frequency define by $log\left(\frac{N}{n_i}\right)$,
- (tf.idf) that use both term frequency and inverse document frequency

We evaluate IR results with the mean average precision (MAP) as it gives a global overview of results and the precision at 5 documents (P@5), as this measure shows system precision on first results which are the relevant ones for precise IR.

5.3 Baseline Experiment

Our baseline method use a vector space indexation based on lemma. This method, described in [20], extracts lemma and uses a filtering on POS tags that keeps only nouns, adjectives and abbreviations. On the three language of ImageCLEFmed this method provides a MAP of 17.25% with tf.idf. Table 2 presents the results of this method on the English part of the collection. By indexing only the English part of the collection, lemma results are lower of 4%. In view of the difference, this part of the collection is sufficient to be used alone in experimentation. The *occ* weighting gives the best results, but all the results remain close. Moreover results are low, and using only lemma seems to be not enough for solving the queries.

Table 2. Stem Indexing

	CLEF		TXT_ENG	
	MAP	P@5	MAP	P@5
Tf.Idf	0.154	0.376	0.181	0.376
Idf	0.157	0.376	0.183	0.360
Tf	0.166	0.368	0.192	0.368
occ	0.166	0.368	0.194	0.368

5.4 Graph Matching

After conceptual graph production, we compare concept index (Table 3), relation index (Table 4) and FCG index (Table 5). Results show that concept indexes perform better relation ones. With relation, only 20 queries are solved on the 25. These 5 queries have no relation or have relations that are not found in CLEF collection. Relations alone are too selective; few documents have all query relations and as concepts of the relation are not detected, the recall decrease.

Concepts indexations MAP are close from lemmas ones in the CLEF evaluation but gives better results in the TXT_ENG evaluation. Concepts are better than lemmas for detecting textual description. FCG mean average precision is higher than concepts or relations one, regardless to the weight. On the CLEF

Table 3. Concept results

	CLEF		TXT_ENG	
	MAP	P@5	MAP	P@5
Tf-Idf	0.146	0.312	0.212	0.336
Idf	0.151	0.336	0.220	0.36
Tf	0.165	0.384	0.236	0.4
occ	*0.167*	0.408	**0.246**	**0.424**

Table 4. Relation results

	CLEF		TXT_ENG	
	MAP	P@5	MAP	P@5
Tf-Idf	0.123	0.34	0.205	0.32
Idf	**0.124**	**0.36**	**0.206**	**0.34**
Tf	0.123	0.34	**0.206**	0.33
occ	**0.124**	**0.36**	**0.206**	**0.34**

Table 5. Fuzzy graph results

concept	relation	CLEF		TXT_ENG	
		MAP	P@5	MAP	P@5
ooc	Tf	0.170	0.4	0.254	0.384
	Idf	0.171	0.4	0.255	0.384
	occ	0.171	0.376	0.255	0.392
	Tf×confidence	0.173	0.416	0.261	0.448
	Idf×confidence	0.172	0.424	0.261	**0.456**
	Occ×confidence	0.172	0.424	0.26	0.448
Tf	Tf	0.17	0.392	0.254	0.384
	Idf	0.171	0.4	0.255	0.392
	occ	0.172	0.4	0.256	0.392
	Tf×confidence	**0.174**	0.4160	**0.264**	0.448
	Idf×confidence	0.173	0.424	0.262	0.448
	Occ×confidence	0.172	0.424	0.262	0.448

evaluation the best FCG index is 4% better than the best concept one and 4.7% better than lemma one. But on the TXT_ENG evaluation, FCG index is 7% better than concept one and around 36% better than lemma one. As a consequence using the FCG index is better than using only a part of it (concept or relation). We also note that P@5 is improved with FCG index, but only when confidence is used (4% on CLEF and 7.5% on TXT_ENG). This is not surprising, confidence permits to select well extracted relation when using only IR score does not distinguish such relation. Using a IR score and an confidence score is valuable for precise IR.

6 Conclusion

Despite the fact that most of IR systems still use the bag of word paradigm we believe that major breakthrough in IR will come from the use of large knowledge sets producing a conceptual indexing. For this reason we have proposed here a model based on fuzzy conceptual graph index with fully automatic graph construction. We chose a incomplete graph construction so to keep most of concepts and relations from the knowledge base. We integrate a confidence score that reflects NLP incompleteness. Experimental results on ImageCLEFmed show that FCG give good performances and that using a confidence score improves results, mainly P@5 which is very useful to build a precision oriented IR system. Confidence in concept detection still need to be experimented also the effect of a better learning corpus with annotated relations.

References

[1] Schank, R.C., Kolodner, J.L., DeJong, G.: Conceptual information retrieval. In: Proceedings of the 3rd annual conference ACM SIGIR, Kent, UK, 94–116 (1980)

[2] Meghini, C., Sebastiani, F., Straccia, U., Thanos, C.: A model of information retrieval based on a terminological logic. In: Proceedings of the 16th annual international ACM SIGIR, pp. 298–307. ACM Press, New York (1993)

[3] Berrut, C., Chiaramella, Y.: Indexing medical reports in a multimedia environment: the rime experimental approach. In: Proceedings of the 12th annual international ACM SIGIR, pp. 187–197. ACM Press, New York (1989)

[4] Sowa, J.F.: Conceptual Structures: Information Processing in Mind and Machine. Addison-Wesley, Reading (1984)

[5] Aït-Mokhtar, S., Chanod, J.P., Roux, C.: Robustness beyond shallowness: incremental deep parsing. Nat. Lang. Eng. 8, 121–144 (2002)

[6] Strzalkowski, T., Stein, G.C., Wise, G.B., Carballo, J.P., Tapanainen, P., Jarvinen, T., Voutilainen, A., Karlgren, J.: Natural language information retrieval: TREC-7 report. In: Text REtrieval Conference, pp. 164–173 (1998)

[7] Zhai, C., Tong, X., Milic-Frayling, N., Evans, D.: Evaluation of syntactic phrase indexing - clarit nlp track report (1997)

[8] Matsumura, A., Takasu, A., Adachi, J.: The effect of information retrieval method using dependency relationship between words. In: Proceedings of the RIAO 2000 Conference pp. 1043–1058 (2000)

[9] Metzler, D.P., Haas, S.W.: The constituent object parser: syntactic structure matching for information retrieval. ACM Trans. Inf. Syst. 7, 292–316 (1989)

[10] Smeaton, A.F.: Using NLP or NLP resources for information retrieval tasks. In: Strzalkowski, T. (ed.) Natural language information retrieval, pp. 99–111. Kluwer Academic Publishers, Dordrecht, NL (1999)

[11] Vintar, S., Buitelaar, P.V.M.: Relations in concept-based cross-language medical information retrieval. In: Proceedings of the ECML/PKDD Workshop on Adaptive Text Extraction and Mining (ATEM) (2003)

[12] Benigno, M.K.D., Cross, G.R.: deBessonet, C.: Corel - a conceptual retrieval system. In: Proceedings of the 9th Annual International ACM SIGIR, ACM, pp. 144–148 (September 1986)

[13] Genest, D., Chein, M.: A content-search information retrieval process based on conceptual graphs. Knowl. Inf. Syst. 8(3), 292–309 (2005)

[14] Ounis, I., Chevallet, J.P.: Using conceptual graphs in a multifaceted logical model for information retrieval. In: The 7th Database and EXpert system Applications Conference, DEXA'96, Zurich, Switzerland, pp. 812–823 (September 1996)

[15] Wuwongse, V., Manzano, M.: Fuzzy conceptual graphs. In: Proceedings on Conceptual Graphs for Knowledge Representation (ICCS'93), London, UK, pp. 430–449 (1993)

[16] Mulhem, P., Leow, W.K., Lee, Y.K.: Fuzzy conceptual graphs for matching images of natural scenes. In: IJCAI pp. 1397–1404 (2001)

[17] Lin, D., Pantel, P.: Discovery of inference rules for question-answering. Natural Language Engineering 7(04), 343–360 (2001)

[18] Lin, D.: Dependency-based evaluation of minipar. In: Workshop on the Evaluation of Parsing Systems, Granada, Spain, May, ACM (1998)

[19] Radhouani, S., Maisonnasse, L., Lim, J.H., Le, T.H.D., Chevallet, J.P.: Une indexation conceptuelle pour un filtrage par dimensions, experimentation sur la base medicale imageclefmed avec le meta thesaurus umls. In: COnference en Recherche Information et Applications CORIA' 2006 pp. 257–271 (2006)

[20] Chevallet, J.P., Lim, J.H., Radhouani, S.: Using ontology dimensions and negative expansion to solve precise queries in clef medical task. In: CLEF Workhop, Working Notes Medical Image Track, Vienna, Austria (September 21-23, 2005)

Combining Vector Space Model and Multi Word Term Extraction for Semantic Query Expansion

Eric SanJuan[1], Fidelia Ibekwe-SanJuan[2], Juan-Manuel Torres-Moreno[1,2], and Patricia Velázquez-Morales

[1] Laboratoire Informatique d'Avignon UAPV, BP 1228 84911 Avignon, Cedex 9, France
{eric.sanjuan,juan-manuel.torres}@univ-avignon.fr
[2] ELICO, Université de Lyon 3. 4 cours Albert Thomas, 69008 Lyon Cedex, France
ibekwe@univ-lyon3.fr
[3] École Polytechnique/DGI CP 6079 Succ. Centre-ville - H3C3A7 Montréal, Canada

Abstract. In this paper, we target document ranking in a highly technical field with the aim to approximate a ranking that is obtained through an existing ontology (knowledge structure). We test and combine symbolic and vector space models (VSM). Our symbolic approach relies on shallow NLP and on internal linguistic relations between Multi-Word Terms (MWTs). Documents are ranked based on different semantic relations they share with the query terms, either directly or indirectly after clustering the MWTs using the identified lexico-semantic relations. The VSM approach consisted in ranking documents with different functions ranging from the classical tf.idf to more elaborate similarity functions. Results shows that the ranking obtained by the symbolic approach performs better on most queries than the vector space model. However, the ranking obtained by combining both approaches outperforms by a wide margin the results obtained by methods from each approach.

1 Introduction

Despite the huge amount of studies on query expansion and document ranking, this topic continues to attract a lot of attention. Indeed, earlier studies have established that information seekers rarely use the enhanced search features available on most search engines or in specialised databases. Average query text consists of 1.8 words [1]. This means that query terms are often too imprecise. In technical fields, it can be expected that a unique semantic category can be associated to each domain term (a noun phrase that refers to a unique concept in some specialised field). When an ontology exists, refining by semantic nearest-neighbour term consists in expanding the query terms using terms in the same category as the query. When the query is too imprecise, this process of refinement by adjoining semantically related terms allows to rank documents according to the frequency of such terms in titles or abstracts available in bibliographic databases.

We target document ranking in a highly technical field with the aim to approximate a ranking that is obtained through an existing ontology or a knowledge

Z. Kedad et al.(Eds.): NLDB 2007, LNCS 4592, pp. 252–263, 2007.

structure. The reference ranking is obtained by refining the query term with terms in the same semantic category in the ontology. For instance, the topmost documents by the reference ranking for the query term *T-cell* contained its variants from the same semantic category (*jurkat T cell, L cell, T cell, activated T cell, cell, endothelial cell, human T, human T cell,...*).

We tested two ranking approaches to query expansion: symbolic and the vector-space model. Both approaches will try to obtain rankings that come as near as possible to the reference ranking but without knowledge of the manually annotated terms nor of the semantic category of a term in the ontology. The vector space model uses several functions for measuring *term – document* similarity and is implemented using the CORTEX system. This system was initially designed for automatic summarisation [2]. The symbolic approach relies on surface linguistic relations between query terms and documents and is implemented via the TermWatch system [3]. This system extracts multi-word terms (MWTs), links them through local morphological, lexical, syntactic and semantic relations, then clusters the MWTs variants based on these relations. Given a query term, these clusters are used to rank documents according to the proportion of shared terms between clusters and documents that also contain the query term. The idea is to refine a query term with its semantic nearest neighbour (S-NN) terms. For instance, for the query term "*T cell*" query, the topmost document chosen by TermWatch will ideally have terms bearing the head word *cell* in their grammatical head position as in *cell, jurkat cell, naive cell, responding cell, stimulated cell, T cell.*

Finally, in a third hybrid approach, relations used for ranking in the symbolic approach are combined with different functions from the vector-space model in order to see if this improves the results. All the methods are then evaluated against a reference ranking obtained by the reference ranking. As a by product, this experiment also provides a new methodology for comparing the different methods issuing from the two approaches.

The rest of the paper is organised as follows: section 2 describes the corpus used in this experiment and how queries were formulated. Section 3.1 describes the symbolic approach, the following section 3.2 describes the vector space model approach and section 3.3 the hybrid approach. Section 4 analyses results while section 5 draws lessons learned from the experiment.

2 The Test Corpus

A pre-requisite for this experiment is that domain terms in the test corpus be previously annotated and assigned a unique semantic category in the ontology. Thus, we needed a corpus with an associated knowledge structure (taxonomy or ontology) where each term in the corpus can be traced back to the ontology. This is because our QR systems extract terms automatically from the corpus and the associated knowledge structure is used to build the reference ranking. The GENIA corpus[1] satisfied our requirements in that it comes with a hand-built

[1] http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/

ontology where terms from the abstracts have been manually annotated and assigned to categories in the ontology by domain specialists. This corpus consists of 2000 bibliographic records drawn from the MEDLINE database using the keywords: *Human, Blood Cells,* and *Transcription Factors*. We shall refer to the titles and abstracts of these records as documents henceforth. The annotations in XML format indicate the terms together with their semantic category, defined as the leaves of a small hand-built ontology, the GENIA ontology. There are 36 such categories at the leaf nodes and a total of $31,398$ terms. The largest category, called "*other name*" has $10,505$ terms followed by the "*protein molecule*" category with $3,899$ terms and the "*dna domain or region*" category with $3,677$ terms. The distribution of terms in the categories follow a zipfian curve. In this context, each annotated term can be viewed as a potential query that will extract all documents in the GENIA corpus containing this term or semantically close terms in the same GENIA category (in the ontology). The extracted documents can therefore be ranked according to the number of annotated terms in the same GENIA category as the term query. The ranking obtained for each query using the manually annotated terms and the GENIA categories constitutes the reference ranking. The QR experiment thus consists in testing the ability of different methods from the two approaches to produce a ranking as similar as possible to the reference ranking. Of course, none of the QR methods tested used the manually annotated terms nor had prior knowledge of their semantic category in the GENIA ontology.

The query terms used in this experiment were manually annotated terms in the GENIA corpus that occurred in at least 50 documents and which were associated with a category other than "*other name*". We also excluded one word terms like "*cell*". In the GENIA corpus, this term will select practically all the documents. Sixteen MWTs matched these criteria. Table 1 shows the query terms together with their GENIA category, the number of elements in this category and the number of documents containing each term. We now describe the two approaches to the QR task.

3 Methodology

3.1 Symbolic Approach

This approach to QR is implemented via the TermWatch system [3] which comprises three modules: a term extractor, a relation identifier which yields the terminological network and a clustering module. Clustering is based on general linguistic relations that are not dependent on a particular domain and do not require specific work for every text collection.

Different linguistic relations for expanding query terms into their S-NN terms were tested, ranging from coarse-grained ones like identity of grammatical head words to fine-grained ones.Thus, any query term is mapped onto the set of automatically extracted S-NN terms. Since these S-NN terms have been grouped into clusters, the query term can be represented by the cluster vector with as many dimensions as there are clusters and, whose values are the number of

Table 1. Queries used in the experiment

Query	**GENIA Category**	**Nb**	**Docs**
activated T cell	cell_type	1723	51
B cell	cell_type	1723	120
Epstein-Barr virus	virus	352	66
glucocorticoid receptor	protein_family_or_group	2452	96
human immunodeficiency virus type 1	virus	352	52
human monocyte	cell_type	1723	69
Jurkat cell	cell_line	1992	66
Jurkat T cell	cell_line	1992	58
NF-kappa B	protein_molecule	3885	271
nuclear extract	cell_component	205	74
nuclear factor	protein_family_or_group	2452	54
nuclear factor of activated T cells	protein_family_or_group	2452	51
protein kinase C	protein_molecule	3885	83
T cell	cell_type	1723	339
T lymphocyte	cell_type	1723	115
transcription factor	protein_family_or_group	2452	487

variants that the query has in each cluster. Since every document can also be represented by a similar vector that gives for each cluster, the number of its terms in the document, the relevance of the document against the query can be evaluated as the scalar product between the two vectors (cluster and document). We describe in more details the linguistic relations used in ranking.

Ranking by head word occurrence (Head). This consists in ranking documents based on an occurrence count of the head word of the query term in the documents that contain that head word but in any grammatical position. The justification for using this coarse relation is the well-known role of head nouns in noun phrases: they depict the subject of phrases and thus also of the queries. Thus documents in which the head word has a high frequency could select documents with the highest number of terms in the same GENIA category. Document ranking with this relation is performed outside TermWatch as it relies simply on an occurrence count of a head word in documents.

Ranking by Basic TermWatch's clusters (TW). The most coarse-grained clustering relation in TermWatch consists in merging all terms sharing the same head word into the same cluster. This relation generated clusters of identical heads and on this corpus produced $3,670$ clusters involving all the extracted multiword terms $(36,702)$. Given a query term, documents are ranked according to the number of their terms which had the head word of the query term also in their head position.

For instance, given the query term *T cell* where *cell* is the head word, the topmost ranked document by this relation had the most number of terms with "cell" in its head position: *B cell, cell, blood cell, differentiated cell, hematopoietic* *cell, HL60 cell, L cell, lympoid cell, macrophage cell, monocyte-macrophage cell, nucleated cell, peripheral blood cell, S cell, T cell.*

Ranking by tight semantic clusters (Comp). This consists in ranking using terms in the connected components formed by spelling variants, substitutions of synonymous variants acquired via WordNet and expansions relations (where only one word was added to a term). The idea is to restrict the S-NN of a query term to only those terms which do not involve a topical shift and are its closest S-NN in terms of all the variation relations used in TermWatch. In this experiment, $2,382$ were found involving only $8,019$ terms.

Ranking by looser semantic clusters (Var). Relations are added to *Comp* ones in order to form bigger clusters involving weaker expansion variants (addition of more than one modifier word) and substitution of modifier words. The idea here is to expand the S-NN of a query term to farther semantic neighbours where the link with the original subject of the query term may be weaker. Clustering in this case produced $3,637$ clusters involving $14,551$ terms. For instance, for the same "*T cell*" query, the topmost document ranked by *Var* clusters had six terms bearing the word *cell* in their head position some of which were also modifier substitutions of the query term (*cell, jurkat cell, naive cell, responding cell, stimulated cell, T cell*). In contrast, the topmost document ranked by the reference ranking obtained through the GENIA ontology contained more variants of the query term (*jurkat T cell, L cell, T cell, activated T cell, cell, endothelial cell, human T, human T cell,...*). This document was ranked 10th by *Var* relations.

3.2 Vector-Based Model Approach

We tested two ways of ranking documents based on the vector model. The first method supposes that word frequency can be estimated on the whole set of documents represented as an inverted file. The second method works on the restricted set of documents containing at least one occurrence of the query term.

Let Δ be the set of all abstracts in the bibliographic database and let Ω bet the set of uniterms (terms with only one word). For any abstract d, we shall denote by Ω_d the set of uniterms occurring at least once in d and by Δ_w the set of documents in which w occurred.

We assume the existence of an inverted file which for any word w and abstract d in the bibliographic database gives the frequency $f_{d,w}$ of w in d. Based on such inverted file, documents can be ranked following the *tf.idf* score of query terms in the document with or without query expansion mechanism *QE*. It consists in first computing the *tf.idf* function and then replacing the query term vector by the sum of the top ranked document vectors. This expanded query is then used to perform another ranking.

Now, we do not more assume the existence of an inverted file. Given a query sequence T in the form of a MWT the following measures are computed on the restricted set of documents $\Delta(T)$ where the string T occurred. These documents are represented in a vector space [4,5] using the CORTEX [2] system that includes a set of independent metrics combined by a Decision Algorithm. This vector space representation takes into accounts nouns, compound words, conjugated verbs numbers (numeric and/or textual) and symbols. Other grammatical categories like articles, prepositions, adjectives and adverbs are eliminated using a stop list. Lemmatisation and stemming [6,7] are performed thus yielding higher word frequencies. Compound words are identified, then transformed into a unique lemmatised/stemmed uniterm using a dictionary.

To describe the selected metrics we used for QR, we shall use the following notations for any $w \in \Omega$ and $d \in \Delta(T)$:

$$\Delta(T)_w = \Delta_w \cap \Delta(T) \quad f_{d,.} = \textstyle\sum_{\omega \in \Omega_d} f_{d,\omega} \quad f_{.,w} = \displaystyle\sum_{\delta \in \Delta(T), w \in \Omega_\delta} f_{\delta,w}$$

$$\Omega(T) = \{\omega \in \Omega : f_{.,w} > 1\} \quad f_{.,.} = \textstyle\sum_{\omega \in \Omega(T)} f_{.,\omega} \quad \Omega(T)_d = \Omega_d \cap \Omega(T)$$

We tested the metrics described above as well as combinations of them: the angle (noted A), three different measures of query overlapping (D, L, O) and the frequency of informative words (F). We also considered the following combinations of sets of metrics $\{A, D, O\}$, $\{A, L, O\}$, $\{A, D, L, O\}$, $\{F, L, A, D, O\}$ based on CORTEX's decision algorithm.

A is the angle between T and d. Although not all words in T have the same informative value since words closed to the term head have an higher probability to be correlated to the term's category. Thus, we have represented the query term $T = t_1 ... t_n h$ by a vector $\boldsymbol{T} = (x_w)_{w \in \Omega(T)}$ where:

$$x_w = \begin{cases} 15 & \text{if } w = h \\ j & \text{if } w = t_i \text{ for some } i \in [1..n] \\ 0 & \text{otherwise} \end{cases}$$

D is the sum of the word frequencies in abstract d multiplied by its probability of occurrence in $\Delta(T)$ as follows: $D(d) = \sum_{w \in \Omega(T)_d} \left(\frac{f_{.,w}}{f_{.,.}} \times f_{d,w} \right)$

O focus on documents involving terms that occurred in almost all documents: $O(d) = \sum_{w \in \Omega(T)_d} (|\Delta(T)_w| \times f_{d,w})$

L reveals documents that overlap with query words but with a larger vocabulary: $L(d) = |\Omega(T)_d| \times \sum_{w \in \Omega(T)_d} (|\Delta(T)_w|)$

F is the term frequency sum $F = f(., w)$ It favours documents with a small vocabulary on tha contrary of metrics D,O,L.

The Decision Algorithm (DA) relies on all the normalised metrics $\hat{\mu}(d)$ combined in a sophisticated way. Here is the decision algorithm that allows to include the vote of each metrics:

$$\alpha = \sum_{\hat{\mu} \in \{X_1, \ldots, X_k\}, \hat{\mu}(d) > 0.5} (\hat{\mu}(d) - 0.5) \; ; \; \beta = \sum_{\hat{\mu} \in \{X_1, \ldots, X_k\}, \hat{\mu}(d) < 0.5} (0.5 - \hat{\mu}(d)) (1)$$

The value Λ attributed to every sentence is then calculated:

$$\text{If } \alpha > \beta \text{ then } \Lambda = 0.5 + \tfrac{\alpha}{k} \text{ else } \Lambda = 0.5 - \frac{\beta}{k}$$

3.3 Hybrid Approach

Clusters built by TermWatch target a high degree of semantic homogeneity. They rely on the existence of a restricted family of linguistic variation relations among terms and thus are generally small in size. As a consequence, when mapping a query term T onto its S-NN terms in clusters, this often grasps only a few clusters. Thus, ranking documents according to their overlap with these clusters produces a substantial proportion of ties. We then tried to use CORTEX's normalised metrics to break these ties. Indeed as pointed out in the preceding section, high scores of selected CORTEX metrics are obtained for documents containing the query words in T and words frequently associated to them, i.e, their co-occurrence contexts. Since document scores based on cluster overlapping are integers, tails can be simply broken by adding to this integer score, CORTEX's decision score which is a real number in $[0, 1]$. This leads to a new document ranking system (summarised in figure 1) where documents are:

1. extracted in full text Boolean mode based on a sentence expressed in natural language,
2. ranked according to the linguistical relations they share with the multiword terms in the query,
3. re-ranked by breaking ties based on vector similarities with the query.

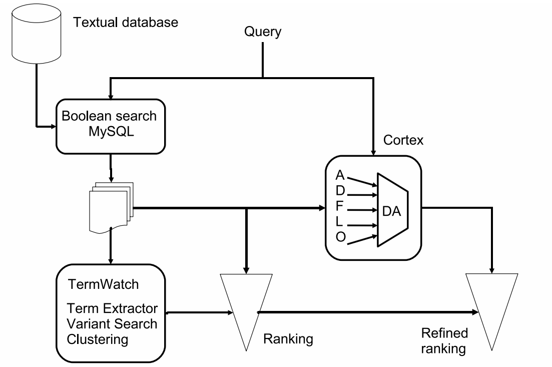


Fig. 1. Hybrid ranking system

4 Results

We now analyse results from the three approaches : vector space, symbolic and hybrid. Given a query term, we evaluate the methods described in sections 3.1 and 3.2 according to their capacity in ranking documents with regard to an existing ontology, i.e., top ranked documents should contain terms from the semantic category in the GENIA ontology as the query term.

For each query, we compared the ranking of documents produced by the different methods to the reference ranking by calculating the Kendall's W coefficient of concordance [8]. This coefficient stems from the family of robust non-parametric tests which do not make any assumption on the Gaussian distribution of the data. Kendall's W coefficient is 1 in the case of complete agreement between two rankings and 0 for total disagreement. As in all statistical tests, to interpret the intermediary values, it is necessary to verify if the score obtained by a method is significantly different from that of a random ranking on the same data. We computed Kendall's W coefficient and its "p-value" using R software for statistical computing with the Concord package[2]. We did not use precision-recall as evaluation metric because all the ranking methods work from the same list of documents, i.e., they are all based on the selection of documents containing the initial query term. What differed was the way in which they ranked these documents. Hence, calculating recall does not make sense here.

4.1 Global Comparison of Methods

Figure 2 gives the boxplots of Kendall's W coefficient of concordance on all queries for each method. According to these boxplots, refining TW's ranking by CORTEX's metrics ($X_1 ... X_k$-tw where $X_1, ..., X_k$ is any combination of {A, D, F, O, L}) outperformed single TW which in turn outperformed the Head method, any one of CORTEX metrics (A, D, F, O, L) taken separately or any of their combinations and MySQL rankings (tf.idf and QE). We now check if these differences are statistically significant. For that, we apply the non parametric paired Wilcoxon signed rank test and Friedman's rank sum test both available in the standard R software package. These two tests are used to compare the median Kendall's W scores obtained by each method.

We first analysed the combinations of CORTEX's metrics to see if any one performs better than the others. Friedman's test showed with a confidence of 99% that there exists significative differences. However, running the same test only on combination of at least two CORTEX measures among {A, D, O, L} shows that there is no statistical evidence of differences among members of this group (p-value > 0.8). This shows that combining CORTEX metrics based on its decision algorithm 3.2 significantly improves the results.

Now observing the group of methods based on a single CORTEX metric significantly differs among themselves as found by Friedman's test with a confidence of 99%. Indeed, based on Wilcoxon test we found out that O and D are not statistically different (p-value=0.86), neither are F and L (p-value=0.82). The first two appear to be more adapted to this experiment than F and L (see their Kendall's W values on Figure 2). Metrics O and D top-rank documents in which the frequent words correspond to the query words or are strongly associated to them, whereas metrics L and F focus on the vocabulary coverage of documents irrespective of the query words. L is very sensitive to documents with a wide vocabulary coverage and F does the reverse. Thus these two rank documents based on criteria intrinsic to the documents but not to the query. Metric A that takes into account the position of each word in the query remains apart. Finally, we

[2] 'http://www.r-project.org/

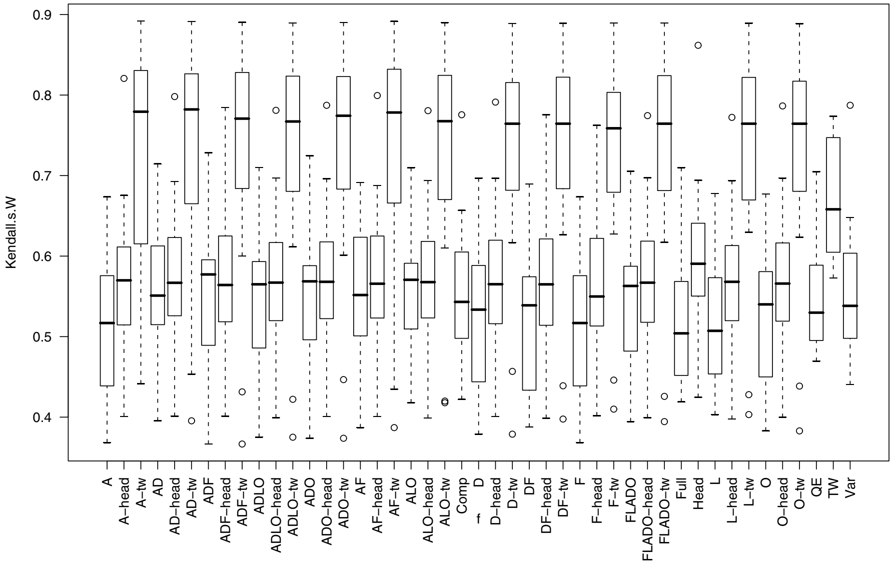


Fig. 2. Boxplots showing median Kendall's W scores and extreme values for each method. Symbols A, D, F, L, O and their combinations in upper case refer to CORTEX metrics (e.g. FLADO); "Head", "TW" and "Var" refer to the rankings based on the respective TermWatch's clusters. Symbols representing CORTEX's metrics followed by lower case "tw" or "head" refer to hybrid approaches. "QE" stands for *tf.idf with QE.*

take a look at performances amongst symbolic methods to see if there is any statistical difference among their rankings. Wilcoxon's test enabled us to ascertain that the hypothesis of equal medians between *basic TW* and Head's rankings can indeed be rejected with a risk lower than 5%. The same test also showed with a confidence of 90% that *Head* method outperformed *Var*, but that the observed differences between *Head* and *COMP*'s rankings were not statistically significant (p-value=0.23).

Let us now compare the rankings obtained by the hybrid approach. We have already observed that there is no statistical difference between median scores of combinations of at least two CORTEX's metrics. We have the same phenomena between any TermWatch's ranking refined by any CORTEX's metric. Indeed the p-value resulting from the Friedman test on this family of methods is higher than 0.54. Since we have already ascertained the effectiveness of CORTEX's decision algorithm, we shall only need to consider *FLADO-tw* which is the refinement of TW ranking based on the combination of all selected CORTEX metrics among all possible combinations. In the same way, we found out that there is no statistical evidence of differences between refinements of Head's rankings with any CORTEX's metrics. Thus we shall only consider the *FLADO-Head* combination. We then obtain, based on Wilcoxon's test, that *FLADO-tw* outperforms *TW* with a confidence of 95%, and that *Head* outperforms *FLADO-head* with a confidence of 99%. Since we have previously shown that *TW* outperforms *Head*, we deduce that *FLADO-tw* clearly outperforms *FLADO-head* and *FLADO*. This turned out to be the case with a confidence level higher than 99.98%.

Following these statistical tests, it appears using that the combination of CORTEX's metrics (FLADO) chosen by its decision algorithm to refine TermWatch's TW's semantic rankings produces the best hybrid approach. Contrarily, refining the ranking produced by the *Head* method with CORTEX's metrics degrades results considerably.

4.2 Query by Query Comparison of Ranking Methods

Global results can mask important differences as suggested by the length of the boxes in figure 2 and by the existence of extreme values. The detailed view of the performances for the main methods is shown in Table 2. This table shows the Kendall's W score for each method per query. For each query, only the relative position of the score between methods can be directly interpreted. Thus, Table 2 can only be read vertically, column by column. Indeed, Kendall's score depends on the number of ranked documents and on the number of tails. The absolute Kendall's W value cannot be interpreted without considering the probability of finding this value in non correlated rankings. The confidence level is the complement of this probability. Table 2 only shows figures with a confidence level of at least 90%. It evaluates the expectation of the correlation between the ranking produced by the methods and the reference ranking.

Table 2 shows that *FLADO-tw* is the only method that produced 14 rankings out of 16 with more than 90% probability of being correlated with the reference ranking. The two non correlated ranking were produced for the longest queries "*nuclear factor of activated T cells*" involving a preposition and "*human immunodeficiency virus type 1*". We will comment on this later.

It also appears clearly that *FLADO-tw* improves *TW* on all queries, thus showing that CORTEX is adapted to resolving ties in *TW*'s rankings. Conversely, a similar combination of metrics degrades Head's ranking, whereas the two methods *TW* and *Head* considered separately obtain similiar Kendall's W scores on several queries where the category is mainly determined by the head word. If we look at CORTEX's metrics in isolation, we obtain weaker results than for *Head* and *TW* methods. However it is interesting to observe that the three measures A, D and O are required in order to cover the whole set of queries where the FLADO combination is significant. It is also interesting to notice that *Comp* method based on tight semantic relations performed well mainly on queries where no CORTEX metric obtained good scores like "*nuclear factor, T lymphocyte, activated T cell*". This points to the fact that a hybrid approach is indeed desirable for query expansion and the two systems TermWatch and CORTEX are indeed complementary for this task.

We now take a look at queries where the hybrid approach did not perform as well as expected, i.e., where independent methods obtained better rankings. The *Head* method significantly outperformed all other methods on the "*Epstein-Barr virus*" query due to the fact that the head word "*virus*" characterises the terms in this GENIA category, i.e., almost all terms in this category include the word "*virus*". Thus counting the occurrences of this head word in documents is equivalent to counting occurrences of terms in this category. There is however a difference between the ranking produced by *Head* and the reference ranking because the latter records the single presence of a term in a document even if the term has multiple occurrences.

Tf.idf function is the only one that obtained a significantly correlated ranking on the query "*human immunodeficiency virus type 1*" notwithstanding the ambiguity of the subject of this query, which is not the last token *1* but the entire phrase *virus type 1.* One query was not included in the table ("*nuclear factor of activated T cells*") because no method attained the confidence level of 90% on it. This query had the particularity of containing a preposition. Permutation variants are amongst those identified by TermWatch and could be used in future work to efficiently process queries with prepositions.

Table 2. Kendall's W scores per query. Only scores with a confidence level of at least 90% appear. Figures with confidence between 90% and 95% are in italic. Figures in Bold have a confidence greater than 99%.

Queries:	B cell	protein kinase C	T cell	NF-kappa B	Jurkat cell	transcription factor	T lymphocyte	Epstein-Barr virus	nuclear extract	glucocorticoid receptor	human monocyte	nuclear factor	Jurkat T cell	activated T cell	human immunodeficiency virus type 1
Head	**0.61**	**0.69**	**0.58**		0.68	**0.60**		**0.86**	0.65		*0.59*	0.58	*0.62*		
FLADO-head	0.58	*0.62*	**0.60**		0.69	**0.58**		**0.77**	*0.61*				*0.63*		
A		0.63		**0.67**	0.65					*0.59*					
D	0.58	0.63	**0.57**	**0.62**	0.69										
F		0.63		**0.67**	0.65					*0.59*					
L		*0.62*		**0.67**	0.65	0.56									
O	*0.56*	0.63	0.55	**0.65**	0.67	0.55									
FLADO	*0.57*	*0.61*	**0.57**	**0.63**	**0.70**	**0.57**									
Comp			0.57				*0.61*					*0.65*		**0.77**	
Var			**0.60**		**0.78**							*0.63*	*0.64*		
TW	**0.74**		**0.72**	0.58	**0.77**	**0.60**	0.65	**0.75**	*0.63*	*0.60*	**0.75**		0.67	0.75	
FLADO-tw	**0.88**	0.67	**0.88**	**0.61**	**0.88**	**0.73**	**0.80**	**0.73**	**0.84**	**0.73**	0.68	**0.75**	**0.80**	**0.77**	
QE		0.65		**0.64**	**0.70**	0.54							*0.61*		
tf.idf								0.68					**0.70**		0.70

5 Conclusion

The task introduced in this paper that we have termed *Semantic Query Expansion oriented Document Ranking* (SQEDR) is quite novel and has not been dealt with in the TREC's campaigns [9]. The results we obtained show on the GENIA corpus that such rankings can be approximated combining MWT term extraction and bag-of-word text representation.

In the recent TREC2005 Robust track, [10] used WSD (word sense disambiguation) and semantic query term expansion in the document retrieval task. WSD is first applied to multi-word query terms in order to determine the exact sense of the constituent words in the context of the query. This is done using all the available information in WordNet. When this fails, the authors resort to a Web search for the WSD process. After WSD is performed, semantically-associated terms to the chosen sense (synset) from WordNet are used to expand the query term. As we can see, query expansion here is heavily reliant on WordNet's coverage of words in the document collection.

Work in progress is carried out in testing if SQEDR could be usefull in this TREC's standard task.

We are also working in drawing records from general MEDLINE corpus. SQEDR can be carried out on this corpus using Mesh thesaurus[3] and the UMLS[4]. However, these two contain only terms from a controlled vocabulary (humanly fabricated terms) which are not necessarily present in MEDLINE's abstracts. Our approach of SQEDR could handle this gap between real terms from texts and terms from a controlled vocabulary.

References

1. Ray, E.J., Seltzer, R., Ray, D.S.: The AltaVista Search Revolution. Osborne-McGraw Hill, New York (1997)
2. Torres-Moreno, J.M., Velázquez-Morales, P., Meunier, J.G.: Condensés de textes par des méthodes numériques. In: JADT 2002, France pp. 723–734 (2002)
3. SanJuan, E., Ibekwe-SanJuan, F.: Text mining without document context. Information Processing and Management 42, 1532–1552 (2006)
4. Salton, G.: The SMART Retrieval System - Experiments un Automatic Document Processing. Englewood Cliffs (1971)
5. Morris, A., Kasper, G., Adams, D.: The effects and limitations of automated text condensing on reading comprehension performance. In: Advances in automatic text summarization, U.S.A, pp. 305–323. The MIT Press, Cambridge, MA (1999)
6. Paice, C.D.: Another stemmer. SIGIR Forum 24(3), 56–61 (1990)
7. Porter, M.F.: An algorithm for suffix stripping. Program 14(3), 130–137 (1980)
8. Siegel, S., Castellan, N.: Nonparametric statistics for the behavioral sciences. McGraw-Hill, New York (1988)
9. Buckley, C.: Looking at limits and tradeoffs: Sabir research at trec, In: Proc. of the 14th Text REtrieval Conference (TREC 2005), Gaithersburg, Maryland, U.S.A 13 (2005)
10. Liu, S., Yu, C.: University of Illinois Chicago at TREC. In: Proc. of the 14th Text REtrieval Conference (TREC 2005), Gaithersburg, Maryland, U.S.A 7 (2005)

[3] Medical Subject Headings, the thesaurus associated to MEDLINE descriptors.

[4] Unified Medical Language System.

The Bootstrapping Based Recognition of Conceptual Relationship for Text Retrieval

Yi Hu, Ruzhan Lu, Yuquan Chen, Xiaoying Chen, and Jianyong Duan

Department of Computer Science and Engineering,
Shanghai Jiao Tong University, Shanghai, China
{huyi,lu-rz,yqchen,chen-xy,duan_jy}@cs.sjtu.edu.cn

Abstract. The dependence analysis is usually the key for improving the performance of text retrieval. Compared with the statistical value of a conceptual relationship, the recognition of relation type between concepts is more meaningful. In this paper, we explored a bootstrapping method for automatically extracting semantic patterns from a large-scale corpus to identify the geographical "be part of" relationship between Chinese location concepts in contexts. Our contributions different from other bootstrapping methods lie in: (1) introducing a bi-sequence alignment algorithm in bio-informatics to generating candidate patterns, and (2) giving a new evaluating metric for patterns' confidence to enhance their extracting qualities in next iteration. In terms of automatic recognition of "be part of" relationship, the experiments showed that the pattern set generated by our method achieves higher coverage and precision than DIPRE does.

1 Introduction

The independence assumption has been widely used in current retrieval models. Although the assumption makes the models easily to build up, the independence between words in language is obviously trustless in fact. This makes searchers consider dependence models further.

Many dependence models cannot always enhance the retrieval performance. There exist two reasons intuitively:

1. The relationships between concepts are often real number values to express the strength and weakness of linkage [10][11] rather than the relation type between concepts. For example, "江苏 (Jiangsu, a province of China)" and "南京 (Nanjing, a city in Jiangsu province)" can definitely capture strong linkage, but it cannot know the accurate relationship that "Nanjing" geographically belongs to "Jiangsu".
2. The value of relationship between (Jiangsu, Nanjing) depends on the statistical information or their distance in domain ontology, but the numerical value is far away from the human cognition, and is hard to explain.

Naturally, a promising idea based on accurate conceptual relationship for information retrieval has been presented [12][13]. We think that a retrieval model based on concepts ought to look for occurrences of concepts and recognize conceptual relationship. Apparently, the deep understanding is closer to human cognition.

Z. Kedad et al.(Eds.): NLDB 2007, LNCS 4592, pp. 264–271, 2007.

Note that in terms of information retrieval based on concepts, it needs recognizing all relationship types between concepts to build up a complete conceptual graph of topic, and then provide response to the users' requirement. This paper only takes the "be part of" relationship as example, and proposes an automatically method for constructing the relationship under bootstrapping learning. We expect this study can be generalized to other recognitions of relationships in the ongoing research.

Extracting information from data can be categorized into supervised and unsupervised metrics in terms of learning method. To supervised learning, it needs to annotate a lot of data in advance [9]. The difficulty of this learning method lies in huge cost of human effort and time. In order to solve this problem, it is naturally to think about using non-annotated data to obtain useful information. The practice proved that this idea is available.

Seen from the previous efforts, unsupervised bootstrapping had been used in many fields of information extraction. With respect to recognizing concept types (weapon names, terrorism organizations etc.), the efforts in [2][5][6][7] are significant attempts. On the other hand, extracting conceptual pairs with certain relationships from English or Chinese corpus are also tested [1][3][4].

The goal of above efforts lies in creating a knowledge base for information service system. For example, when a user inquires that where is the headquarter of BOEING, the information service system can directly answer "SEATTLE". Note that our study does not care whether a word sequence is a weapon name or SEATTLE is the headquarter location of BOEING, but the support for determining whether two concepts in given context have certain relationship. The change of goal brings the IR system more support from IE technique.

2 Our Contributions

Our method is implemented based on the idea of DIPRE [1] by proposing a new SPG (Semantic Pattern Getter) system, which is under the bootstrapping framework to iteratively generate patterns for recognizing the Chinese conceptual relationship from contexts. The contributions of this paper lie in:

1. Introducing the bi-sequence alignment algorithm in bioinformatics to extract multiple common subsequences (MCS) for getting flexible expression of contexts rather than the single longest common subsequence (LCS) in DIPRE. We give a new algorithm.
2. Defining a new evaluation metric for the confidence of a pattern, which improves the extracting quality.

2.1 Pattern Definition

In the large-scale corpus, SPG finds all the sentence-level contexts containing both the two location concepts (pre-location, post-location). For example, we can find such a context of pair (Minhang, Shanghai) in corpus: " The progressive Minhang district is located in the southwest of Shanghai city."

In practice, "pre-location" possibly belongs to "post-location" and "post-location" also possibly belongs to "pre-location". This makes our extraction more flexible and more hopeful to get more patterns. The two location concepts separate a context into three parts: the word sequence before pre-location (“left”); the word sequence between the two locations (“middle”) and the word sequence following pos-location (“right”). A candidate pattern comes from the three parts. At last, a context is a six-tuple (left, pre-location, middle, post-location, right) and a formalized pattern is a five-tuple: (*prefix, middle, suffix, order, confidence*). Where,

prefix: the components of the pattern extracted from the “left” part of contexts;
middle: the components of the pattern extracted from the “middle” part of contexts;
suffix: the components of the pattern extracted from the “right” part of contexts;
order: a boolean value, is defined by:

$$order = \begin{cases} 1 & pre-location \text{"}be\ part\ of\text{"}\ post-location \\ -1 & post-location \text{"}be\ part\ of\text{"}\ \ pre-location \end{cases} \tag{1}$$

confidence: the confidence value of the pattern.

2.2 Generating Patterns

Patterns are captured from the contexts of seeds. The procedure is illustrated as:

Step1. Find all the occurrences of every seed in the large-scale corpus and record the left, middle and right strings of the context.
Step2. Use the bi-sequence alignment algorithm to extract patterns. Each two contexts will generate a candidate pattern.
Step3. All the candidate patterns will be chosen through the validation rules of pattern. If a pattern can be retained, then it is added the pattern set.

This idea for generating pattern has large difference with [1][3]. In [1], the pattern components are the longest common subsequence in similar contexts. In [3], in terms of a conceptual pair, its contexts create only one pattern represented by a vector. The pattern matching needs an experimental threshold predefined by human. Therefore, the precision and recall cannot be guaranteed simultaneously.

We generate a candidate pattern from every two contexts of a seed. Because there is detailed description in [14], we do not describe the bi-sequence alignment algorithm in details. But we can see an instance generated from two contexts:

Context1: 公司地址：中国上海市江浦路1515号。
(Company Address: No. 1515, Jiang Pu Rd., Shanghai City, China.)
Context2: 公司通信地址：中国上海市国顺路549号。
(Company Corresponding Address: No.549, Guo Shun Rd., Shanghai City, China.)

The bi-sequence alignment algorithm extracts a pattern denoting the “be part of” relationship between two location concepts: *(公司<ANY_STRING>地址：, NULL, 市<ANY_STRING>路<ANY_STRING>号。, -1, <Confidence>)*. In order to make the pattern not to be overly generalized, i.e., “<ANY_STRING>” strings are too more. We define the following validation rules to choose new candidate patterns with higher qualities.

Rule 1. The “prefix” and “suffix” cannot be just a < ANY_STRING >;
Rule 2. Both the most right component of “prefix” and the most left component of “suffix” cannot be an “<ANY_STRING>”;
Rule 3. The “prefix”, “middle” and “suffix” cannot just be punctuation;

These rules are simple and easy to understand. They are the experiences from extracting patterns. If we can introduce more linguistic knowledge, we will design more valid rules for filtering candidates to improve the quality of patterns.

In bootstrapping iterations, to each tuple *t* in seed set *Tuples*, the system looks for all the contexts containing *t* in the training corpus and creates a context set named C_t. If the context set is fixed, every sentence has its fixed order in the set. Therefore, to every two contexts in the set, the algorithm creates a candidate pattern and then filters all the candidates via the validation rules. Figure 1 illustrates this procedure.

```
$GeneratePatterns(Tuples)$
  $for\ each\ tuple\ t \in Tuples$
    $C_t\ =\ CreateContexts(t);$
    $for\ each\ c_i\ \in\ C_t$
      $for(\ j = i;\ j \leq |\ C_t\ |;\ j++)$
      {
        $P_{ij}\ =\ BiSequenceAlignment(c_i, c_j);$
        $if\ P_{ij}\ satisfies\ the\ three\ rules\ R1, R2\ and\ R3$
          $Patterns\ \leftarrow\ P_{ij}\ ;$
      }
  $Return\ Patterns;$
```

Fig. 1. Pattern generating algorithm based on bi-sequence alignment

Please note that, the system removes those Chinese stop-words, such as “的 (of)” and “了 (le)”. In terms of a context set containing *N* sentences, the function *GeneratePatterns* can at most generate C_N^2 candidate patterns. But a lot of them will be removed after the validation rules are used.

2.3 Pattern Confidence

How to evaluate the pattern confidence is the key factor affecting the coverage and precision of final pattern set. We define a new confidence definition in (2).

$$Conf_{RlogF}(P) = \left(\frac{P_{positive}}{P_{positive} + P_{negative}} \right) \times [log_2(P_{new} + 1)]^{\alpha} \tag{2}$$

Where, *P* denotes some pattern and $P_{positive}$ denotes the number of correct pairs in all the extracted pairs in current iteration; $P_{negative}$ denotes the number of wrong pairs; and P_{new} is the number of newly extracted pairs. The pattern confidence (2) expresses the precision ingredient ($P_{positive}/(P_{positive}+P_{negative})$) and the recall ingredient ($[log_2(P_{new}+1)]^{\alpha}$) of the pattern *P*. and α is the rate of correct pairs in new ones. This shows that when P_{new} is more and α is larger, the pattern confidence is enlarged more. If P_{new} or α is equal to 0, the pattern confidence backs to its initial definition in [5].

2.4 Pair Extraction

After the system generates the pattern set, it scans the corpus and gathers all the pairs whose contexts can be matched by these patterns. In order to avoid generating too much candidate pairs, we need to evaluate them either [3]. To a candidate pair, the patterns that can extract the pair in corpus groups into a set $PSet = \{P_i\}$. We further use the pattern confidence to estimate the pattern probability of correctly extracting a pair and these patterns are regarded as independent. Therefore the confidence of *t* can be calculated by:

$$Conf(t) = 1 - \prod_{i=0}^{|PSet|} \left(1 - Conf(P_i)\right) \tag{3}$$

We only choose the pairs whose confidence is higher than a threshold (= 0.3) as candidate pairs. We assume that the existed seed set is *S*, and we do not use the seeds ever being used. Then the next iteration the system use the seed set *S'*:

$$S^{'} \leftarrow P_{positive} - S \tag{4}$$

3 Experiment Results

In this section we give the experiment results of SPG and compare them with the results of DIPRE. We employ the corpus CWT100g (Chinese Web Test collection with 100GB web pages). This web collection provides training and testing collections for information retrieval and information extraction. It covers most topics.

For the limited ability of the computers in our laboratory, we only use the sub-collection ranging from #29 to #39 as training corpus and use the sub-collections ranging from #60 to #63 as testing corpus. We process the web pages in advance by removing the HTML tags from training and testing corpus [8]. This paper uses the following pairs as initial seeds and train the system in terms of "be part of" relation.

Table 1. Initial Chinese Seed List

吉林 (Ji Lin)	长春 (Chang Chun)
中国 (China)	黑龙江 (Hei Longjiang)
北京 (Bei Jing)	海淀 (Hai Dian)
陕西 (Shan Xi)	西安 (Xi An)
上海 (Shang Hai)	静安 (Jing An)

In our experiments, we choose DIPRE to be compared with our SPG. Because we use the pure text content, we do not add the URL information into DIPRE. We run three iterations. The results of experiments can be viewed in the following table.

Table 2. Pairs and Patterns in every iteration in training collection

	SPG		DIPRE	
Iterations	Tuples	Patterns	Tuples	Patterns
1	5	28	5	21
2	57	51	31	33
3	171	169	20	25

Column 2 and 4 in Table 2 show the number of seeds used in iterations of SPG and DIPRE. In the first run, the two systems are both use the same five seeds as initial input and the last two runs use the seeds extracted from the previous iteration. Column 3 and 4 show the number of valid patterns. Seen from the experiment results, SPG can extract more conceptual pairs than DIPRE does from the same collections.

Then we need to consider the applying ability of SPG in constructing the conceptual relation between concepts. We use the patterns obtained from the three iterations into the test collection, respectively. Because the test collection is fixed, the system extracting MORE correct pairs has the higher recall. And their precision is easily evaluated manually. Figure 2 and Figure 3 gives the coverage and precision of the two systems, respectively.

Seen from Figure 2, SPG has the better coverage than DIPRE because SPG can get more conceptual pairs. On the other hand, the precision illustrated in Figure 3 also shows that SPG does better than DIPRE: the precision of SPG in three iterations keeps about 90% with slightly fall; but DIPRE's precision drops very quickly, and

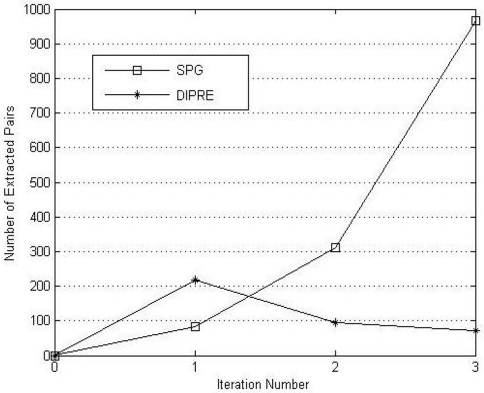


Fig. 2. The number of pairs extracted from test collection via patterns obtained after each training run

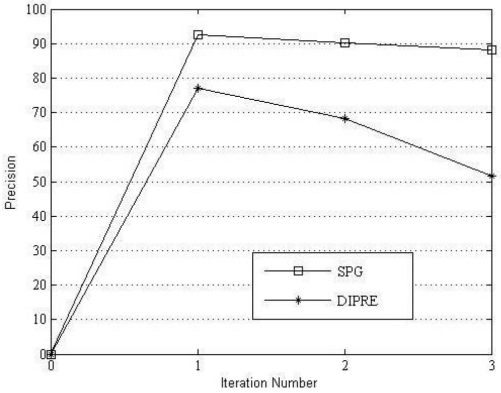


Fig. 3. The precision of pairs extracted from test collection via patterns obtained after each training run

arrives at 50% in its third iteration. After the experiment on testing corpus, SPG system totally obtain 1504 pairs at last, and 1358 of them are correct (precision = 90.29%). DIPRE obtains 588 pairs, and 384 of them are correct (precision = 60.20%). The experiment shows that the pattern set of SPG has the higher coverage and precision than DIPRE does.

4 Conclusion

In this paper, we propose a system named SPG for determining whether two concepts satisfy the "be part of" relation in given context. We introduce a bi-sequence alignment algorithm in bio-informatics to capture clearer and more understandable patterns. And we also define a new confidence evaluating method for patterns. After training the SPG system, the experiment results show that our system performs better than DIPRE in terms of coverage and precision.

The better idea ought to refine the <ANY_STRING> by identifying named entities in contexts, such as time, location and person name. Thus the semantic patterns will be more understandable and accurate.

As mentioned above, our study aims at serving the retrieval model based on concepts. Therefore the pattern set ought to give the support to determine the relationship between two concepts in contexts. The pattern set should be used in many web pages and the precision should be guaranteed at the same time. For a long-term goal, recognizing the accurate relationships definitely bring IR huge benefits, but to the performance of current system, it is still a hard job. The key points of our future work lies in developing an advanced SPG system and extending the method proposed in this paper to the recognition of other relationships.

Acknowledgments. This work is supported by NSFC Major Research Program 60496326: Basic Theory and Core Techniques of Non Canonical Knowledge.

References

1. Brin, S.: Extracting patterns and relations from the World Wide Web. In: Proc. of the 1998 International Workshop on the Web and Databases (1998)
2. Riloff, E., Jones, R.: Learning dictionaries for information extraction by multi-level bootstrapping. In: Proc. of the Sixteenth National Conference on Artificial Intelligence (1999)
3. Agichtein, E., Gravano, S.: Snowball: Extracting relations from large plain-text collections. In: Proc. of the 5th ACM International Conference on Digital Libraries (2000)
4. Zhang, Y., Zhou Joe, F.: A trainable method for extracting Chinese entity names and their relations. In: Proc. of the second Chinese Language Processing Workshop (2000)
5. Thelen, M., Riloff, E.: A Bootstrapping Method for Learning Semantic Lexicon using Extraction Pattern Contexts. In: Proc. of the 2002 Conference on Empirical Methods in Natural Language Processing (2002)
6. Lin, W., Yangarber, R., Grishman, R.: Bootstrapped Learning of Semantic Classes from Positive and Negative Examples. In: Proc. of the ICML-2003 Workshop on the Continuum from Labeled to Unlabeled Data, Washington DC (2003)
7. Etzioni, O., Cafarella, M., Downey, D., Popescu, A.M., et al.: Methods for Domain-Independent Information Extraction from the Web: An Experimental Comparison. In: Proc. of the AAAI Conference (2004)
8. Han, H., Elmasri, R.: Learning Rules for Conceptual Structure on the Web. Journal of Intelligent Information System 22(3), 237–256 (2004)
9. Fisher, D., Soderland, S., McCarthy, J., Feng, F., Lehnert, W.: Description of the Umass systems as used for MUC-6. In: Proc. of the 6th Message Understanding Conference. Columbia, MD (1995)
10. Gao, J., Nie, J.-Y., Guangyuan, et al.: Dependence language model for information retrieval. In: Proc. of the 27th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 170–177 (2004)
11. Nallapati, R.: J. Allan. Capturing term dependencies using a language model based on sentence tree. In: Proc. of CIKM'02. pp. 383-390 (2002)
12. Genest, D., Chenin, M.A: Content-Search Information Retrieval Process Based on Conceptual Graphs. Knowledge and Information Systems Journal 8, 292–309 (2005)
13. Roussey, C., Calabretto, S., Pinon, J.-M.: A New Conceptual Graph Formalism Adapted for Multilingual Information Retrieval Purposes. DEXA, pp. 92–101 (2001)
14. Sammeth, M., Morgenstern, B., Stoye, J.: Divide-and-conquer multiple alignment with segment-based constraints. Bioinformatics 19(2), 189–195 (2003)

A Framework of NLP Based Information Tracking and Related Knowledge Organizing with Topic Maps

Asanee Kawtrakul[1], Chaiyakorn Yingsaeree[1], and Frederic Andres[2]

[1] NAiST Research Laboratory, Kasetsart University, Bangkok, Thailand
asanee.kawtrakul@nectec.or.th, chaikorn@gmail.com
[2] National Institute of Informatics, Tokyo, Japan
andres@nii.ac.jp

Abstract. This paper presents a computational framework for information extraction and aggregation which aims to integrate and organize the data/information resources that spread throughout the Internet in the manner that makes them useful for tracking events such as natural disaster, and disease dispersion. We introduce a simple statistical information extraction technique for summarizing the document into a predefined structure. We apply the topic maps approach as a semantic layer in aggregating and organizing the extracted information for smart access. In addition, this paper also carries out a case study on disease dispersion domain using the proposed framework.

1 Introduction

In order to monitor important events, such as disease dispersion, the occurrence of tsunami and terrorism connections, an operator and a decision maker need data, information and knowledge. Internet news and other online articles (e.g. wiki-like knowledge and web logs) are the good resources for these kinds of information which describe the world around us rapidly by talking about the update events, states of affairs, knowledge, people and experts who participate in. However, sources of these data are scattered across several locations and web sites with heterogeneous formats that offer a large volume of unstructured information. Moreover, the needed knowledge was too difficult to find since the traditional search engines return ranked retrieval lists that offer little or no information on semantic relationships among those scattered information, and, even if it was found, the located information often overload since there was no content digestion. Accordingly, the automatic extraction of information expressions, especially the spatial and temporal information of the events, in natural language text with question answering system has become more obvious as a system that strives for moving beyond information retrieval and simple database query.

However, one major problem that needs to be solved is the recognition of events which attempts to capture the richness of event-related information with their temporal and spatial information from unstructured text. Various advanced technologies including name ehintities recognition and related information extraction, which need natural language processing techniques, and other information technologies, such as GIS, are utilized to enable emerging of new methodologies for

Z. Kedad et al.(Eds.): NLDB 2007, LNCS 4592, pp. 272–283, 2007.

information extraction and aggregation with problem-solving solutions (e.g. the know-how from livestock experts from countries with experiences in handling bird flu situation). Ontology and Topic Map model are also applied for organizing related knowledge or related topics.

In this paper, we present a systematic attempt to provide a computational framework for information extraction and aggregation which aims to integrate and organize the data/information resources dispersed across web resources in a manner that makes them useful for tracking events such as natural disaster, and disease dispersion. The remainder of this paper is structured as follows: Section 2 describes the nontrivial problems in information tracking; Section 3 gives the conceptual framework for information collection, extraction and aggregation including the information service for different target user groups. Section 4 gives more details of the system process regarding the information extraction module. Section 5 discusses the knowledge service and visualization module. Finally, in Section 6, we conclude and discuss the next step and challenges.

2 Non-trivial Issues in Information Tracking

Lessons learned from special monitoring areas or areas that has past experiences with the interested events (e.g. the best practice for governments to handle bird flu situation), the collection of important events and their related information (e.g. virus transmission from one area to other locations and from livestock to humans) are important. However, collecting and extracting these data from the Internet have two main nontrivial problems: overload and scattered information, and salient information extraction from unstructured text.

2.1 Overloaded and Scattered Information

The knowledge applicable to an intended problem solving consists of data items and/or information that are organized and processed to convey understanding, experience, accumulated learning, and expertise. However, sources of these data are scattered across several locations and websites with heterogeneous formats. For example, the information about Bird Flu consisting of policy for controlling the events, disease infection management, and outbreak situation may appear in different websites as shown in Fig. 1. Consequently, collecting the needed information from scattered resources is very difficult since the semantic relations among those resources are not directly stated. Although we can gather those information, the collected information often overload since there is no content digestion. Accordingly, manually solving those problems will consume a lot of time and power, and the system that can collect, extract and organize those information automatically will definitely become a useful tool for knowledge construction and organization.

2.2 Salient Information Extraction from Unstructured Texts

In order to reduce time consumption for users to consume the information, only salient information must be extracted. As it happens, most of those information, such as time of the event, location that event occurred, and the detail of the event, are left implicitly in the texts. For example: in the text in Fig. 2, the time expression “15 February” mentioned only “date and month” of the bird flu event but did not mention the ‘year’. The patient and her condition (i.e. ’37-year-old female’, and ‘died’) was caused by bird flu which is written in the text as ‘Avian influenza’ and ‘H5N1 avian influenza’. Accordingly, the essential component of computational model for event information capturing is the recognition of interested entities including time expression, such as ‘yesterday’, ‘last Monday’, and ‘two days before’, which becomes an important part in the development of more robust intelligent information system for event tracking.

Information extraction in traditional way extracts a set of related entities in the format of slot and filler, but the description of information in Thai text such as locations, patient’s condition, and time expressions can not be limited to a set of related entities because of the problems of using zero anaphora [1]. Moreover, to activate the frame for filling the information, name entity classification must be robust as it has been shown in [2].

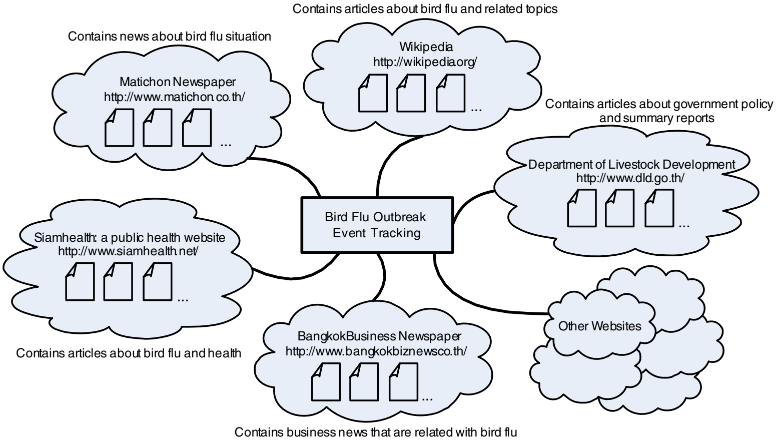
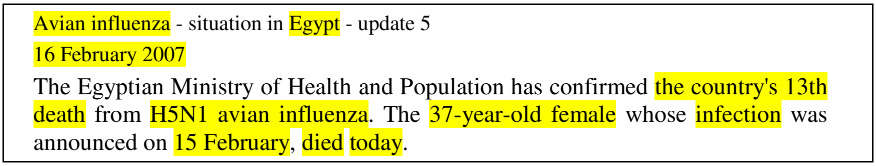

Contains articles about bird flu and related topics
Wikipedia
http://wikipediaorg/
...
Contains news about bird flu situation
Matichon Newspaper
http://www.matichon.co.th/
...
Contains articles about government policy and summary reports
Department of Livestock Development
http://www.dld.go.th/
...
Bird Flu Outbreak Event Tracking
Siamhealth: a public health website
http://www.siamhealth.net/
...
Contains articles about bird flu and health
BangkokBusiness Newspaper
http://www.bangkokbiznewsco.th/
...
Other Websites
Contains business news that are related with bird flu

Fig. 1. The information required for tracking bird flu outbreak is scattered across the Internet



Avian influenza - situation in Egypt - update 5

16 February 2007

The Egyptian Ministry of Health and Population has confirmed the country's 13th death from H5N1 avian influenza. The 37-year-old female whose infection was announced on 15 February, died today.

Fig. 2. The example of the document containing bird flu outbreak situation

3 A Framework of Information Extraction for Event Tracking

A crucial first step in the automatic extraction of information from unstructured texts is the capacity to identify what events are being described and to make it explicit when these events occurred. Since the web consists of a large extent of unstructured or semi-structured natural language text, several techniques (i.e., language engineeering and knowledge engineering) are applied to information extracting and integrating. For language engineering, word segmentation [3], named entity recognition [2], shallow parsing [4], shallow anaphora resolution and discourse processing [2,5,6] are utilized. For knowledge engineering, the concept of frame for structuring the extracted information is applied. For ontological engineering, task-oriented ontology, ontology maintenance [7] and Topic Maps [8] model are applied for information aggregation and organizing for smart access. Fig. 3 overviews the system architecture which has been designed for event tracking and its related knowledge organization for aiding multi-users information service provision [7,9,10]. The framework consists of six main parts:

Information and Knowledge Extraction: To generate useful knowledge from collected documents, two important modules, information extraction and knowledge extraction, are utilized. Ontological topic maps are used as a knowledge base to facilitate the knowledge construction process. The information extraction and integration module is responsible for summarizing the document into a predefined frame-like/structured database, such as <disease name, dispersion location and time, status of patient's condition>. The knowledge extraction and generalization is responsible for extracting useful knowledge (e.g. general symptom of disease) from collected document. The extracted knowledge is represented as a structured knowledge and rules. The output of both modules is stored in RDF/OWL repository.

Distributed Information Collection: The information, both unstructured and semi-structured documents are gathered from many sources. Periodic web crawler and HTML Parser [11] are used to collect and organize related information. The domain specific parser [12] is used to extract and generate meta-data (e.g. title, author, and date) for interoperability between disparate and distributed information. The output of this stage is stored in the document warehouse.

Content-based Metadata Extraction: To organize the information scattered at several locations and websites, Textual Semantics Extraction [10] is used to create a semantic metadata for each document stored in the document warehouse. Guided by the ontology stored in Ontological Topic Map, the extraction process can be taught of as a process for assigning a topic to considered documents.

Knowledge Organization: After all required information and knowledge is generated, the Topic Map (ISO ISO13250) including topics and related associations is a proxy to access resource occurrences. The generation is done by combining the ontological topic map and the metadata extracted from content-based metadata extraction. The generated topic map is represented as a XTM document and, then, sent to the Knowledge Visualization module.

Knowledge Service: This module is responsible for creating response to users' query. The query processing is used to interact with the RDF/OWL Knowledge Repository, while inference engine is used to infer new knowledge that is not explicitly stored in the knowledge repository.

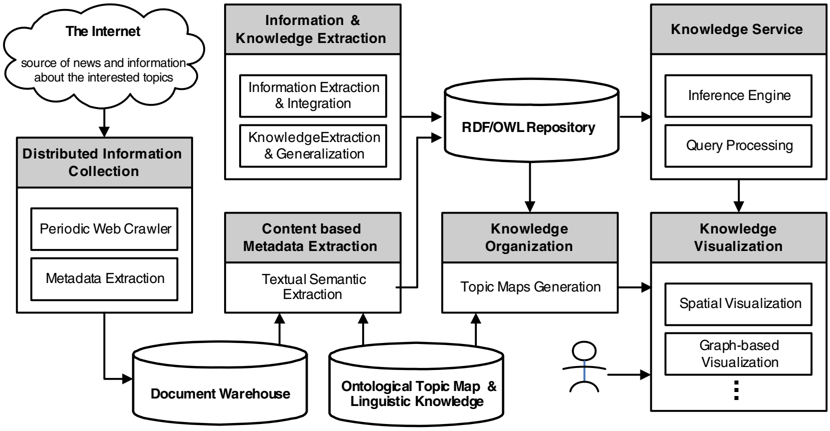

The Internet
source of news and information about the interested topics
Distributed Information Collection
Periodic Web Crawler
Metadata Extraction
Information & Knowledge Extraction
Information Extraction & Integration
KnowledgeExtraction & Generalization
Content based Metadata Extraction
Textual Semantic Extraction
RDF/OWL Repository
Knowledge Organization
Topic Maps Generation
Knowledge Service
Inference Engine
Query Processing
Knowledge Visualization
Spatial Visualization
Graph-based Visualization
Document Warehouse
Ontological Topic Map & Linguistic Knowledge

Fig. 3. The architecture of the proposed framework

Knowledge Visualization: After obtaining all required information from the previous module, the last step is to provide the means to help users consume that information in an efficient way. To do this, many visualization functions is provided. For example, Spatial Visualization can be used to visualize the information extracted from the Information Extraction module and Graph-based Visualization can be used to display hierarchal categorization in the topic maps in an interactive way [10].

Due to page limitation, this paper will focus in only Information Extraction module, Knowledge Service module and Knowledge Visualization module.

4 Information Extraction

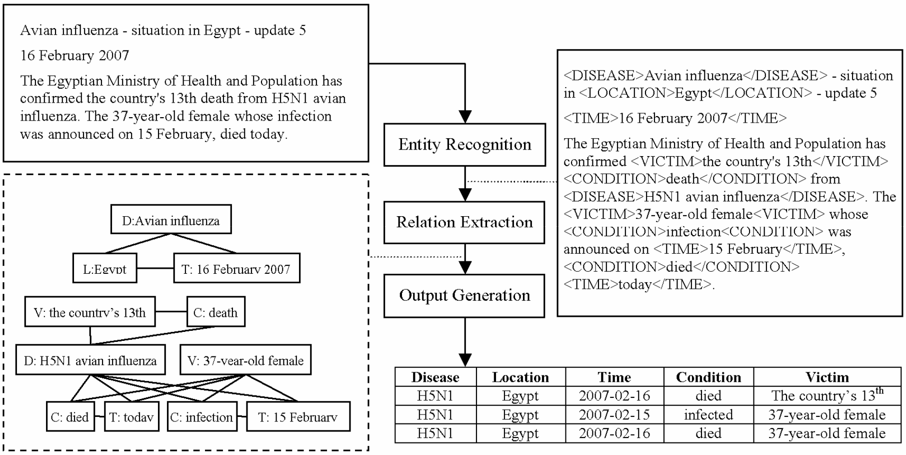
The proposed model for extracting information from unstructured documents consists of three main components, namely Entity Recognition, Relation Extraction, and Output Generation, as illustrate in Fig. 4. The Entity Recognition module is responsible for locating and classifying atomic elements in the text into predefined categories such as the names of diseases, locations, and expressions of times. The Relation Extraction module is responsible for recognizing the relations between entities recognized by the Entity Recognition module. The output of this step is a graph representing relations among entities where a node in the graph represents an entity and the link between nodes represents the relationship of two entities. The Output Generation module is responsible for generating the n-tuple representing extracted information from the relation graph. The details of each module are described as followed.

4.1 Entity Recognition

To recognize an entity in the text, the proposed system utilizes the work of H. Chanlekha and A. Kawtrakul [2] that extracts entity using maximum entropy [13], heuristic information and dictionary. The extraction process consists of three steps. Firstly, the candidates of entity boundary are generated by using heuristic rules, dictionary, and statistic of word co-occurrence. Secondly, each generated candidate is then tested against the probability distribution modeled by using maximum entropy. The features used to model the probability distribution can be classified into four categories: Word Features, Lexical Features, Dictionary Features, and Blank Features as described in [2]. Finally, the undiscovered entity is extracted by matching the extracted entity against the rest of the document. The experiment with 135,000 words corpus, 110,000 words for training and 25,000 words for testing, shown that the precision, recall and f-score of the proposed method are 87.60%, 87.80%, 87.70% respectively.

4.2 Relation Extraction

To extract the relation amongst the extracted entities, the proposed system formulates the relation extraction problem as a classification problem. Each pair of extracted entity is tested against the probability distribution modeled by using maximum entropy to determine whether they are related or not. If they are related, the system will create an edge between the nodes representing those entities. The features used to model the probability distribution are solely based on the surface form of the word surrounding the considered entities; specifically, we use the word n-gram and the location relative to considered entities as features. The surrounding context is classified into three disjointed zone: prefix, infix, and suffix. The infix is further segmented into smaller chunks by limiting the number of words in each chunk. For example, to recognize the relation between VICTIM and CONDITION in the sentence "The [VICTIM] whose



Avian influenza - situation in Egypt - update 5

16 February 2007

The Egyptian Ministry of Health and Population has confirmed the country's 13th death from H5N1 avian influenza. The 37-year-old female whose infection was announced on 15 February, died today.

<DISEASE>Avian influenza</DISEASE> - situation in <LOCATION>Egypt</LOCATION> - update 5

<TIME>16 February 2007</TIME>

The Egyptian Ministry of Health and Population has confirmed <VICTIM>the country's 13th</VICTIM> <CONDITION>death</CONDITION> from <DISEASE>H5N1 avian influenza</DISEASE>. The <VICTIM>37-year-old female<VICTIM> whose <CONDITION>infection<CONDITION> was announced on <TIME>15 February</TIME>, <CONDITION>died</CONDITION> <TIME>today</TIME>.

Disease	Location	Time	Condition	Victim
H5N1	Egypt	2007-02-16	died	The country's 13th
H5N1	Egypt	2007-02-15	infected	37-year-old female
H5N1	Egypt	2007-02-16	died	37-year-old female

Fig. 4. Overview of the information extraction module

[CONDITION] was announced on ....", the prefix, infix and suffix in this context is 'the', 'whose', and 'was announced on ....' respectively.

To determine the best parameter of the system, we conduct the experiment with 257 documents, 232 documents for training and 25 documents for testing. We vary the n-gram parameter from 1 to 7 and set the number of words in each chunk as 3, 5, 7, 10, 13, and 15. The result is illustrated in Fig. 5. The legend in the figure is the number of words in each chunk; for example, WLLTY3 means that the number of words is set as 3. The evident shows that f-score is maximum when n-gram is 4 and number of words in each chunk is 7. The precision, recall and f-score at the maximum f-score are 58.59%, 32.68% and 41.96% respectively.

4.3 Output Generation

After obtaining a graph representing relations between extracted entities, the final step of information extraction is to transform the relation graph into the n-tuple representing extracted information. Heuristic information is employed to guide the transformation process. For example, to extract the information about disease outbreak (i.e. disease name, time, location, condition, and victim), the transformation process will starts by analyzing the entity of the type condition, since each n-tuple can contain only one piece of information about the condition. It then travels the graph to obtain all entities that are related to considered condition entity. After obtaining all related entities, the output n-tuple is generated by filtering all related entities using constrain imposed by the property of each slot. If the slot can contains only one entity, the entity that has the maximum probability will be chosen to fill the slot. In general, if the slot can contain up to n entities, the top-n entities will be selected. In addition, if there is no entity to fill the required slot, the mode (most frequent) of the entity of that slot will be used to fill instead. The time expression normalization using rule-based system and synonym resolution using ontology are also performed in this step to generalize the output n-tuple. The example of the input and output of the system are illustrated in Fig. 4.

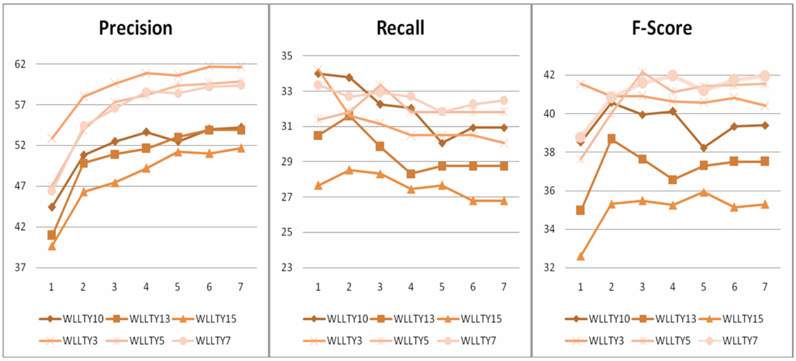


Fig. 5. Experimental results of relation extraction module

5 Knowledge Service and Visualization

After all required information and knowledge is generated and stored in the Ontological Topic Map and RDF/OWL Repository, users can consume those information and knowledge by using Knowledge Visualization module which combined the extracted information by interacting with Knowledge Service module and the topic maps model by interacting with Knowledge Organization module to generate visualizations that helps users consume the information in an efficient way. The details of Knowledge Service and Knowledge Visualization module are described as followed.

5.1 Knowledge Service

The Knowledge Service module is responsible for interacting with RDF/OWL Repository to generate the response to user's request. The framework currently supports four types of query. The detail of each query type is summarized in Table 1.

5.2 Knowledge Visualization

The Knowledge Visualization is responsible for representing the extracted information and knowledge in an efficient way. For example, we can create many visualization techniques that response to users who need concise and knowledge organization, such as Spatial Visualization and Graph-based Visualization as described below.

Spatial Visualization
The spatial-based visualization functions help users to visualize the extracted information (e.g. the bird flu outbreak situation extracted in Fig. 4.) using geographical information system, such as Google Earth. This kind of visualization allows the users to click on the map to get the outbreak situation of the area that they want. In addition, by viewing the information in the map users can see the spatial relations amongst the outbreak situations easier than without the map. The Google Earth integrated system for visualizing the extracted information about bird flu situation is shown in Fig. 6.

Graph-based Visualization
The graph-based visualization function is useful to show the global structure of topic maps and relations between different nodes in a 3D visual space. In a topic maps structure, various topics are associated to each other based on relationships. The graph viewer provides a better global understanding of the content by exploring through graph nodes. The kind of intuitive visualization of the Topic Maps allows browsing through all the topics and related relationships defined in the Topic Maps as shown in Fig. 7. The graph can be moved and restructured along its topological view according to the user's need.

Table 1. Detail of four query types supported by the Knowledge Service module

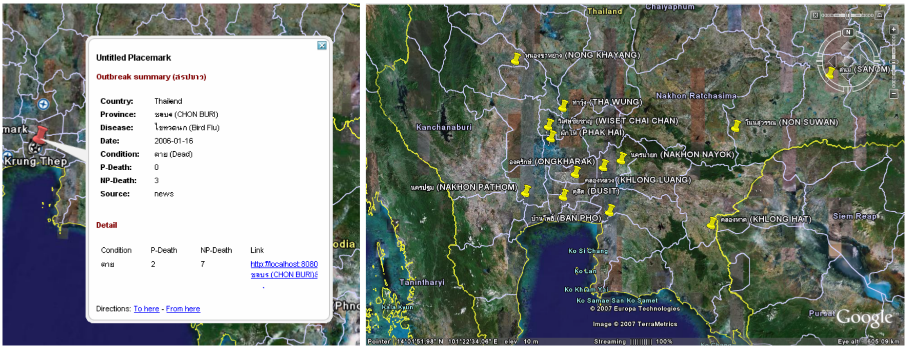
Query type	Description

Query by Object	A mechanism employed when users know the object but want to acquire more information/knowledge about it. The query example is as following:
	SELECT qa_who,lblWho
	FROM {qa_who} ne:text {lblWho}
	WHERE (lblWho like "*เด็ก*")
	USING NAMESPACE ne = <http://naist.cpe.ku.ac.th/EventTracking#>

Query by Relationship	A mechanism employed when users know the relation label. For example, user can access knowledge repository such as "ne:atLocation". The query example is as following:
	SELECT disease,lblDisease,location,lblLocation
	FROM {disease} ne:text {lblDisease},
	{disease} ne:atLocation {location},
	{location} ne:text {lblLocation}
	WHERE (lblDisease like "*หวัดนก*") AND (lblLocation like "*เวียด*")
	USING NAMESPACE ne = <http://naist.cpe.ku.ac.th/EventTracking#>

Query by Instance	A mechanism employed when users know the instance or some parts of instance label that can access to knowledge repository such as "ne:Disease-3-10". The query example is as following:
	SELECT disease,lblDisease,location,lblLocation
	FROM {disease} ne:text {lblDisease},
	{disease} ne:atLocation {location},
	{location} ne:text {lblLocation}
	WHERE (lblDisease like "*หวัดนก*") AND (lblLocation like "*เวียด*")
	USING NAMESPACE ne = <http://naist.cpe.ku.ac.th/EventTracking#

Knowledge Reasoning	A mechanism used for inferring new knowledge from existing information/knowledge by using inference engine provided by OWLIM plug-in. The custom designed rule set is required to create new knowledge from existing one. For example, to generate new knowledge about the region that have bird flu situation, one can custom rule sets as following:
	RegionRules {
	Id : event_tracking
	Location <ne:locationOf> District
	District <ne:districtOf> Province
	Province <ne:ProvinceOf> Country
	Country <ne:CountryOf> Region
	...
	}
	SELECT disease, lblDisease, region, lblRegion
	FROM {location} ne: inRegionOf {region},
	{region} ne:text {lblRegion}
	WHERE (lblRegion like "เอเชียตะวันออกเฉียงใต้")
	USING NAMESPACE ne = <http://naist.cpe.ku.ac.th/EventTracking#>



Untitled Placemark

Outbreak summary (สรุปข่าว)

Country:	Thailand
Province:	ชลบุรี (CHON BURI)
Disease:	ไข้หวัดนก (Bird Flu)
Date:	2006-01-16
Condition:	ตาย (Dead)
P-Death:	0
NP-Death:	3
Source:	news

Detail

Condition	P-Death	NP-Death	Link
ตาย	2	7	http://localhost:8080 ชลบุรี (CHON BURI)

Directions: To here - From here

Fig. 6. Google Earth visualization for bird flu outbreak tracking

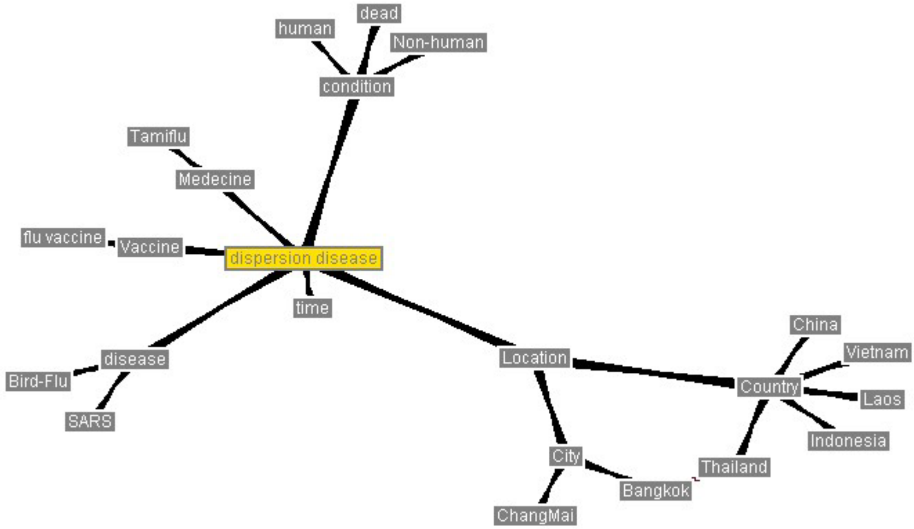


Fig. 7. Graph-based visualization of topic maps about dispersion disease

6 Related Work

The framework described in this paper is closely related to ProMED-PLUS [16], a system for the automatic “fact” extraction from plain-text reports about outbreaks of infectious epidemics around the world to database, and MiTAP [17], a prototype SARS detecting, monitoring and analyzing system. The difference between our framework and those systems is that we also emphasize on generating the semantic relations among the collected resources and organizing those information by using topic map model.

The proposed information extraction model that formulates the relation extraction problem as a classification problem is motivated by the work of J.Suzuki et. al. [19]

that proposed a HDAG kernel to solve many problems in natural language processing. The use of classification methods in information extraction is not new. Intuitively, one can view the information extraction problem as a problem of classifying a fragment of text into a predefined category which results in a simple information extraction system such as a system for extracting information from job advertisements [20] and business cards [21]. However, those techniques require the assumption that there should be only one set of information in each document, while our model could support more than one set of information.

7 Conclusion and Future Work

This paper presents a framework for extracting information and knowledge from unstructured documents that spread throughout the Internet by emphasizing on information extraction technique, event tracking and knowledge organizing. The work is going to develop the Textual Semantic Extraction for providing the automated topic maps construction process. This challenging work needs more complicate natural language processing with deeply semantic relations interpretation.

Acknowledgement

The work described in this paper has been supported by the grant of National Electronics and Computer Technology Center (NECTEC) No. NT-B-22-14-12-46-06, under the project “A Development of Information and Knowledge Extraction from Unstructured Thai Document”.

References

1. Kongwan, A., Kawtrakul, A.: Know-what: A Development of Object Property Extraction from Thai Texts and Query System. In: Proceedings of SNLP-2005, Bangkok, Thailand pp. 157–162,(2005)
2. Chanlekha, H., Kawtrakul, A.: Thai Named Entity Extraction by incorporating Maximum Entropy Model with Simple Heuristic Information. In: Proceedings of IJCNLP-2004, Hainan, China pp. 49–55 (2004)
3. Sudprasert, S., Kawtrakul, A.: Thai Word Segmentation based on Global and Local Unsupervised Learning. In: Proceedings of NCSEC-2003. Chonburi, Thailand (2003)
4. Satayamas, V., Thumkanon, C., Kawtrakul, A.: Bootstrap Cleaning and Quality Control for Thai Tree Bank Construction. In: Proceedings of NCSEC-2005, Bangkok Thailand pp. 849–860 (2005)
5. Grosz, B., Joshi, A., Weinstein, S.: Centering: A Framework for Modeling the Local Coherence of Discourse. Computational Linguistics 21, 203–225 (1995)
6. Chareonsuk, J., Sukvakree, Y., Kawtrakul, A.: Elementary Discourse unit Segmentation for Thai using Discourse Cue and Syntactic Information. In: Proceedings of SNLP. Bangkok, Thailand, pp. 85–90 (2005)

7. Kawtrakul, A., Suktarachan, M., Imsombut, A.: Automatic Thai Ontology Construction and Maintenance System. In: Proceedings of Ontolex Workshop on LREC, pp. 68–74 (2004)
8. Biezunski, M., Bryan, M., Newcomb, S.: ISO/IEC JTC1/SC34 (May 22, 2002) Available at http://www1.y12.doe.gov/capabilities/sgml/sc34/document/0322.htm
9. Kawtrakul, A., Imsombut, A., Thunyakijjanukit, A., Soergel, D., Liang, A., Sini, M., Johannsen, G., Keizer, K.: Automatic Term Relationship Cleaning and Refinement for AGROVOC. In: Proceedings of EFITA/WCCA. Vila Real, Portugal (2005)
10. Rajbhandari, S., Andres, F., Naito, M., Wuwongse, V.: Topic Management in Spatial-Temporal Multimedia Blog. In: the 1st IEEE International Conference on Digital Information Management (ICDIM 2006) Bangalore, India, December 6-8, 2006, pp. 81–88 (2006)
11. Thamvijit, D., Chanlekha, H., Sirigayon, C., Permpool, T., Kawtrakul, A.: Know-who: Person Information from Web Mining. In: Proceedings of NCSEC. Bangkok, Thailand, pp. 849–860 (2005)
12. Kawtrakul, A., Yingsaeree, C.: A Unified Framework for Automatic Metadata Extraction from Electronic Document. In: Proceedings of The International Advanced Digital Library Conference. Nagoya, Japan (2005)
13. Berger, A.L., Pietra, S.-A.D., Pietra, V.-J.D.: A maximum entropy approach to natural language processing. Computational Linguistics 22, 39–71 (1996)
14. Dean, M., Connolly, D., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F., Stein, L.A.: OWL Web Ontology Language Reference. W3C Recommendation (February 10, 2004) Available at http://www.w3.org/TR/owl-ref/
15. Garshol, L.M.: Living with Topic Maps and RDF (May 2003) Available at http://www.idealliance.org/papers/dx_xmle03/papers/02-03-06/02-03-06.html
16. Yangarber, R., Jokipii, L., Rauramo, A., Huttunen, S.: Information Extraction from Epidemiological Reports. In: Proceedings of HLT/EMNLP-2005. Canada (2005)
17. Damianos, L., Bayer, S., Chisholm, M.A., Henderson, J., Hirschman, L., Morgan, W., Ubaldino, M., Zarrella, J.: MiTAP for SARS detection. In: Proceedings of the Conference on Human Language Technology. Boston, USA, pp. 241–244 (2004)
18. Naito, M., Andres, F.: Application Framework Based on Topic Maps. In: TMRA 2005. Leipzig, Germany (2005) LNCS, vol. 3873, February 2006 pp. 42–52, DOI 10.1007/11676904_4, Charting the Topic Maps Research and Applications Landscape: First International Workshop on Topic Map Research and Applications, TMRA 2005, Leipzig, Germany, October 6-7, 2005, Revised Selected Papers Editors: Lutz Maicher, Jack Park ISBN: 3-540-32527-1
19. Suzuki, J., Sasaki, Y., Maeda, E.: Kernels for structured natural language data. In: Proceeding of NIPS 2003 (2003)
20. Zavrel, J., Berck, P., Lavrijssen, W.: Information extraction by text classification: Corpus mining for features. In: Proceedings of the workshop Information Extraction meets Corpus Linguistics. Athens, Greece (2000)
21. Kushmerick, N., Johnston, E., McGuinness, S.: Information extraction by text classification. In: Proceedings of IJCAI-2001 Workshop on Adaptive Text Extraction and Mining (2001)

DLSITE-1: Lexical Analysis for Solving Textual Entailment Recognition

Óscar Ferrández, Daniel Micol, Rafael Muñoz, and Manuel Palomar

Natural Language Processing and Information Systems Group
Department of Computing Languages and Systems
University of Alicante
San Vicente del Raspeig, Alicante 03690, Spain
{ofe,dmicol,rafael, mpalomar}@dlsi.ua.es

Abstract. This paper discusses the recognition of textual entailment in a text-hypothesis pair by applying a wide variety of lexical measures. We consider that the entailment phenomenon can be tackled from three general levels: lexical, syntactic and semantic. The main goals of this research are to deal with this phenomenon from a lexical point of view, and achieve high results considering only such kind of knowledge. To accomplish this, the information provided by the lexical measures is used as a set of features for a Support Vector Machine which will decide if the entailment relation is produced. A study of the most relevant features and a comparison with the best state-of-the-art textual entailment systems is exposed throughout the paper. Finally, the system has been evaluated using the *Second PASCAL Recognising Textual Entailment Challenge* data and evaluation methodology, obtaining an accuracy rate of 61.88%.

1 Introduction

Textual Entailment has been proposed recently as a generic framework for modeling semantic variability in many Natural Language Processing (NLP) applications. An entailment relation between two text snippets (text-hypothesis pair) is produced when the hypothesis' meaning can be inferred from the text's.

Some examples of NLP applications that need to detect when the meaning of a text can be inferred from another one could be the followings. In a Question Answering (QA) system, the same answer could be expressed in different syntactic and semantic ways, and a textual entailment module could help such system to identify the forecast answers that entail the expected one. In other applications such as Information Extraction (IE), the textual entailment tool is applied to different variants that express the same relation. In multi-document summarization (SUM), for instance, we could use such tool to extract the most informative sentences, omitting the redundant information. In general, a textual entailment tool would be useful in order to obtain a better performance in a wide range of NLP applications.

Recognising entailment relations is a very complex task that integrates many levels of linguistic knowledge [2] (i.e. lexical, syntactic and semantic levels). Such

Z. Kedad et al.(Eds.): NLDB 2007, LNCS 4592, pp. 284–294, 2007.

complexity has been proven in the two editions of the *PASCAL Recognising Textual Entailment (RTE) Challenge*[1] [6,3]. These editions of the *PASCAL RTE* have introduced a common task and evaluation framework for textual entailment, covering a broad range of semantic-oriented inferences needed for practical tasks such as the aforementioned applications (concretely QA, IE, Information Retrieval (IR) and SUM). The systems that participated in the challenges used different strategies that combined a wide variety of NLP techniques in order to detect textual entailment. For instance, it is clearly stated that the use of n-grams and subsequence overlap [12,5], syntactic matching [8], logical inference [4,13] and Machine Learning (ML) classification [4,1] is quite appropriate for identifying entailment inferences.

In this paper we propose a system, which we have called *DLSITE-1*, to determine entailment relations based on a wide variety of lexical similarity measures. The aim of using only lexical measures is to achieve a reliable system without need of syntactic and semantic knowledge. Once we have a robust system considering lexical similarities, we will be able to add syntactic and semantic knowledge to it.

The remainder of this paper is structured as follows. The second section details our system and the lexical similarity measures used. The third one illustrates the performed experiments and includes a discussion about the results. Finally, the fourth and last section presents the conclusions of our research and proposes future work.

2 System Description

Our system computes the extraction of several lexical measures from the text-hypothesis pairs, which allow us to determine if the entailment relation is produced. Such measures are basically based on word co-occurrences in both the hypothesis and the text, as well as the context where they appear.

Prior to the calculation of the measures, all texts and hypothesis are tokenized and lemmatized. Later on, a morphological analysis is performed as well as a stemmization,[2] in order to obtain both the grammatical category and the stem for each word belonging to the two snippets. Once these steps are completed, we are able to create several data structures containing the tokens, stems, lemmas, functional[3] words and the most relevant[4] ones corresponding to the text and the hypothesis. Furthermore, having these structures will allow us to know which of them are more suitable to recognize entailment.

In the following paragraphs we describe in detail the measures applied to the data structures obtained from the previous analysis.

[1] http://www.pascal-network.org/Challenges/RTE/ and http://www.pascal-network.org/Challenges/RTE-2/

[2] We use a Porter stemmer implementation.

[3] As functional words we consider nouns, verbs, adjectives, adverbs and figures (number, dates, etc.).

[4] Considering only nouns and verbs.

- **Simple matching:** word overlap between text and hypothesis is initialized to zero. If a word (token, stem, lemma or functional word) in the hypothesis appears also in the text, an increment of one unit is added to the final weight. Otherwise, no increment is produced. Finally, this weight is normalized dividing it by the length of the hypothesis, calculated as the number of words, as shown in Equation 1.

$$spMatch = \frac{\sum_{i \in H} match(i)}{|H|} \tag{1}$$

where H is the set of tokens, stems, lemmas or functional words of the hypothesis, and $match(i)$ is computed as follows:

$$match(i) = \begin{cases} 1 & \text{if } \exists \text{ j} \in T \text{ i=j}, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

being T the set of tokens, stems, lemmas or functional words of the text.

- **Levenshtein distance:** it is similar to simple matching. However, in this case we calculate the function $match(i)$ for each element of H as:

$$match(i) = \begin{cases} 1 & \text{if } \exists j \in TLv(i,j) = 0, \\ 0.9 & \text{if } \nexists j \in TLv(i,j) = 0 \wedge \\ & \exists k \in TLv(i,k) = 1, \\ \max\left(\frac{1}{Lv(i,j)} \forall j \in T\right) & \text{otherwise.} \end{cases} \tag{3}$$

where $Lv(i,j)$ represents the Levenshtein distance between i and j. In our implementation, the cost of an insertion, deletion or substitution is equal to one and the weight assigned to $match(i)$ when $Lv(i,j) = 1$ has been obtained empirically.

- **Consecutive subsequence matching:** this measure assigns the highest relevance to the appearance of consecutive subsequences. In order to perform this, we have generated all possible sets of consecutive subsequences, from length two until the length in words (tokens, stems, lemmas or functional words depending on the data structure used), from the text and the hypothesis. If we proceed as mentioned, the sets of length two extracted from the hypothesis will be compared to the sets of the same length from the text. If the same element is present in both the text and the hypothesis set, then a unit is added to the accumulated weight. This procedure is applied for all sets of different length extracted from the hypothesis. Finally, the sum of the weight obtained from each set of a specific length is normalized by the number of sets corresponding to this length, and the final accumulated weight is also normalized by the length of the hypothesis in words minus one. This measure is defined as follows:

$$LCSmatch = \frac{\sum_{i=2}^{|H|} f(SH_i)}{|H|-1} \tag{4}$$

where SH_i contains the hypothesis' subsequences of length i. Also, $f(SH_i)$ is defined as follows:

$$f(SH_i) = \frac{\sum_{j \in SH_i} match(j)}{|H|-i+1} \tag{5}$$

being

$$match(j) = \begin{cases} 1 & \text{if } \exists \text{ k} \in ST_i \text{ k=j}, \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

where ST_i is the set that contains the text's subsequences of length i.

One should note that this measure does not consider non-consecutive subsequences. In addition, it assigns the same relevance to all consecutive subsequences with the same length. Also, the more length the subsequence has, the more relevant it will be considered.

- **Tri-grams:** two sets containing tri-grams of characters belonging to the text and the hypothesis were created. All the occurrences in the hypothesis' trigrams set that also appear in the text's will increase the accumulated weight in a factor of one unit. Finally, the calculated weight is normalized dividing it by the total number of tri-grams within the hypothesis.
- **ROUGE measures:** ROUGE measures have already been tested for automatic evaluation of summaries and machine translation [10,9]. For this reason, and considerifng the impact of n-gram overlap metrics in textual entailment, we believe that the idea of integrating these measures in our system is very appeal. We have implemented these measures as defined in [9]. Next, we will proceed to explain them.
  - **ROUGE-N:** determines an n-gram recall between a candidate hypothesis and the reference text. It is computed as follows:

$$ROUGE-N = \frac{\sum_{gram_n \in H} Count_{match}(gram_n)}{\sum_{gram_n \in H} Count(gram_n)} \tag{7}$$

where n indicates the length of the n-gram ($gram_n$), $Count_{match}(gram_n)$ is the maximum number of n-grams that appear in both the hypothesis and the text, and $Count(gram_n)$ is the number of n-grams within the hypothesis. In our approach, the n-grams are created from the tokens, stems, lemmas and functional words extracted from the text and the hypothesis, and a set of previous experiments determined that the most suitable values for n are two and three.

- **ROUGE-L:** prior to calculating this measure, we obtained the longest common subsequence (LCS) between the hypothesis and the text, defined as $LCS(T, H)$. The LCS problem consists in finding the longest sequence which is a subsequence of all sequences in a set of sequences[5]. Later on, we applied an LCS-based F-measure to estimate the similarity rate as follows:

$$\begin{aligned} R_{LCS} &= \frac{LCS(T,H)}{|T|} \\ P_{LCS} &= \frac{LCS(T,H)}{|H|} \\ F_{LCS} &= \frac{\left(1+\beta^2\right)\cdot R_{LCS}\cdot P_{LCS}}{R_{LCS}+\beta^2\cdot P_{LCS}} \end{aligned} \tag{8}$$

  where $\beta = 1$, and T and H are the sets that contain the tokens, stems, lemmas or functional words corresponding to the text and the hypothesis.

- **ROUGE-W:** is quite similar to the ROUGE-L measure. The difference relies on the extension of the basic LCS. ROUGE-W uses a weighted LCS between the text and the hypothesis, $WLCS(T, H)$. This modification of LCS memorizes the length of consecutive matches encountered considering them as a better choice than longer non-consecutive matches. We computed the F-measure based on WLCS as follows:

$$\begin{aligned} R_{LCS} &= f^{-1}\left(\frac{WLCS(T,H)}{f\left(|T|\right)}\right) \\ P_{LCS} &= f^{-1}\left(\frac{WLCS(T,H)}{f\left(|H|\right)}\right) \\ F_{LCS} &= \frac{\left(1+\beta^2\right)\cdot R_{LCS}\cdot P_{LCS}}{R_{LCS}+\beta^2\cdot P_{LCS}} \end{aligned} \tag{9}$$

  where f^{-1} is the inverse function of f. One property that f must have the is that $f(x+y) > f(x) + f(y)$ for all positive integer values.[6] In our experiments we used $f(k) = k^2$, $f^{-1}(k) = k^{1/2}$ and $\beta = 1$.

- **ROUGE-S:** this measure is based on skip-ngrams. A skip-ngram is any combination of n words in their sentence order, allowing arbitrary gaps. ROUGE-S measures the overlap of skip-ngrams between the hypothesis

[5] Definition extracted from http://www.wikipedia.org/

[6] This property ensures that consecutive matches has more scores than non-consecutive matches.

and the text, $SKIP_n(T,H)$. As the aforementioned ROUGE measures, we compute the ROUGE-S-based F-measure as follows:

$$\begin{aligned} R_{LCS} &= \frac{SKIP_n(T,H)}{C\left(|T|,n\right)} \\ P_{LCS} &= \frac{SKIP_n(T,H)}{C\left(|H|,n\right)} \\ F_{LCS} &= \frac{\left(1+\beta^2\right)\cdot R_{LCS}\cdot P_{LCS}}{R_{LCS}+\beta^2\cdot P_{LCS}} \end{aligned} \tag{10}$$

where $\beta = 1$, C is a combinational function and n is the length of the selected skip-gram. For our experiments we developed skip-bigrams and skip-trigram ($n = 2$ and $n = 3$), due to the fact that higher values of n produced meaningless skip-ngrams.

The whole system's architecture is shown in Figure 1. It illustrates how the different modules interact between them as well as the ML algorithm used to decide whether there is entailment or not. Different ML classifiers were considered, being the Support Vector Machine (SVM) the best one for our needs. We have used the SVM implementation of Weka [14], considering each lexical measure as a feature for the training and test stages.

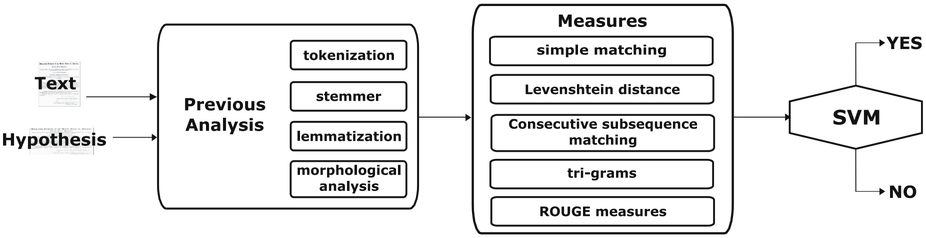


Fig. 1. *DLSITE-1* system architecture

3 Experiments and Discussion

The aim of the performed experiments is to check whether our research on lexical measures as a SVM classifier features can achieve satisfactory results considering that only lexical information is used. In this section we present the evaluation environment and the different sets of features obtained applying a selection process. Later on, we show and analyze the results obtained.

3.1 Evaluation Environment

To evaluate our system we believe that it is appropriate to use the corpora from the two editions of *PASCAL RTE Challenge*. The organizers of this challenge provide participants with development and test corpora, both of them containing

800 sentence pairs (text and hypothesis) manually annotated for logical entailment. It is composed of four subsets, each of which corresponds to typical success and failure settings in different tasks, such as Information Extraction (IE), Information Retrieval (IR), Question Answering (QA), and Multi-document Summarization (SUM). For each task, the annotators selected positive entailment examples (annotated YES), as well as negative examples (annotated NO) where entailment is not produced (50%-50% split). The judgments returned by the system will be compared to those manually assigned by the human annotators. The percentage of matching judgments will provide the *accuracy* of the system, i.e. the percentage of correct responses.

Regarding our system's training stage, we used the development corpus from the first and second edition of RTE, namely RTE-1 and RTE-2, respectively. However, the evaluations were performed using only the test corpus provided in RTE-2. The use of the two development corpora increased the number of significant examples in the training data, and, therefore, also increased the final accuracy rate.

3.2 Feature Selection

The lexical measures implemented in our system provide a set of 45 features. They have been applied to the text-hypothesis pairs, and, concretely, to their respective words, stems and lemmas. In addition, there are two kinds of lexical measures: those that consider only functional words, and those that only take into account nouns and verbs. The mentioned features were processed as a pool of potentially useful features.

In order to select the best features for our system's purpose, we performed a top-down strategy starting with all available features and iteratively removing one of them in each iteration. The removal criterium was the one that had the lowest information gain. The best feature sets generated using the mentioned strategy were the followings:

- **all_features:** initial set containing all features (*simple matching*, *Levenshtein distance*, *Consecutive subsequence matching*, *Tri-grams* and *ROUGE measures* considering tokens, stems, lemmas and functional words extracted from the text and the hypothesis).
- **R1set:** removing from the all_features set the ones obtained by the $ROUGE-S$ measure (when $S = 2$ and $S = 3$).
- **R2set:** R1set without considering the feature derived from the $ROUGE-L$ and $ROUGE-W$ measures.
- **R3set:** R2set but *the simple matching*, *Levenshtein distance*, *Consecutive subsequence matching*, *Tri-grams* and $ROUGE-N$ measures were only applied to tokens, stems and lemmas extracted from the text and the hypothesis.

3.3 Result Analysis

Table 1 summarizes the results obtained with a 10-fold cross validation over the development data and the final system's accuracy using the test corpus provided by RTE-2.

Table 1. Results obtained by the PASCAL RTE-2 evaluation script

	10-fold Cross Validation	Accuracy (test data)				
	overall	overall	IE	IR	QA	SUM
SVM$_{all\_feautures}$	0.5941	0.6062	0.5250	0.6050	0.5400	0.7550
SVM$_{R1set}$	0.5897	0.6062	0.5250	0.6000	0.5450	0.7550
SVM$_{R2set}$	0.5919	0.6088	0.5300	0.6150	0.5400	0.7500
SVM$_{R3set}$	0.6013	**0.6188**	0.5300	0.6300	0.5550	0.7600

As we can observe in the previous table, the differences between feature sets are reduced, being *R3set* the one that achieves better results in both the development and test corpus sets. This fact reveals that the least significant features are produced by the ROUGE measures (except ROUGE-N). In addition, the application of lexical measures to tokens, stems and lemmas obtain better performance than considering functional words or only nouns and verbs.

According to the performed feature analysis and the information gain provided by each one in the training phase, we can deduce that the most significant lexical measures were *Consecutive subsequence matching* and *tri-grams* applied to the tokens and lemmas extracted from the text-hypothesis pair. One should note that these statements depend on the idiosyncrasies of the RTE corpora. However, these corpora are, nowadays, the most reliable for evaluating textual entailment systems.

On the other hand, the fact that the proposed system only uses lexical information reduces its capability to recognise entailment relation. One example could be the pair number 38 from the RTE-2 test corpus, which is shown as follows:

Text: Considering the amount of rain that soaked Riviera, Campbell didn't expect to complete his second round Friday in the Nissan Open.

Hypothesis: Campbell finished his second round Friday.

In this case, the hypothesis' subsequences "*Campbell*" and "*his second round Friday*" match exactly with the text, producing a high lexical similarity value. The lack of semantic knowledge causes that the system suggests true entailment even although the entailment relation does not exist. This deficiency could be solved adding modules that deal with synonyms and negations contributing to establish different meaning to the text and the hypothesis.

Finally, a comparison between the RTE-2 participating systems is exposed. Table 2 shows the results that *DLSITE-1* such systems obtained in the RTE-2 Challenge.

As we can see in Table 2, *DLSITE-1* would have reached the fifth place in the RTE-2 ranking, out of twenty four participants.

The baseline we set for our system was to achieve better results than the ones we obtained with our last participation in RTE-2 (see [11], last row in Table 2). As stated in [11], our previous system obtains a semantic similarity score by means of logic forms derived to the dependency trees from the pair text-hypothesis and WordNet. However, although its results were promising we desired to improve

Table 2. Comparative evaluation within the RTE-2 environment

System	Accuracy (test data)				
	overall	IE	IR	QA	SUM
(Hickl et. al, 2006) [7]	0.7538	0.7300	0.7450	0.6950	0.8450
(Tatu et. al, 2006) [13]	0.7375	0.7150	0.7400	0.7050	0.7900
(Zanzotto et. al, 2006) [15]	0.6388	–	–	–	–
(Adams, 2006) [1]	0.6262	0.505	0.595	0.685	0.720
DLSITE-1	0.6188	0.5300	0.6300	0.5550	0.7600
(Bos et. al, 2006) [4]	0.6162	0.505	0.660	0.565	0.735
...					
(Ferrández et. al, 2006) [11]	0.5563	0.4950	0.5800	0.6100	0.5400

them tackling the recognision of textual entailment from a concrete setting (in this case a lexical setting). This approach allows us to achieve good result considering only lexical information and, subsequently add others kinds of information (e.g. syntactic and semantic) in order to improve the system.

In addition, we would like to emphasize the fact that all systems shown in Table 2 used more knowledge than the information which could be provided by lexical measures. For example, in [1] the author uses WordNet in order to obtain the lexical relation between two tokens as well as a negation detector. The approach in [7] combines lexico-semantic information obtained by a large collection of paraphrases and NLP applications (e.g. named entity recognition, temporar/spatial normalization, semantic role labeling, coreference, etc.). Finally, the system exposed in [13] contains a knowledge representation based on a logic proving setting with NLP axioms.

4 Conclusions

The main contribution of this research is the development of a system for solving textual entailment relations considering only lexical information. To achieve this, we implemented and applied a wide variety of lexical measures. The reason why we make use of this amount of measures and information is motivated by the fact that the integration of more complex semantic knowledge is a delicate task as it is demonstrated by the amount of work developed in the last years. Therefore, our goal is to develop a robust system without complex syntactic-semantic knowledge. Such expertise may be added to our approach in a near future.

In a nutshell, *DLSITE-1* is a textual entailment system that deals with the entailment phenomenon from a lexical point of view, applying relevant lexical measures to deduce entailment relations. It successfully overcomes the RTE task achieving overall accuracy rates higher than 61%. Based on this, the authors of this paper believe that it is easier to perform the recognition task in three separate levels (lexical, syntactic and semantic) and, afterwards, combine them into a complete system.

As of future work, we are interested in improving our system investigating the addition of syntactic and semantic knowledge. Due to the fact that the system achieves high accuracy rates considering only lexical similarities, the next step would be to integrate different tools and strategies to add other kinds of knowledge, such as syntactic and semantic. For instance, we could use resources to generate syntactic dependency trees and obtain similarities between them, including modules that process synonyms and other semantic relations. In addition, extraction of speech knowledge representations by means of techniques based on named entity recognition, co-references and role labeling could be an important improvement.

Moreover, we would like to emphasize that, although the proposed lexical similarity measures need some language dependent tools (e.g. lemmatizer, stemmer and morphological analyzer), the system could be easily ported to other languages. This research line would represent possible future work as well.

Acknowledgments

This research has been partially funded by the Spanish Government under project TIN2006-15265-C06-01 and by the QALL-ME consortium, which is a 6th Framenwork Research Programme of the European Union (EU), contract number: FP6-IST-033860. The authors would like to thank the EU for the financial support and the partners within the consortium for a fruitful collaboration. For more information about the QALL-ME consortium, please visit the consortium home page, http://qallme.itc.it/.

References

1. Adams, R.: Textual Entailment Through Extended Lexical Overlap. In: Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment, pp. 128–133, Venice, Italy (April 2006)
2. Bar-Haim, R., Szpecktor, I., Glickman, O.: Definition and analysis of intermediate entailment levels. In: Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment, pp. 55–60, Ann Arbor, Michigan, June (2005)
3. Bar-Haim, R., Dagan, I., Dolan, B., Ferro, L., Giampiccolo, D., Magnini, B., Szpektor, I.: The Second PASCAL Recognising Textual Entailment Challenge. In: Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment, pp. 1–9 (April 2006)
4. Bos, J., Marker, K.: When logical inference helps determining textual entailment (and when it doesn't). In: Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment, pp. 98–103, Venice, Italy (April 2006)
5. Clarke, D.: Meaning as Context and Subsequence Analysis for Entailment. In: Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment, pp. 134–139 (April 2006)
6. Dagan, I., Glickman, O., Magnini, B.: The PASCAL Recognising Textual Entailment Challenge. In: Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment, pp. 1–8, Southampton, UK (April 2005)

7. Hickl, A., Bensley, J., Williams, J., Roberts, K., Rink, B., Shi, Y.: Recognizing Textual Entailment with LCC's GROUNDHOG System. In: Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment, pp. 80–85, Venice, Italy (April 2006)
8. Kouylekov, M., Magnini, B.: Tree Edit Distance for Recognizing Textual Entailment: Estimating the Cost of Insertion. In: Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment, pp. 68–73, Venice, Italy (April 2006)
9. Lin, C.-Y.: ROUGE: A Package for Automatic Evaluation of Summaries. In: Moens, S.S.M.-F. (ed.) Text Summarization Branches Out: Proceedings of the ACL-04 Workshop, pp. 74–81, Barcelona, Spain, Association for Computational Linguistics (July 2004)
10. Lin, C.-Y., Och, F.J.: Automatic Evaluation of Machine Translation Quality Using Longest Common Subsequence and Skip-Bigram Statistics. In: ACL, pp. 605–612 (2004)
11. Ferrández, Ó., Terol, R.M.: Muñoz, R., Martínez-Barco, P., Palomar, M.: An approach based on Logic forms and wordnet relationships to textual entailment performance. In: Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment, pp. 22–26, Venice, Italy (April 2006)
12. Pérez, D., Alfonseca, E.: Application of the Bleu algorithm for recognising textual entailments. In: Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment, pp. 9–12, Southampton, UK (April 2005)
13. Tatu, M., Iles, B., Slavick, J., Novischi, A., Moldovan, D.: COGEX at the Second Recognizing Textual Entailment Challenge. In: Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment, pp. 104–109, Venice, Italy, April (2006)
14. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)
15. Zanzotto, F.M., Moschitti, A., Pennacchiotti, M., Pazienza, M.T.: Learning textual entailment from examples. In: Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment, pp. 50–55, Venice, Italy (April 2006)

Text Segmentation Based on Document Understanding for Information Retrieval

Violaine Prince and Alexandre Labadié

LIRMM,
161 rue Ada 34392 Montpellier Cedex 5, France
{prince,alexandre.labadie}@lirmm.fr

Abstract. Information retrieval needs to match relevant texts with a given query. Selecting appropriate parts is useful when documents are long, and only portions are interesting to the user. In this paper, we describe a method that extensively uses natural language techniques for text segmentation based on topic change detection. The method requires a NLP-parser and a semantic representation in Roget-based vectors. We have run the experiment on French documents, for which we have the appropriate tools, but the method could be transposed to any other language with the same requirements. The article sketches an overview of the NL understanding environment functionalities, and the algorithms related to our text segmentation method. An experiment in text segmentation is also presented and its result in an information retrieval task is shown.

1 Introduction

Information retrieval needs to match relevant texts with a given query. The latter is seldom expressed as a sentence, but most frequently as a set of key-words, all in natural language (NL). If several research works have been dealing with the problem of matching the query content with available documents (on the Web for instance), the issue we are here focusing on is how to provide the user, not only with the relevant document, but with the most appropriate fragments of this document relevant to his/her queries.

Selecting appropriate parts is useful when documents are long, and only portions are interesting to an user. This approach has already been proned by ([22], [8], [4], [14]) and many other works. Two major techniques are to be applied :

- Looking for the fragment that contains the biggest set of words of the query, and selecting the n sentences containing it ([22],[15]). It provides the involved segments, but could be silent about portions, semantically related to the query as consequences or causes, that do not directly contain the query keywords.
- Segmenting the retrieved text into parts that are topically based, and matching the query content with these segments ([8]) being one of the first to suggest it.

Z. Kedad et al.(Eds.): NLDB 2007, LNCS 4592, pp. 295–304, 2007.

In this paper, we describe a method belonging to the second category. Its advantage is that it extensively uses NL techniques for text segmentation based on topic change detection. This means that it undertakes a task of document understanding. Its advantage in information retrieval is that it might select as relevant text fragments semantically and topically related to the query, about which word-based methods are silent.

Since the method needs an NL-parser and a semantic representation in Roget-based vectors (a first major use of Roget-based representations in NL processing is described in ([21]), we have run the experiment on French documents, for which we have the required NL-environment. Transposition for English or other languages could be made with the appropriate parsers.

In section 2 we describe text segmentation as an issue, briefly browsing its state-of-the-art, related to information retrieval, and present the grounds on which our method is founded. In section 3 we provide an overview of our NL understanding environment functionalities, and the algorithms related to our text segmentation method. In section 4 we describe an experiment in text segmentation, and show its output to queries. Finaly we conclude about the accomplished research and its possible extensions for information retrieval.

2 Topical Text Segmentation

2.1 What Is Text Segmentation

Topic based text segmentation consists in finding, inside a text, sentences that will be borderlines of topical segments. There are three main approaches to detect these sentences :

- Similarity based methods, which measure proximity between sentences by using (most of the time) the cosine of the angle between vectors representing sentences. The c99 algorithm ([1]) for example uses a similarity matrix to generate a local classification of sentences and isolate topical segments.
- Graphical methods, which graphically represent terms frequencies and use these representations to identify topical segments (which are dense dot clouds on the graphic). The Dotplotting algorithm ([19]) is the most common example of the use of a graphical approach of text segmentation.
- Lexical chains based methods, which links multiple occurrences of a term and consider a chain is broken when there are too much sentences between two occurrences of a term. Segmenter ([12]) uses this methods for text segmentation with a subtle adjustment as it determine the number of necessary sentences to break a chain in function of the syntactical category of the term.

These methods are all word / term based, and so view thetext as a "bag of words". If they can help retrieving relevant segment of text in big documents, they cannot solve the problem of relevant segments not using the same lexical field than the query.

2.2 Text-Segmentation Based Approaches in Information Retrieval

Since text segmentation could be associated to the fact that segments could be named and indexed, it was an evidence that text segmentation was a requirement for information retrieval. However, a great majority of the recent literature is devoted to word segmentation as a major issue and not to topical fragments retrieval. Also, most recent papers are related to languages like Chinese where word segmentation is a real ambiguous problem ([9] is, in this respect, one of the most cited papers in the domain). Among this literature, [20] suggest to detect indexing segments in Chinese texts, with a heuristic based method that outperformed the boundary method previously used (a method close to lexical chains).However, their method is limited to words, and does not undertake a complete document understanding. Topic change detection methods applied to information retrieval are present in many works inspired from NL processing, mostly in the preceding decade. [18] describe a topic-based text segmentation. [7] suggests a text tiling algorithm detecting subtopics of a given topic. In the same year, [10] insist on redefining segments retrieval methods. Nevertheless, all these methods are lexically based, either on lexical cohesion determination ([16]) or on lexical chains delimitation. The few methods that enlarge the horizon of topic change detection towards discourse function (style, syntax, etc.) are found in ([11]) for stylistic variation. The latter is also used in multimedia information retrieval especially with speech and speaker recognition. More oriented towards syntax, [17] describe a grammar-based method for discourse partitioning.

One of the limitations to be found with the most popular methods in topic change detection is that although lexical cohesion, as a ground assumption, is a good candidate for defining a topic it has the following drawbacks:

– Most words of any natural language are polysemous. Their multiple meanings are only disambiguated by understanding the sentences they are in, because sentences are a natural way to select a word meaning for a human reader/speaker. So a representation of the word as modulated by the sentence is necessary to constrain word sense disambiguation, a thing that word-based methods tend to neglect.
– A text segment might be related to a topic by directly using the words that are prototypical of this topic. Describing the consequences of an action might not necessarily contain the action name. So word-based methods overlook these segments in their passages retrieval.

2.3 Requirements for an Adapted Text Segmentation

Text segmentation, to be useful in an information retrieval task, doesn't need to have very precise topical boundaries, so boundary based methods such as [19] are not necessarily the most adapted. Note that is was also a result found by [20] for their word segmentation for information retrieval. The rationale is that one or two sentences of margin won't significantly affect information retrieval performances, from the user point of view. But to really improve results on this kind of task, a text segmentation method should :

- Represent text segments in a simple and lexically independent way : lexical dependency might burden information retrieval with side effects such as polysemy (introducing noise) or synonymy (introducing silence). The previous subsection discussions show that representations of syntax (for the sentence) and discourse relations are necessary to retrieve the best segments (note that [4] reaches a similar conclusion for enriching the idea of local coherence).
- Allow to match topically close segments together: matching methods are several; they could be by measuring length (on chains), similarity (on vectors), or distance (in bayesian networks in clustering methods).
- Allow to match queries with text segments.

It, indeed, also needs to find cuts between segments, but, as it is said before, fuzzy boundaries should work as well as precise ones.

In the next section we present a tool based on NL processing. It detects topic coherence by using a deep syntactic analysis employed as an input for semantic calculus of the sentence. The local topics of the sentence are thus determined as related to concepts defined in a thesaural ontology. Afterwards, each sentence is agglutinated to the preceding and a new calculus is performed. Topic change is detected when a new sentence (or a new bundle of sentences) strongly differs from the preceding one. Therefore, document understanding and segments topic comparison are performed by the environment and method described hereafter.

3 A Natural Language Environnement for Topic Change Detection

3.1 A NL Parser Providing Syntactic and Semantic Text Analysis

The NL environment we use is composed of a parser that provides constituents and dependencies in the sentence. Constituents are words that have a part-of-speech atomic tag such as Noun, Verb, Adjective, and so forth, but also sets of words labelled with compound tags such as Noun Phrase, Verb Phrase, Prepositional Noun Phrase, etc. Dependencies are relations between constituents that determine semantic and syntactic functions in the sentence. Subject, Object, Complement are the basic dependencies. They tend to express a notion of government (defined by Chomsky) thus showing that constituents are not equal in importance as semantic elements in a given sentence. The impact of dependencies on defining the general semantics of a sentence is great, and might influence the relevance of this sentence to a given topic (and onwards, to a given query). For instance, if the word "doctor" belongs to a query and appears in a given sentence of a given text as a very secondary complement, the semantic impact of this word on the sentence meaning is weak. Therefore, retrieving this sentence as a core for a relevant text segment would be introducing noise. Whereas if the word is central and governor (like a subject, or sometimes an object), then the sentence containing it could be seen as an interesting candidate for relevance to the query.

The Parser in a Nutshell. The parser we use is called SYGFRAN([3]) and works with 12,000 rules written with a Markov's algorithm formalism. Its characteristics, calculated on a French corpus of 300,000 sentences of an average of 25 words each are given in table 1. SYGFRAN guaranties a 34% precision in complete sentence analysis for any corpus (it has been run on several different corpora and the ratio does not change). However in all other cases, SYGFRAN is not silent : it provides a recall of 85% in partial sentence analysis. Both measures are intimately related to dependencies detection measures.

Table 1. SYGFRAN parser measures

	Recall	Precision
Constituents detection (atomic and compound)	100%	97%
Dependencies detection	85%	34%
Partial sentence analysis	85%	85%
Complete Sentences deep and surface analysis	34%	34%

Semantic Calculus. This parser calculates a semantic representation of the sentence based on a vector representation. Vectors are inspired from the Roget approach in NLP, which has already been proved as interesting for lexical semantics ([21]) and corpus based research ([5]). All words of the language are represented in a dictionary as vectors in a space of a fixed dimension of 873 for French (1000 for English). The basic 873 (respectively 1000) are organized as a conceptual ontology defined in ([24]) and every word is indexed by one or many elements of this ontology.

The technique for calculating each sentence vector is based on calculating each constituent vector, and then using dependencies to define the impact of each constituent on the sentence meaning ([2]). Once the impact defined, the sentence vector is calculated by linear combination of constituent vectors. The more the constituent has impact on the sentence meaning, the more weight it will have in the linear combination (for example, verbs will be more important than adjectives).

A segment vector is defined as a centroid of the sentence vectors it contains (centroids are already used in the domain in ([6]) : among centroid possibilities, we choose to represent segment vectors by a barycenter. But we apply different weights on sentence vectors depending on the position of the sentence represented in the segment. First sentences of the segment have great weights, and weights decrease progressively as we advance in the segment. In a classical structure of an argumentative paragraph, first sentences carry the main subject (topic) of the paragraph and as we advance, we encounter examples and explanations. Finally, a "good" argumentative paragraph ends with some transition sentences which conclude the current paragraph and introduce the next one. We supposed that a topical segment should have the same structure.

Thematic Distance. Most methods using a vectorial representation of text use the cosine as a similarity measure. We preferred to use the angular distance, which we call thematic distance in this case, to compare sentences or text segments. So the thematic distance between X and Y should be :

$$D_A(X,Y) = \arccos \frac{\boldsymbol{X}.\boldsymbol{Y}}{|\boldsymbol{X}|.|\boldsymbol{Y}|} \tag{1}$$

Where X and Y are vectorial representation of sentences. This measure seems better to us for two main reasons :

- It is a mathematical distance. So we can use it more freely than the cosine that is, most of the time, wrongly considered as probability.
- It is a decreasing function which is strongly non-linear between $\frac{\pi}{4}$ and 0. This property is very interesting in our case, because it allows us to be more precise when vectors are close.

The thematic distance will help us finding text segments and frontiers during the segmentation process, but also identifying relevant segments of text.

3.2 Defining Topical Text Segmentation in This Environment

Our segmentation method use the thematic distance and the vectorial representation of segments and sentence to identify what we call "transition zones". We made the hypothesis that topics' boundaries aren't, most of the time, standalone sentences, but small group of sentences concluding the previous topic and introducing the next one. So we can represent two successive topic segment as in Figure 1.

To detect transition zones, we use a window which slides along the text and gives to the sentence in the middle of the window a value called transition value. This value is calculated by considering the first half of the window (current sentence excluded) as a topical segment and the second half as well (current sentence included). We calculate the centroid of each supposed segment and then the thematic distance between them. This distance becomes the transition value of the current sentence.

The transition value of each sentence is compared to a threshold value that has been learnt on three different thematic corpora (law, computer science, and political discourses) of respectively 433456, 4722 and 303373 sentences provided in the evaluation campaign DEFT 2006, as proposed by [13]. What these authors seemed to hint at is that the threshold behaves a sort of a constant in text building. For a given domain, topical units tend to have a more or less fixed amount of sentences. If more, they tend to split into sub-topics. If the transition value is higher than the threshold, we consider this sentence to be a candidate for a transition. If two or more consecutive sentences are higher than the threshold value then we have a transition zone. Transitions zones, and their computing are illustrated in Figures 2 and 3.

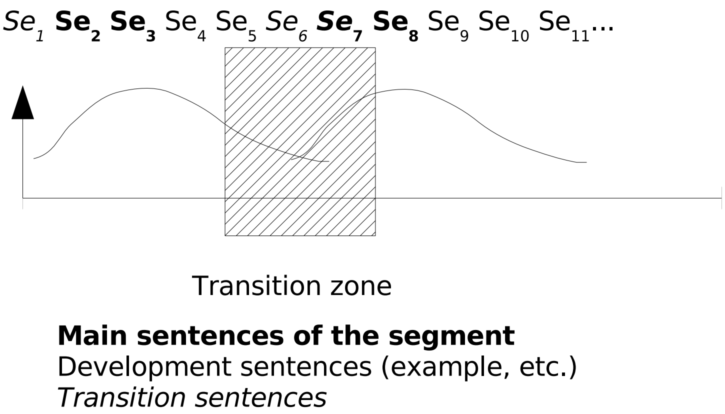

Se1 Se2 Se3 Se4 Se5 Se6 Se7 Se8 Se9 Se10 Se11...
Transition zone
Main sentences of the segment
Development sentences (example, etc.)
Transition sentences

Fig. 1. Topical Structure in the sentence s stream

Finding the right boundary sentence in the transition zone can be done by many means, the simplest (and the one we used here) is to select the sentence with the highest transition value. Other methods are currently experimented.

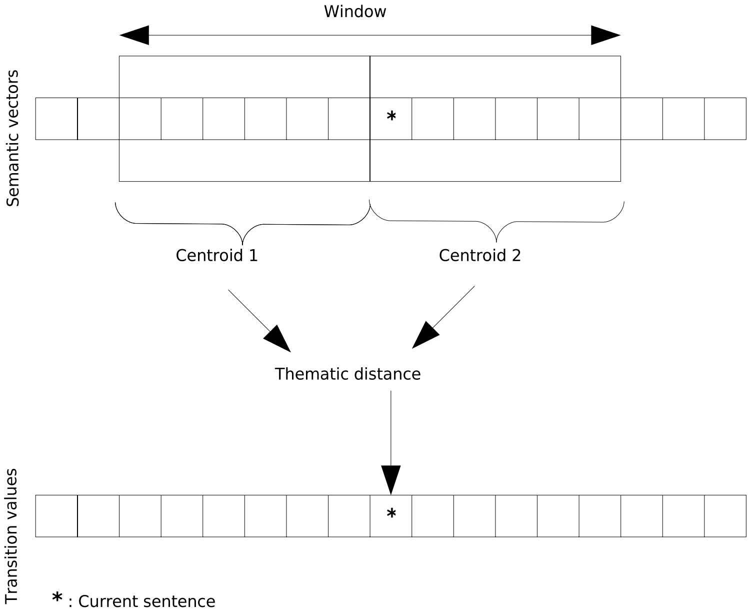


Fig. 2. Computing Transition Zones for Topic Change Detection, 1st step

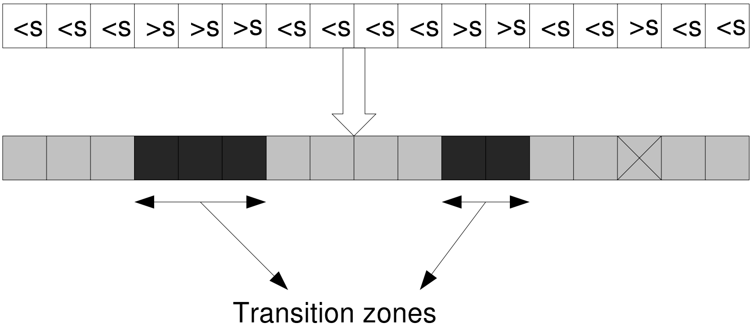


Fig. 3. Computing Transition Zones for Topic Change Detection, 2nd step

4 Experimenting on a Corpus with Queres

During the segmentation task, the centroid of each identified segment is saved. Finding the right segments only means comparing the semantic vector of the query with the centroid of each segment using the thematic distance and the threshold which has been learnt on the corpora described in subsection 3.2. To evaluate the capabilities of our approach as a question-answer system, we have used a corpus of about 15,000 sentences, with an average of 27 words per sentence. Its domain is law texts, and it served as a test corpus for the DEFT06 evaluation conference on text segmentation. Our questions were the following: 1. What are the official languages in Europe? 2. What is the regulation concerning employment in the nuclear industry? 3. Which are the rules of formation of a limited company? 4. What is the regulation concerning the marketing of medical drugs?

4.1 The Method

Each question, given as one or more sentences in natural language, was projected in the vector space and its semantic vector calculated (see section 3). The corpus is already segmented, before entering the query-answer evaluation, and independently from the query contents. The idea is to compare the semantic vector of the query with all detected segments (with a transition of 2 sentences), and retrieve those segments whose angular distance with the query does not exceed 0.8. This value roughly corresponds to an angle of 45 ($\frac{\pi}{4}$) or less, which is half the maximum possible angular distance according to the formula given before. If two vectors make an angle of 45 and less, they are considered to be relatively close to each other. Transposed as a relationship between query and fragments, this means that the fragment is (semantically, topically) relevant to the query. The closer to 0 the angle is, the more relevant the fragment is.

4.2 The Results

Obtained results are summarized in table 2. The segmentation evaluation was made by the organizers of DEFT06 competition, so we just reproduce the values relative to the law test corpus (two other corpora of different domains were provided). The evaluation of segments relevance to queries was made by another group of persons. The idea was the following: a segment was considered as lacking if the human jury considered this segment as relevant to a query and not provided by the system (this played on the recall percentages). The segment was considered as totally relevant and scored 1 in the total if it was a very close or exact answer to the question. It was considered as partially relevant, and score 0.5 in the total if it was sufficiently related to the query to be seen as "interesting". Both values affected the precision percentages.

Table 2. Segmentation and Query-answer results

	Recall	Precision
Segmentation results	**0.806**	0.164
Question 1	0.666	0.518
Question 2	0.16	1
Question 3	0.96	0.36
Question 4	0.29	0.18

5 Conclusion

First obtained results are encouraging. The advantages of this approach could be listed as follows: (1)a query could be a big fragment or a small one, a question or a text, this doesn?t temper with the fragment retrieval method. (2) Fragments containing other words than those in the query have been retrieved. They were judged partially and sometimes totally relevant by the human jury. With a word-based method, they would have been discarded. (3) Small fragments have been retrieved, which is much easier to read for a human user, and the ?informative power? of these fragments is higher than a big text into which the relevant part is littered with irrelevant segments.

Moreover, numerical results don't show some interesting links established during the process. The method, sometimes, bing back sentences and segment which aren't "officially" answer to the question, but that make sense.

References

1. Choi, F.Y.Y.: Advances in domain independent linear text segmentation. In: Proc. NAACL-00 2000, pp. 26–33 (2000)
2. Chauché, J., Prince, V., Jaillet, S., Teisseire, M.: Classification Automatique de Textes partir de leur Analyse Syntaxico-Smantique Proc. 12th International Conference on Natural Language Processing (TALN), pp. 55–65 (2003)
3. Chauché, J.: Un outil multidimensionnel de l'analyse du discours. In: Proc. Coling'84, pp. 11–15 (1984)

4. Chan, S.W.K.: Using heterogeneous linguistic knowledge in local coherence identification for information retrieval Journal of Information Science, pp. 313–328 (2000)
5. Ellman, J., Tait, J.: Roget's thesaurus: An additional Knowledge Source for Textual CBR? In: Proc. 19th SGES Int. Conf. on Knowledge-Based and Applied AI, pp. 204–217 (1999)
6. Eui-Hong, H., Karypis, G.: Centroid-Based Document Classification: Analysis and Experimental Results. In: Zighed, A.D.A., Komorowski, J., Żytkow, J.M. (eds.) PKDD 2000. LNCS (LNAI), vol. 1910, pp. 424–431. Springer, Heidelberg (2000)
7. Hearst, M.A.: TextTiling: Segmenting text into multi-paragraph subtopic passages. Computational Linguistics 23, 33–64 (1997)
8. Hearst, M.A., Plaunt, C.: Subtopicstructuring for full-length document access. In: Proc. ACM SIGIR-93, p. 59-68 (1993)
9. Huang, X., Peng, F., Schuurmans, D., Cercone, N., Robertson, S.E.: Applying Machine Learning to Text Segmentation for Information Retrieval. Information Retrieval (2003)
10. Kaszkiel, M., Zobel, J.: Passage retrieval revisited. Proc. Twentieth International Conference on Research and Development in Information Access, pp. 178–185 (1997)
11. Karlgren, J.: Stylistic variation in an information retrieval experiment. In: Proc. NeMLaP-2 Conference (1996)
12. Kan, M., Klavans, J.L., McKeown, K.R.: Linear segmentation and segment significance. In: Proceedings of WVLC-6 (1998)
13. Labadié, A., Chauché, J.: Segmentation thmatique par calcul de distance smantique. In: Proc. DEFT06 (2006)
14. Llopisand, F., Ferrandezand, A., Vicedoand, J.L., Gelbukh, A.: Textsegmentation for efficient information retrieval. In: Proc. CICLing, pp. 373–380 (2002)
15. Moffat, A., Sacks-Davis, R., Wilkinson, R., Zobel, J.: Retrieval of partial documents. Proc. of the Second Text Retrieval Conference TREC-2, pp. 181–190 (1994)
16. Morris, J., Hirst, G.: Lexical cohesion computed by thesaural relations as anindicator of the structure of text. Computational Linguistics, pp. 21–48 (1991)
17. Nomoto, T., Yoshihiko, N.: A grammatico-statistical approach to discoursepartitioning. In: Proc. COLING'94, pp. 1145–1150 (1994)
18. Ponte, J., Croft, B.: Text segmentation by topic. In: Proc. First EuropeanConference on Research and Advanced Technology for Digitial Libraries, pp. 1145–1150 (1997)
19. Reynar, J.C.: Topic Segmentation: Algorithms and Applications. Phd thesis, University of Pennsylvania (1998)
20. Yang, C.C., Li, K.W.: A heuristic method based on a statistical approach for chinese text segmentation. Journal of the American Society for Information Science and Technology, pp. 438–1447 (2005)
21. Yarowsky, D.: Word Sense Disambiguation Using Statistical Models of Roget's Categories Trained on Large Corpora. In: Proc. Coling'92, pp. 454–460 (1992)
22. Salton, G.: Automatic text processing: the transformation, analysis, and retrieval of information by computer. Addison-Wesley Longman Publishing Co, Boston, MA, USA (1989)
23. Roget, P.: Thesaurus of English Words and Phrases. Longman London, London, England (1992)
24. Larousse, Thesaurus Larousse Larousse Paris, France (1992)

Named Entity Recognition for Arabic Using Syntactic Grammars

Slim Mesfar

LASELDI, Université de Franche Comté,
25030, Besançon, France
mesfarslim@yahoo.fr

Abstract. Named entities (NE) occur frequently in Arabic texts, and their recognition is essential. Recognizing and categorizing NE requires both internal (morphological) and external (syntactic) evidences. This paper describes a system that combines a morphological parser and a syntactic parser, that are built with the NooJ linguistic development environment.

Keywords: Named entity recognition, NER, Arabic, morphological analysis, gazetteers, syntactic grammars, NooJ.

1 Introduction

Named Entity recognition involves processing a text and identifying certain occurrences of words or expressions as belonging to particular categories of Named Entities (NE). A NE recognition system is considered as an important preprocessing tool for tasks such as document classification or clustering, machine translation (MT) information extraction (IE), information retrieval tasks, indexing and search and other text processing applications [8].

Traditionally[1], Named Entity Recognition (NER) refers to the recognition and categorization by type of person names, organizations, geographical locations, numerics such as percentages, as well as times/dates.

2 NER Definition

According to MUC, we distinguish three types of entities to be recognized and categorized:

- **TIMEX:** temporal expressions of time and dates.
- **NUMEX:** numerical expressions of percentage, height, monetary expressions, etc.
- **ENAMEX:** proper names. We distinguish, at least, 3 sub-categories:

[1] As defined within the Message Understanding Conferences (MUC) and the Multilingual Entity Task (MET) conferences.

Z. Kedad et al.(Eds.): NLDB 2007, LNCS 4592, pp. 305–316, 2007.

- Persons: person names "جُون كنيدي" (juwn kinydy - John Kennedy)[2],
- Organizations: companies, banks, associations, universities … e.g. "إيرْبُوس" (íyrbuws– Airbus), "يُونِيسْكُو" (yuwniyskuw- Unesco), etc.
- Localizations: typonyms such as names of countries, cities, states, seas, oceans, mountains, rivers … e.g. "فرَنْسَا" (firansaā - France), "بَاريس" (baārîs - Paris …), "البحر الأبيض المتوسّط" (el baħr eláabyad' elmutawassiṭ – Mediterranean Sea).

3 Why a NER System for Arabic?

Given the explosion of Arabic resources, especially on-line, with more than 20,000 Arabic sites on the Web and more than 300 million users, we recognized the necessity of developing an Arabic component for NooJ platform, which would allow us to process and take advantage of this readily available data. We started building Arabic NooJ module with the purpose of providing automatic analysis of texts written in standard Arabic. This module will be used to a better understanding of the Arabic language based on description of its vocabulary and its transformational syntax according to the theories of Chomsky and Harris [3].

As first step of the processing, we used the lexical module of the linguistic platform NooJ [11], described in section 4, for vocabulary formalization and tokenization[3]. Then, we evaluated the lexical coverage of our Arabic module on LASELDI's[4] corpora collected from the Net. These corpora are composed of journalistic articles of the newspaper "Le monde diplomatique"[5] for five years (2001-2006), which include about 150 000 different terms. The lexical analysis, of these corpora, shows coverage of about 93% by our lexical and morphological resources [7].

The unrecognized forms are classified in 3 subsets:

- 8 400 transliterated named entities : proper names of person such as "شِيرَاك" (šîraāk - Chirac) with some derived forms such as "شِيرَاكِيَّة" (šîraākiyyat - Chiraquism), cities such as "مَرْسِيلِيَا" (marsîliyaā - Marseille) and organisations such as "مَايكْرُوسُوفِتْ" (maāyikruwsuwfit - Microsoft),
- About 1400 borrowing terms such as "ميتَافيزيقَا" (mîtaāfîzîqaā - metaphysics),
- 600 spelling mistakes.

This analysis showed that the great majority of unrecognized forms are proper names (names of people, organizations or localities). Although, these unrecognized forms are words or sequences of words, called named entities (NEs), cannot be found in common dictionaries, they encapsulate important information that can be useful for the semantic interpretation of texts.

[2] There are many transliterating systems for Arabic, official or not official. We could say that each author has his own transliterating system. In this paper, we use the UAT, Unified Arabic Transliteration, described in Wikipédia web site.

[3] The Arabic language is a strongly agglutinant language; its morphological analyzer should separate and identify the component morphemes of input words.

[4] LASELDI: LAboratoire de SEmio-Linguistique, Didactique et Informatique, university of Franche-Comté.

[5] www.mondiploar.com

Since there are not yet defined standards when writing or transliterating proper names[6], the simple lookup approach was impossible to adopt. In fact, it is impossible to enumerate all proper names in lists, to collect and maintain these lists, to deal with name variants and finally to resolve the resultant ambiguity. So, we built a named entities recognition system based on the syntactic module of NooJ and using its syntactic grammars. These used local grammars represent predefined rules based on internal and external [4] evidences in named entity recognition where:

- **Internal evidence:** is taken from within the sequence of words that comprise the name, such as the content of lists of proper names (gazetteers).
- **External evidence:** is provided by the context in which a name appears. It takes advantage of:
  - characteristic properties in syntactic relations (nouns, adjectives) with a proper noun. Such properties can be used to provide confirming or criteria-based evidence for a name's category
  - important complementary morpho-syntactic information provided by morphological analysis

The adequacy of this solution was retained within the last MUC conference. It will be developed for Arabic Named Entity Recognition within the developmental environment NooJ, the tool used for identifying and categorizing Arabic NEs.

4 NooJ and Arabic NER System

NooJ is a linguistic developmental environment which can analyze texts of several million words in real time. It includes tools to construct, test and maintain large-coverage lexical resources, as well as morphological and syntactic grammars. Dictionaries and grammars are applied to texts in order to locate morphological, lexicological and syntactic patterns, solve ambiguities, and tag simple and compound words.

NooJ recognizes all Unicode encodings and the runtime code that applies lexical transducers to input strings and is completely language independent. Thus the code that runs the Arabic analyzer is exactly the same code that processes a dozen other languages, including some Romance, Germanic, Slavic, Semitic and Asian languages, etc.

NooJ can build lemmatized concordances of large texts from Finite-State or Context-Free grammars, and can accordingly perform cascading transformation operations on texts, in order to annotate the text, or to generate paraphrases [12]. It's used to perform our Arabic NER system.

Like other Semitic writing systems, Arabic does not exhibit differences in orthographic cases. Unlike English-language mixed-case texts, there is no obvious clue such as initial capitalized letters to indicate the presence of a name constituent. This seems to impose a requirement of understanding of the morphological nature of each token; especially part-of-speech and distributional information (e.g.: Human, Country, Currency …). However, Arabic language is a strongly agglutinant language, most

[6] Sometimes, transcription systems depend on author origin.

forms in Arabic writing can correspond to a succession of one or more prefixes, a radical and one or more suffixes (as described in the next section); its morphological analyzer should separate and identify component morphemes of the input word, labeling them somehow with sufficient information to be useful for the tasks at hand.

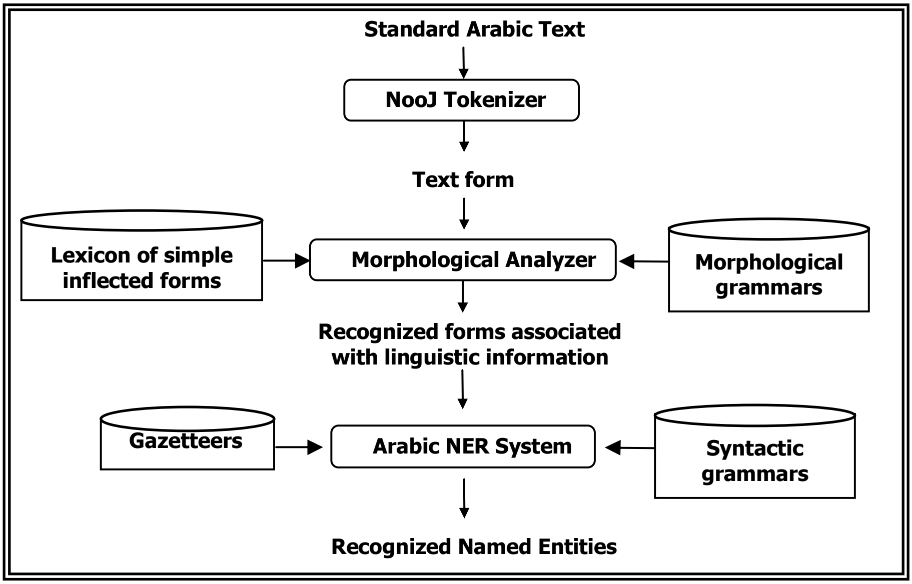


Fig. 1. Architecture of Arabic NER system

Therefore, our Arabic NER system is a two-step process. Initially, we try to collect the maximum of information for contextual recognized forms. Then, this information will be used within syntactic grammars to locate relevant sequences.

4.1 Morphological Analyzer

Our morphological analyzer uses finite state technology to parse vowelled texts, as well as partially and non-vowelled texts. It is based on large-coverage electronic dictionaries as well as morphological grammars covering all grammatical rules and tokenizing agglutinated forms. In agglutination cases, the tokenizer splits the form to identify attached affixes (conjunctions, prepositions, personal pronouns, etc.)

In fact, the morphological capability within NooJ identifies affixes such as conjunctions "وَ"[wa - and] and "فَـ" [fa - and], prepositions "بِـ" [bi – with / in], "لِـ" [li - for], or "كَـ" [ka - as], personal pronouns like "ـهُ" [hu – his], "ـنَا" [naā - our], or "ـهُمْ" [hom - their], as well as their potential combinations. If there was no such tokenization functionality, then these affixes will not be recognized anywhere. This has the effect that trigger words[7], first names or location names with an attached prefix, will not be recognized since the system no longer correctly tokenizes these forms and

[7] Trigger words are described in the next section.

associates them with useful linguistic information. In fact, by means of morphological grammars, we tokenize agglutinated word forms as follows:

- "وَمَدِينَة" (wamadînatu – and the city of) => "وَ " + "مَدِينَة" (wa + madînatu – and + the city of), where "مَدِينَة" (madînatu – the city of) is a trigger word.
- "بِبَارِيس" (bibaārîs – in Paris) => "بِ" + "بَارِيس " (bi + baārîs – in + Paris), where " "بَارِيس (baārîs – Paris) is a city name.
- "فَلِمُؤَسّسته" (falimoẃassasatihi – and for his company) =>"فَ" + "لِ" + "مُؤَسّسة" + "هِ" (fa +li + moẃassasati + hi – and + for + company + his); where "مُؤَسّسة" (moẃassasati – company) is a trigger word.

Each recognized form is associated by the lookup algorithm of NooJ with a set of linguistic information: lemma, POS tag, gender and number, syntactic information (e.g. +Transitive), distributional information (e.g. +Org), etc. This information is very useful for the next task; for example, rather than listing all inflected forms (singular, dual, plural, masculine, feminine) of key words such as 'طبيب' (tabîb, doctor) we can use the syntax of NooJ regular expressions where <طبيب > (<tabîb>, <doctor>) can designate all these potential vowelled, partially vowelled as well as fully vowelled inflected forms.

4.2 Named Entity Recognizer

The NER system within NooJ is based on the use of some knowledge sources:

Gazetteers: They are lexical marker lists, containing names that are known beforehand and have been classified into named-entity types. Lists of names are employed for locations, personal titles, organizations, dates/times and currencies. The following lists of names are used:

Table 1. Gazetteers content

Gazetteers	Content	Examples	Transliteration	Translation
Person names	12400 entries: Arabic first names and transliterated foreign ones	محمد	muhammad	Mohamed
		عَبْد الله	Åbd àllăh	Abdullah
		جُون	juwn	John
		جُونَاتَان	juwnaātaān	Jonathon
Location names	5,038 entries: countries, cities[8], seas, mountains, rivers, etc.	فرَنْسَا	firansaā	France
		بَاريس	baārîs	Paris
		البحر الأبيض المتوسّط	el baħr eláabyad' elmutawassiț	Mediterranean Sea
		جِبَال الألب	Jibaāl el àlib	The Alps
Organization names	250 entries: companies, associations,etc	مَايكْرُوسُوفِت	maāyikruwsuwfit	Microsoft
		إيرْبُوس	íyrbuws	Airbus
		أُوبِيك	úwbîk	OPEC

[8] We listed the major cities and states in the world with a population of more than 100,000.

Table 1. (*continued*)

Personal titles	50 titles	الدكتور	eddoktuwr	Doctor
		السيّد	essayyid	Mr.
Currency units	175 currencies and their subdivisions	يُورُو	yûrû	Euro
		دُولَار	dûlaār	Dollar
		سنتيم	sentîm	Centime
Temporal expressions	lists of day names, month names[9], etc.	الِاثْنَيْن	el ìŧnayn	Monday
		يناير	yanaānir	January
		أكتوبر	octuwbir	October

We also use lists of trigger words which indicate that the surrounding tokens are probably named entity constituents and may reliably permit the type or even the subtype of the named entity to be determined (e.g. religious and political terms are subtypes of person names category) [14]. Lists of triggers were produced manually.

Table 2. Trigger word lists content

Trigger words	**Lists**	**Examples**	**Transliteration**	**Translation**
Person names[10]	354 nationalities and gentilics[11]	الفرنسيّ	al firansiyy	The French
	296 professions	الطّبيب	attabîb	The doctor
	65 political functions	الرّئيس	arraýîs	The president
	52 military titles	القُبْطَان	al qubtaān	The captain
	24 religious titles	البَابَا	al baābaā	The Pope
	23 sports	المصارع	al muşaāriå	The fighter
Localizations	32 trigger words	مَدِينَة	madînat	the city of
		جَبَل	jabal	the mountain
		عَاصِمَة	åaāsimat	the capital of
Organizations	26 trigger words	مؤسسة	mowássasat	company
		جمعية	jamåiyya	association

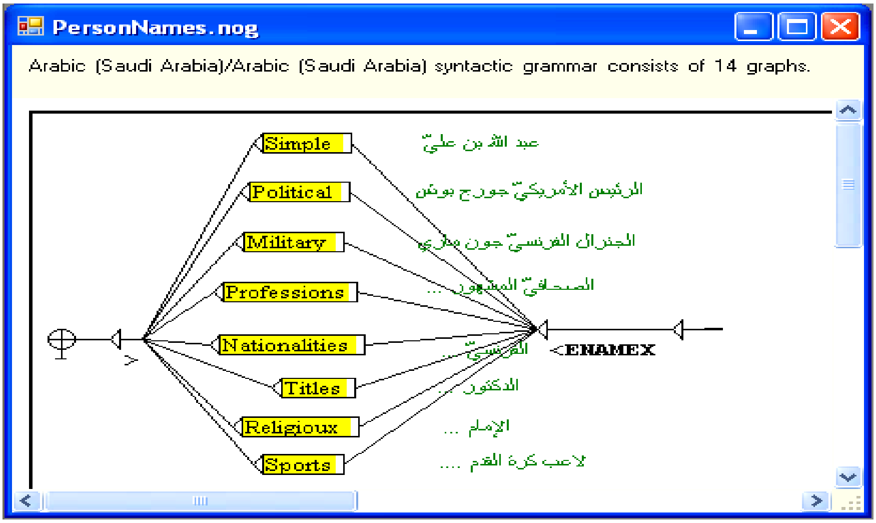
The above key words are tagged as result of the morphological analysis, and are used in Named entity grammar rules.

[9] For month names, we listed 7 different lists of calendar months for:
-Hijri: شَوَّال, شَعْبَان, رَمَضَان, رَجَب, رَبِيع الْأَوَّل, رَبِيع الثاني, ذُو القَعْدَة , ذُو الْحَجَّة, جُمَادَى الْأُولَى,جُمَادَى الثاني,
مُحَرَّم, صَفَر.
-Arabic: ديسمبر, نوفمبر, أكتوبر, سبتمبر, أغسطس, يوليو, يونيو, مايو, أبريل, مارس, فبراير , يناير.
-Syrian: كانون الأول, تشرين الثاني, تشرين الأول, أيلول, آب, تموز, حزيران, أيار, نيسان, آذار, شباط , كانون الثاني.
-Tunisian/Algerian: ديسمبر, نوفمبر, أكتوبر, سبتمبر, أوت, جويلية, جوان, ماي, أفريل, مارس, فيفري , جانفي.
-Libyan: الكانون, الحرث, التمور, الفاتح, هانيبال, ناصر, الصيف, الماء, الطير, الربيع, النوار , أين النار.
-Mauritanian: دجمبر, نوفمبر, أكتوبر, شتمبر, أغشت, يوليو, يونيو, مايو, إبريل, مارس, فبراير , يناير.
-Moroccan: دجمبر, نونبر, أوكتوبر, شتنبر, غشت, يوليوز, يونيو, ماي, أبريل, مارس, فبراير , يناير.

[10] We distinguish eight subtypes of person names, as shown in the grammar given in fig. 2.

[11] A gentilic or a demonym is a word that denotes the inhabitants of a place.

Grammars: They are compiled into Finite-State transducers, Context-Free grammars (stack automata) and eRTNs (enhanced Recursive Transition Networks). A syntactic grammar[12] represents word sequences described by manually created rules, and then produces some kind of linguistic information such as type of the recognized NE.



Arabic (Saudi Arabia)/Arabic (Saudi Arabia) syntactic grammar consists of 14 graphs.

Fig. 2. ENAMEX NooJ syntactic grammar

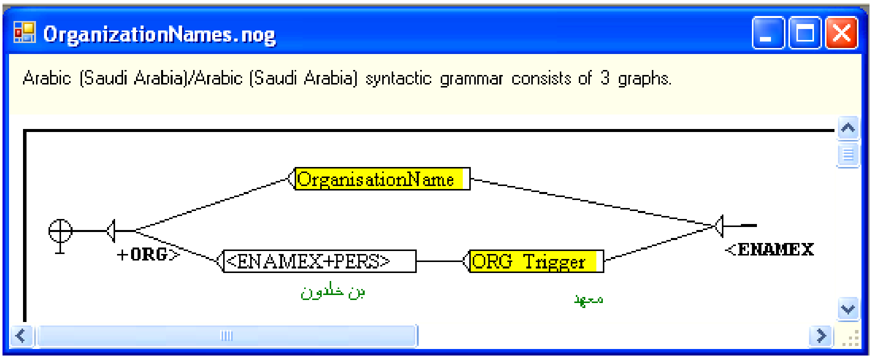
A grammar rule is generally made of, at least, a trigger word, some tagged words and occasionally unknown words in order to group together elements pertaining to the same entity. Sequences of words can be accurately tagged given an appropriate context especially if a trigger word or an entry from gazetteers disambiguates the sequence.

The preponderance of unknown words within NEs induces a lack of information; added to problems of determination of stop words that allow knowing where to stop, which increases boundary errors. NooJ syntactic grammars respect some heuristics when applying rules. They locate the "longest match" for one grammar and "all matches" for the whole of grammars.

- **Person names (ENAMEX+PERS):** The majority of NE rules are for personal names since care needs to be taken when describing potential combinations of first names, family names, person titles, functions or professions, in addition to some special lexical elements involved by Islamic names such as "اِبْن" (ibn – the son of), "بن" (bin – the son of), "أَبُو" (abù – the father of), etc. Concerning these Islamic names, we surprisingly found little use of such person names in our journalistic corpora. Details of a sub-graph of the Named Entity Recognition transducer are given in **Appendix1**.

[12] We give an example of a syntactic grammar in fig. 2.

- **Organization names (ENAMEX+ORG):** There is a good number of rules for organization names. They may contain any other proper names (such as person names, location names) as well as trigger words, and their combinations. Since, in NooJ, we can assign priority to a grammar over other ones and use previous given annotations. We set a high priority to grammar recognizing personal names and localizations to use already annotated NEs by means of NooJ regular expressions such as <ENAMEX+PERS> which can indicate any person name or <ENAMEX+LOC> which indicates any localization name.



Arabic (Saudi Arabia)/Arabic (Saudi Arabia) syntactic grammar consists of 3 graphs.

Fig. 3. ENAMEX NooJ syntactic grammar (2)

- **Localizations names (ENAMEX+LOC):** Concerning localizations, there are not so many rules for location names because they are recognized mainly in the morphological analysis stage by looking up in the lists of cities, states, countries, and other place names.
- **TIMEX and NUMEX:** Rules for monetary and time expressions have been collected by analyzing some manually collected expressions from our evaluation corpora.

The extracted named entities are displayed into a concordance window to give users a quick overview of the contents of documents. This is particularly useful when applied to large document collections especially when sorting, identifying and filtering out bad concordances, as well as producing statistics on the contents of the whole corpora. This would, also, allow us to extend our gazetteers and syntactic rules in order to enlarge the set of identified expressions.

5 Problems with Arabic NER

In addition to lack of obvious clues such as initial capitalized letters to indicate the presence of a proper name, there are some specific problems related to Arabic NER.

5.1 Non-vocalisation

Non-vocalisation is due to a lack of short vowels in usual texts from which a high degree of ambiguity ensues. In theory, only the Koran, and children’s books are fully vowelled; our automatic analysis allows parsing of fully vowelled, partially vowelled and unvowelled texts.

Non-vocalisation can affect NER system when potential vocalizations can lead to different senses which can designate trigger words for two or more different NE type such as the case of unvowelled form "مؤسسة" [mowássʼsat] that can accept, between others, the two vocalizations[13]:

- "مُؤَسَّسَة" [mowássasat – a company] => trigger word of an organization name.
- "مُؤَسِّسَة" [mowássisat – a founder] => trigger word of a personal name

5.2 Delimitation Problems

Delimitation problems are related to a lack of information about unknown words within NEs, an antonomastic usage where proper names are substituted with a phrase or conversely as well as the presence of some homonyms[14] which increases ambiguity when trying to mark NE constituents such as:

- "أَشْرَف" [achrafa] which can be a first name, an inflected verbal form meaning "he supervised", an elatif adjective which means "the most honorable", etc.
- "أَحْمَد" [ahmadu] which can be a first name, an inflected verbal form meaning "I thanks", etc.

In fact, we delimitated Arabic NEs using morphological information which allows distinctions between likely and unlikely name constituents, which is particularly important when deciding where a name ends and the non-name context begins [5]. So, we recognized that this delimitation of NEs can be based on identification of unlikely name constituents of a proper name such as:

- Invariable words such as preposition, adverbs, etc.
- Inflected forms of verbs such as "يَكْتُبُ" (yaktubu – he writes) except for certain first names; e.g. "يَزِيدُ" (yazîdu – he adds).
- Suffixed forms; e.g. "كِتَابُهُ" (kitaābuhu – his book)
- Some lexical elements such as verbs of speaking such as "قَالَ" (qaāla – to say) or "كَلَّمَ" (kallama – to talk) and cognition verbs such as "عَرِفَ" (årifa - know).
- Forms with subject or object suffixes; e.g. "يُكَاتِبُه" (yukaātibuhu - He will correspond with him).

6 Evaluation

Traditionally, the scoring report compares the answer file with a carefully annotated file. The system was evaluated in terms of the complementary precision[15] (P) and recall[16] (R) metrics. Briefly, precision evaluates the noise of a system while recall

[13] If we consider all declensions, unvowelled form "مؤسسة" [mowássʼsat] can accept ten different vocalizations.

[14] A homonym is a word that has the same pronunciation and spelling as another word, but a different meaning.

[15] Precision is calculated according to formula (1) and (1').

[16] Rappel is calculated according to formula (2) and (2').

evaluates its coverage. These metrics are often combined using a weighted harmonic called the F-measure[17] (F).

$$\textbf{P= \# of correct entities detected / \# of entities detected} \quad (1)$$

$$\textbf{R= \# of correct entities detected / \# of entities manually labeled} \quad (2)$$

$$\textbf{F = 2 P R / P+R} \quad (3)$$

Since we had problems with NE's delimitation, we had to redefine evaluation parameters to take account of partially correct answers [1]. Evaluation metrics become:

$$\textbf{P'= \# of correct + partially correct entities detected / \# of entities detected} \quad (1')$$

$$\begin{gathered}\textbf{R'= \# of correct + partially correct entities detected /}\\ \textbf{\# of entities manually labeled}\end{gathered} \quad (2')$$

$$\textbf{F' = 2 P' R' / P'+R'} \quad (3')$$

The evaluation carried out on part of our corpora of the newspaper "Le Monde Diplomatique", in its Arabic version, gives the following scores:

Table 3. Experiments on journalistic corpora

		Precision : P'	Recall : R'	F-mesure : F'
TIMEX		97%	95%	96%
NUMEX		97%	94%	95,5%
ENAMEX	Person names	92%	79%	85%
	Organizations	90%	78%	84%
	Localizations	82%	71%	76%

7 Conclusion

In this paper, we deal with the description of a system of recognition of proper names, dates, and numerics in standard Arabic text through a combination of a morphological analysis and a rule-based NER system using NooJ syntactic grammars. It is also used to classify unknown proper names and thereby improve the name recognition process and the system coverage.

We are working on evaluation enhancement. On one hand, we need to extend scoring on the totality of our corpora. On the other hand, we have to use co-reference to improve NE results by assigning entity type to previously unclassified names, based on relations with classified NEs. We are also refining our categories to adopt more precise categorization, allowing description of all authorized contexts of named entities [6].

[17] F-measure is calculated according to formula (3) and (3').

References

1. Cunningham, H., Bontcheva, K.: Named Entity Recognition. In: Proceedings of RANLP 2003, Borovets, Bulgaria (2003)
2. Friburger, N.: Automatic Recognition of Proper Names: An Application in Automatic Clustering of Journalistic Texts, University of Tours, PhD Thesis (2002)
3. Harris, Z.S.: Transformational Theory. Language 41(3), 363–401 (1965)
4. Mac Donald, M.: Internal and external evidence in the identification and semantic categorization of proper names. Corpus processing for Lexical Acquisition. Massachussetts Institute of Technology, pp. 21–39 (1996)
5. Maloney J., Niv, M.: TAGARAB: A fast, accurate Arabic name recogniser using high-precision morphological analysis. In: Proceedings of COLING-ACL workshop on Computational Approaches to Semitic Languages, Montréal, Canada (1998)
6. Maurel, D., Piton, O., Grass, T.: Description of a multilingual database of proper names. In: Proceedings of PorTal 2002, Faro, Portugal (2002)
7. Mesfar, S.: Standard Arabic formalization and linguistic platform for its analysis. In: Proceedings of Arabic NLP/MT conference, London, England (2006)
8. Mikheev, A., Moens, M., Grover, C.: Named Entity Recognition without Gazetteers. In: Proceedings of EACL 1999, Bergen, Norway (1999)
9. Poibeau, T.: Sur le staturéférentiel des entités nommées. In: Proceedings of TALN 2005, Dourdan, France (2005)
10. Poibeau, T.: The multilingual Named Entity Recognition framework. In: Proceedings of EACL 2003, Budapest, Hungary (2003)
11. Silberztein, M.: NooJ’s Dictionnaries. In: Proceedings of LTC 2005, Pozman, Poland (2005)
12. Silberztein, M.: NooJ Manual (2006) Download from http://www.nooj4nlp.net
13. Stevenson, M., Gaizauskas, R.: Using corpus derived name lists for Named Entity Recognition. In: Proceedings of ACL 2000, Hong Kong, China (2000)
14. Wakao, T., Gaizauskas, R., Cunningham, H.: Description of the LaSIE system as used for MUC-6. University of Sheffield (1995)

Appendix 1: Example of a Named Entity Recognition Transducer

Here are details of the syntactic grammar of recognition of person names. We distinguish 8 subtypes of person names (cf. fig. 2):

- Simple person names: <ENMAEX+PERS>
- Political persons names : <ENMAEX+PERS+POL> (cf. fig. 4)
- Military person names: <ENMAEX+PERS+MILIT>
- Person names with a profession: <ENMAEX+PERS+PRO>
- Person names with nationality: <ENMAEX+PERS>
- Person names with a title: <ENMAEX+PERS>
- Religious person names: <ENMAEX+PERS+RELIG>
- Sports person names: <ENMAEX+PERS+SPORT>

The following figure shows details of "Politicians" "Political Functions" and subgraphs. It includes two types of rules:

- Right trigger context rules: e.g. "الوزير الأوّل طوني بلير"[el wazîr el áwwal ṭony blîr - Prime minister Tony Blair]
- Left trigger context rules: e.g. " طوني بلير؛ الوزير الأوّل الإنجليزي "[ṭony blîr, el wazîr el áwwal el ínğlîziyy - Tony Blair, British prime minister]

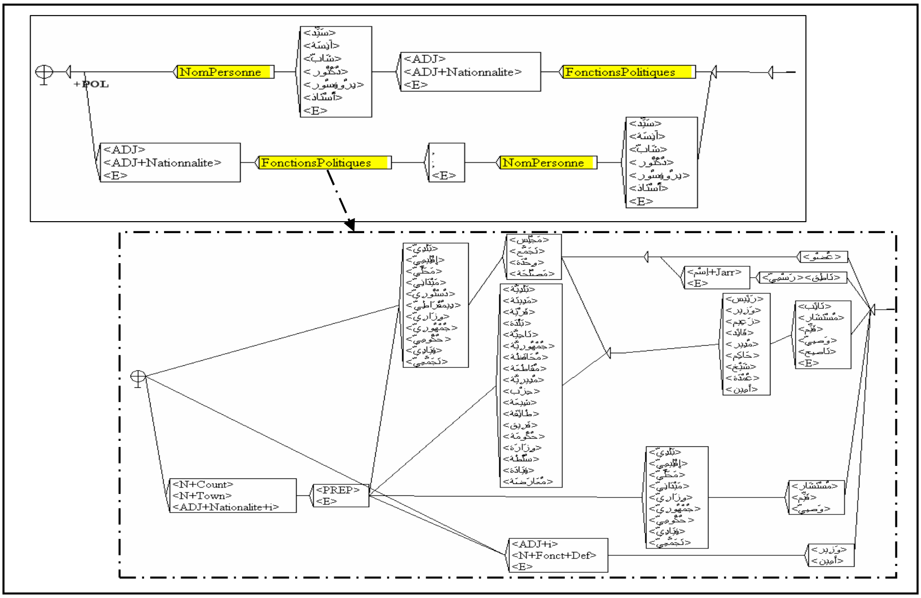


Fig. 4. Details of political person names sub-graph

Four Methods for Supervised Word Sense Disambiguation

Kinga Schumacher

German Research Center for Artificial Intelligence,
Knowledge Management Department
Kaiserslautern, Germany
kinga.schumacher@dfki.de

Abstract. Word sense disambiguation is the task to identify the intended meaning of an ambiguous word in a certain context, one of the central problems in natural language processing. This paper describes four novel supervised disambiguation methods which adapt some familiar algorithms. They built on the Vector Space Model using an automatically generated stop list and two different statistical methods of finding index terms. These proceedings allow a fully automated and language independent disambiguation. The first method is based upon Latent Semantic Analysis, an automatic indexing method employed for text retrieval. The second one disambiguates via co-occurrence vectors of the target word. Disambiguation relying on Naive Bayes uses the Naive Bayes Classifier and disambiguation relying on SenseClusters[1] uses an unsupervised word sense discrimination technique. These methods were implemented and evaluated to experience their performance, to compare the different approaches and to draw conclusions about the main characteristic of supervised disambiguation. The results show that the classification approach using Naive Bayes is the most efficient, scalable and successful method.

Keywords: Word Sense Disambiguation, Term weighting, Machine Learning.

1 Introduction

Ambiguity is one of the main issues for automatically processing natural language documents. The meanings of homonyms can only be determined by considering the context in which they occur. Approaches to this problem are based on the contextual hypothesis of Charles and Miller [1], in which words with similar meanings are often used in similar contexts and similar contexts of an ambiguous word also suggest the similar meaning.

In some cases of automatic text processing, it is adequate to examine the number of different senses of a word and to group the contexts of the ambiguous word based on their intended meaning, so called word sense discrimination [2]. Best suited techniques for this map contexts in vector space and cluster them in order to find similar groups, e.g. SenseClusters.

[1] http://www.d.umn.edu/~tpederse/senseclusters.html

Z. Kedad et al.(Eds.): NLDB 2007, LNCS 4592, pp. 317–328, 2007.

In other cases, it is required assigning contexts of a homonym from a predefined set of possible meanings, named word sense disambiguation [2, 3]. Knowledge-based disambiguation methods use prescribed knowledge sources like WordNet[2] to match the intended meaning of the target word. Corpus-based methods do not rely upon extensive knowledge bases; they use machine learning algorithms to learn from annotated training data to disambiguate new instances. The main approaches of corpus based disambiguation are to use the context vector representation, to interpret clusters with a semantic network and to make senses with decision lists [6].

The adoption of statistical analysis to represent contexts as vectors provides several advantages. Mapping text data in Vector Spaces enables language independent[3] fully automated processing and the usage of efficient statistical and probabilistic algorithms for disambiguation. Hence the methods introduced in this paper are based on the Vector Space Model. They are capable to learn, are language independent and fully automated.

This paper is structured as follows. Chapter 2 gives a state of the art overview. The generation of stop lists and the two different indexing strategies used by the methods are described in chapter 3, the disambiguation methods in chapter 4. The first disambiguation method applies Singular Value Decomposition and dimension reduction like LSA, is described in chapter 4.1. The second one, which creates co-occurrence vectors of the homonym for each meaning, is presented in chapter 4.2. The disambiguation method using the Naive Bayes Classifier is described in 4.3. The fourth one basing on SenseClusters is described in 4.4. Chapter 5 sums up the result of the evaluations and the paper is completed in chapter 6 with the conclusions.

2 Related Work

Schütze gives in [2] a good introduction to word sense discrimination and Purandare describes in [9] comprehensive the particular techniques which have been used by SenseClusters. The comparison of some word sense discrimination techniques is to find in [3]. The papers [4] and [5] explain two knowledge-based disambiguation methods which use WordNet. Levow gives in [6] an overview of the main corpus-based techniques, especially using context vectors, neuronal networks or decision lists.

A Vector Space Model-based disambiguation method is described and compared with previous works in [7]. Karov and Edelman developed a disambiguation method using a word similarity and a sentence similarity matrix [15]. In recent works on Word Sense Disambiguation the knowledge-based approach is applied [4, 17] which is, due to the multilingualism, less adequate than the news-domain[4].

[2] http://wordnet.princeton.edu/

[3] Language independent means that no adaption needed to apply the methods for corpora in a certain language.

[4] The methods have been developed in the context of the EU-project NEWS (www.news-project.com).

3 Indexing

There are several different ways to find index terms and construct the Vector Space Model of a given text data. The standard approach to weight the terms is using *tf/idf* [8] to weaken words which are present in nearly all documents and reinforce rare terms making the usage of a stop list unnecessary. The problem is to weight terms in single documents or context not included in the training data. The work presented here automatically generates the stop lists based on the property of stop words which is a high document frequency (*df*). Terms which occur in the most of the documents are not useful for finding different features, they are stop words. The benefit for statistical disambiguation approaches besides being language independent is to have a well adopted stop list for the current context set. After removing all stop words we have only statistical significant index term-candidates. The four methods use two different ways to appropriate the index terms. "Disambiguation with LSA" and "Disambiguation with Naive Bayes" select terms with a *tf* above a predefined threshold computed over all training data. "Disambiguation with SenseClusters" and "Disambiguation with Co-occurrence vectors" use terms as index terms, which are parts of characteristic co-occurrences. Characteristic co-occurrences (e.g. cat - miaow) can be found by computing the log-likelihood ratio of each pair of terms they occur near by each other [9]. Characteristic are only co-occurrences with a log-likelihood ratio beyond the 'Degree of Freedom' (3.841)[5].

4 Methods

4.1 Disambiguation with LSA

Latent Semantic Analysis (LSA) is an automatic indexing method deployed for text retrieval and established for several Information Retrieval challenges due to its beneficial properties.

The starting point of "Disambiguation with LSA" is the term-context matrix (TCM) of *tf*. Determining the Singular Value Decomposition (SVD) the latent semantic structure in the data is opened up [10]. SVD computes from TCM X the singular values S_0, transposed singular vectors of contexts D_0' and singular vectors of terms T_0, based on associations between terms, contexts, and between contexts and terms [11]:

$$X = T_0 S_0 D_0^{'} . \tag{1}$$

Let t to be the number of terms, d the number of contexts, respectively X has a dimensionality of $t \times d$, T_0 of $t \times m$, D_0' of $m \times d$ and S_0 of $m \times m$, where m is the rank of X. A reduction of the dimensionality from m to k is accomplished by deleting entries of low singular values and also the appropriated singular vectors [11]. Remaining singular values (S) context (D') and term vectors (T) are used to produce the so-called Latent Semantic Space [10]:

[5] This value comes from the chi-square distribution. Co-occurrences with a log-likelihood above this critical value are considered to be strongly associated [8].

$$\hat{X} = TSD' . \tag{2}$$

The SVD and dimension-reduction have several effects. Synonyms, different expressions for the same thing, are mapped; characteristic co-occurrences are detected; the major features of text data are extracted, less intense features and noise in the data are omitted [12]; contexts and terms are represented in the same space and homonyms are mapped to the centroid of their meanings.

Due to the last effect, processing SVD on the complete set of context would cause the aggregation of all meanings in one vector and a more enclosed representation of context vectors. Terms which build characteristic co-occurrences with the target word would then be mapped as terms with a related meaning. For this reason each meaning requires its dedicated Vector Space. This solution has the benefit that not only the target word has a more exact representation but also all other ambiguous words in its context have; this correlates with Charles and Miller's thesis [1].

To disambiguate a new context means to map it into the Latent Semantic Spaces and to compare it with its context vectors on the basis of their cosine or another similarity measure. In order to decrease the costs of disambiguation, it is necessary to reduce the set of vectors which are representatives of a space. Therefore we implemented two reduction ways. One procedure is based on the assumption that contexts are generally shorter than documents, hence they have fewer distinguishing features and a lot of context vectors are close to each other. A group of such vectors can be placed with respect to their centroid. We call the remaining context vectors the base vectors of the space. Another procedure is to find context vectors that discriminate a Latent Semantic Space from the others; those are the most discriminative ones. This can be done by first mapping the context vectors onto all other spaces and then compute similarities with their centroids. The most discriminative vectors are the ones with the smallest similarity.

Mapping a new context in a Vector Space is done by first creating the vector *q* of *tf* of index terms and then by placing it into the centroid of term vectors of the Latent Semantic Space weighted with the corresponding value in *q*:

$$\hat{q} = q'TS^{-1} . \tag{3}$$

The intended meaning of a target word in this new context can be estimated by choosing the Latent Semantic Space with the most similar representative vectors.

This method has more advantages than handling synonyms and extracting major features of data by LSA. The model is extensible since new terms and contexts can be integrated. Integrating a new term is done by placing it into the centroid of the contexts, which contain it. Context can be integrated in the same way. Such a meaning representation is cost-saving since the dimensionality is reduced to $k<m$. Model fitting is facilitated by k.

4.2 Disambiguation with Co-occurrence Vectors

This method relies on the idea that characteristic co-occurrences in a context assign the meaning of the target word. Consequently it is necessary to find characteristic co-occurrences in the context and build the co-occurrence vector of the target word.

Disambiguation can then be done by comparing the vector of the new context with the co-occurrence vector of each meaning.

Index terms are terms of co-occurrences; the initial matrix is a context-term matrix of *tf*s. Given of the advantages offered by SVD and dimensionality reduction, these were also applied here. Since that SVD maps homonyms to the centroid of their meanings, a dedicated Vector Space is created for all predefined meanings of the target word. In analogy to Disambiguation with LSA, SVD decomposes the initial matrix into three component matrices (*T, S, D'*) shown in (2). The co-occurrence vector of the target word can be found by computing the corresponding term-term matrix (TTM):

$$TTM = TS(TS)' \ . \tag{4}$$

The weight $w_{i,j}$ in *TTM* expresses the intensity of the correlation between term *i* and term *j*. The co-occurrence vector of the target word is the corresponding vector in the matrix. This vector shows how much an index term contributes to the identification of the target word's meaning. In order to make the vectors of different spaces comparable, the *TTM*s have to be scaled.

A new context can be disambiguated by creating its *tf*-weighted vector *c*. Since the weights of a co-occurrence vector *cv* represent the strength of the association to the target word, the similarity can be seen as the weighted average of them:

$$sim(c,cv) = \frac{\sum_{i=1}^{\dim(c)} c_i cv_i}{\sum_{i=1}^{\dim(c)} c_i} \ . \tag{5}$$

Dim(c), the dimension of the context vector is equal to the dimension of the co-occurrence vector i.e. the number of index terms. The division by the number of index term occurrences induces a shift of emphasis to the existence and the distribution of terms. This feature insures that similarities between different context vectors and a co-occurrence vector are comparable.

Like for Disambiguation with LSA (3.1), most of the benefits of dealing with synonyms come from SVD and dimension-reduction. Extracting the main features of the data helps discriminating the different meanings of the target word.

Disambiguating homonyms in a new context is compared to LSA much more cost-saving. The model can not be extended with new terms or contexts since the TTM does not include context vectors.

4.3 Disambiguation with Naive Bayes

Supervised disambiguation can be seen as a classification task where classes are the predefined potential meanings of homonyms. Annotated training contexts are the instances with attributes as their index terms. Many learning methods for supervised classification exist, the Naive Bayes Classifier has been chosen for its low complexity and good results by text classification. This method is based on the simple context-term matrix of *tf*. Naive Bayes requires attributes to be conditionally independent of each other, given the class [13]. The applied ‘bag of words’ approach [14] meets even more than this requirement, since natural language data is considered as a disordered set of words where all words have the same concern. Learning from training data is done by computing the a priori probabilities of appearance a potential attribute-value pair with reference to a class [13]:

$$p(H_j) = \frac{number\_of\_c_j}{number\_of\_c},$$

$$p(E_i \mid H_j) = \frac{number\_of\_c_{E_i,j}}{number\_of\_c_j} \text{ where} \tag{6}$$

c: context, c_j: context of class *j*, $c_{Ei,j}$: context with evidence *i* of class *j*
E_i: attribute-value combinations, H_j: classes.

The appliance of the Laplace Approximation with parameter μ (e.g. μ=1) assures the computability of a posterior probability by zero a priori values. It is done by adding *μ(number of classes)* on *(number of* c_j*)* in both equation. A target word in a new context can be disambiguated by being converted to a context vector and then be processed through the Bayes’ rule:

$$p(H_j \mid E_1 ... E_n) = \frac{(\prod_{i=1}^{n} p(E_i \mid H_j))p(H_j)}{\sum_{l=1}^{m} \left[(\prod_{i=1}^{n} p(E_i \mid H_l))p(H_l) \right]}. \tag{7}$$

The result of (7) is the a posterior probability that the target word is in the context of meaning *j*. To extend this model with new terms or contexts all a priori probabilities have to be computed again. However the learning and disambiguating steps in this method are not expensive.

4.4 Disambiguation with SenseClusters

SenseClusters[6] is a freely available word sense discrimination system using an unsupervised clustering approach. The core of SenseClusters is based on a powerful context representation relying on first or second order context vectors. Therefore, only one part of the context collection is used to gather index terms to create a term-term matrix (TTM) of log-likelihood values whereas the rest is used to create context vectors and cluster them. A first order context vector contains the *tf* of index terms in the context [9]. A second order context vector is the average of the vectors from the TTM which match terms in this particular context. Each vector of the TTM is weighted by the number of its occurrences in the context [9]. In this method, second

[6] http://senseclusters.sourceforge.net/

order vectors have been chosen relying on the evaluations done in [3] showing better results on small data collections. SenseClusters uses hierarchical methods to find clusters of contexts which represent different meanings of the target word. In case of supervised disambiguation, the training data is annotated and it is necessary to acquire some extra knowledge to disambiguate new contexts. In this new approach, called "Disambiguation with SenseClusters", the K-Means clustering algorithm[7], a well-known partitioning method, is used to deliver the clusters of different meanings but also additional information about their centres. Hence, whereas the mapping procedure to disambiguate a new context *q* is the same as for creating a second order context vector from a training data, the intended meaning of the target word in *q* can now simply be found by determining the most similar cluster centre.

This method is the most cost-expensive one and extending the model requires to retrain the whole system. Moreover, the amount of training data needed is higher than for other methods since part of the data is used to create the TTM and the rest is used to compute and cluster the context vectors.

5 Evaluation

5.1 Evaluation Data and Method

The disambiguation methods were tested with data from the Reuters Corpus[8] RCV1 containing 800.000 English news articles for the period 1996-1997. The two ambiguous words 'Washington' and 'Bush' have been chosen with predefined meanings 'Washington DC', 'George Washington', 'Washington State' and respectively 'Bush Junior' and 'Bush Senior'. The word 'Bush' defines the most difficult case since both meanings are often used in very similar contexts involving terms like 'US President', 'Washington', 'White House', 'USA' etc. The news articles were randomly chose from the set of articles which contains 'Bush' or 'Washington'. For both target words two corpora with different sizes[9] have been used. Table 1 contains the number of news articles and the number of contexts per corpus. The number of contexts is computed using a "context window" over 40 terms (20 terms before and 20 terms after the target word). The proportion of news articles relative to a meaning should map the one in the reality. The data has been manually annotated.

[7] K-Means chooses *k* random instances as initial cluster-centres, where *k* is the number of predefined meanings. All instances are ranked to the most similar centre, with respect to the measure cosine. After all instances have been processed, the new cluster centre is the centroid of its associated vectors. These two steps have to be carried out in alteration just as long as it takes to have the cluster centres remaining in the same position.

[8] http://about.reuters.com/researchandstandards/corpus/

[9] The number of articles per set is an estimation of the news agencies' demand (Project NEWS). The smaller sets represent the frequency of less common, the larger sets the frequency of common ambiguous words per day in a big news agency. These data sets are comparatively to the common evaluation-sets small but the experiments of Banko and Brill in [16] show that the performance of disambiguation methods increase with the size of data.

Table 1. The number of news articles and contexts in each evaluated corpora

	Corpus	Number of news	Number of contexts
Bush Jr./ Bush Sr.	Bush_large	87/56	147/97
	Bush_small	45/28	59/43
G.W./W. DC./W. State	Washington_large	46/80/60	50/101/74
	Washington_small	22/28/23	23/33/29

The overall performance of the disambiguation methods is checked by computing the single-success rates. The data was evaluated using 10-folds cross-validation method with stratification[10].

5.2 Results

All four methods have been implemented to be highly parametrisable. The abbreviations used below are defined as followed: WS: window size for context; WS/2 terms + target word + WS/2 terms; CS: window size for co-occurrences; defines the maximal interspace (CS-2 terms) between characteristic term pairs.

5.2.1 Disambiguation with LSA

Table 2 shows the single-success rates of the method with base vectors. The percentage of meanings which have been correctly mapped, i.e. when the prediction of the meaning in the new context is the same as the meaning of the most similar vector, is given in the column "most similar vector". The highest average of prediction computed over all vectors of one Vector Space is given in the column "average similarity". The prediction based on the distribution of meanings in the 2*(number of predefined meanings)+1 most similar vectors is given in the last column. The values given below have been obtained using optimal parameters.

The best success rates can be achieved when using small data sets and considering the average similarity. Moreover, there are some significant differences between target words with two and three possible meanings showing the limitations of this method. Following values for the dimensionality k (see 4.1) appear to be optimal for this method: k=40% for the Bush-Corpora or k=30% for the Washington-Corpora. The difficulty of the disambiguation of the word 'Bush' explains why k must be increased to maintain significant results. The base vectors are computed as the centroid of context vectors with a high similarity. However, the resulting number of base vectors is then extremely low, around 10-15% of all vectors.

Table 3 presents the results of disambiguation with the most discriminative vectors. The best results are obtained by using large corpora and the average similarity. Like in the case of base vectors, the number of possible meanings plays an important role. The dimensionality is reduced to k=40% for 'Bush' or k= 20% for 'Washington'. The highest success rates are achieved by defining 70% of context vectors as the most discriminative ones.

[10] 10-folds cross validation partitions the training data in 10 parts. In each of the 10 passes one part is used for testing and the other 9 parts for learning until all parts have been used as test set. The result is computed as the average of the results of particular passes.

Table 2. Single-success rates (%) of "Disambiguation with LSA", base vectors

Dis. with LSA - Base vectors - Single-success rates (%)			most similar vector		average similarity		(2*number of meanings+1) most similar vectors	
			Per meaning	total	Per meaning	total	Per meaning	total
Bush	small	B. Jr.	78.00	71.05	87.00	*75.37*	94.00	71.38
		B. Sr.	65.00		63.75		48.75	
	large	B. Jr.	73.22	58.83	80.36	71.85	84.29	62.43
		B. Sr.	44.44		63.33		40.56	
Washington	small	G. W.	47.50	46.11	65.00	*61.94*	62.50	52.50
		W.DC.	48.33		68.33		55.00	
		W. St.	42.50		52.50		40.00	
	large	G. W.	40.00	49.33	44.00	50.14	52.00	50.01
		W.DC.	53.00		60.00		52.30	
		W. St.	55.00		46.43		45.72	

Table 3. Single-success rates (%) of "Disambiguation with LSA", most discriminative vectors

Dis. with LSA - discriminative vectors - Single-success rates (%)			most similar vector		average similarity		(2*number of meanings+1) most similar vectors	
			Per meaning	total	Per meaning	total	Per meaning	total
Bush	small	B. Jr.	86.00	69.10	81.00	67.38	78.00	72.75
		B. Sr.	52.20		53.75		67.50	
	large	B. Jr.	76.69	63.49	94.29	*72.98*	86.79	68.37
		B. Sr.	50.29		51.67		49.94	
Washington	small	G. W.	27.50	46.39	82.50	61.39	25.00	50.56
		W.DC.	56.67		56.67		61.67	
		W. St.	55.00		45.00		65.00	
	large	G. W.	31.00	49.19	57.00	*63.36*	31.00	54.54
		W.DC.	73.00		89.50		84.05	
		W. St.	43.57		43.57		48.57	

If we compare both methods, base vector method appears to be best suited for small corpora and discriminative vector method for large ones. Tests showed that a 1% higher success rate can be achieved with a small corpora and 1-3% lower success rate with a large corpora, compared to disambiguation using all context vectors.

5.2.2 Disambiguation with Co-occurrence Vectors

Optimal parameters for this method are WS=20, CS=3. The original dimensionality of the Vector Spaces is reduced to 40%. The window size CS for contexts varies between 2 and 5 without any significant changes in the single-success rate. This method is very sensible to the changes made on the stop list or on the index terms.

The best result with 86.12% is obtained with two possible meanings for the target word and a small corpus. This method was only capable of detecting two of the three meanings of 'Washington'. That a better rate has been obtained with "Washington_ large" compared to "Washington_small" can be explained by the fact that the break-even-point for the set of training contexts per meaning has not been achieved with the

Table 4. Single-success rates (%) of “Disambiguation with Co-occurrence vectors”

Dis. with Co-occ. vectors Single-success rates (%)			Per meaning	Total
Bush	small	B. Jr.	86.00	*86.13*
		B. Sr.	86.25	
	large	B. Jr.	86.43	76.27
		B. Sr.	66.11	
Washington	small	G. W.	37.05	45.68
		W. DC.	0.00	
		W. St.	100.00	
	large	G. W.	62.00	*52.33*
		W. DC.	0.00	
		W. St.	95.00	

small corpus. Indeed, computing characteristic co-occurrences requires a minimal frequency of co-occurrences. This also explains why this method is quite sensitive to the stop lists and to the index terms.

5.2.3 Disambiguation with Naive Bayes

The single-success rates in table 5 are obtained with WS=50 and using a large stop list in comparison to the other methods. “Disambiguation with Naive Bayes” is scalable with respect to the number of possible meanings; tests show similar single-success rates when extending the Washington-corpora to four possible meanings.

Table 5. Single-success rates of “Disambiguation with Naive Bayes”

Dis. with Naive Bayes Single-success rates (%)		total
Bush	Bush_small	99.64
	Bush_large	96.80
W.	Washington_small	99.76
	Washington_large	92.27

5.2.4 Disambiguation with SenseClusters

Table 6 embraces the results of this method including the single-success rates by clustering. Since the error rate by clustering is already quite high, this explains the high error rate in disambiguating new contexts.

5.2.5 Machine vs. Manual Stop List

The methods were tested with a manual stop list[11] in order to compare the results with the results of the automatically generated stop list. The single-success rates are in average 7% higher by using the generated stop list than the rates obtained with the manual stop list. It appears that automatically generated stop lists, based on the document frequency, are well suited for statistical disambiguation approaches since these stop lists are adapted to the training set and only statistical significant terms can be index terms.

[11] http://www.cs.utexas.edu/users/mooney/ir-course/

Table 6. Single-success rates (%) of clustering and disambiguation by WS= 40, CS = 3

Dis. with SenseClusters Single-success rates (%)			Per meaning	total	clustering
Bush	small	B. Jr.	58.00	45.86	58.90
		B. Sr.	33.75		
	large	B. Jr.	65.71	49.25	58.48
		B. Sr.	32.78		
Washington	small	G. W.	35.00	34.44	36.95
		W.DC.	68.33		
		W. St.	0.00		
	large	G. W.	48.00	32.00	35.73
		W. DC.	48.00		
		W. St.	0.00		

6 Conclusions

In this paper we have presented a set of full automatically language independent supervised disambiguation methods based on the Vector Space Model. The methods adapt some familiar algorithms which have been deployed for different tasks, especially LSA, the SenseClusters approach and the Naive Bayes classifier. Since the method "Disambiguation with Naive Bayes" is the less cost-expensive, the most scalable and trusted method, it turns out that handling disambiguation as a classification task presents a lot of advantages. Compared with previous works are the results of this method good. The disambiguation method described in [15] achieves an average success rate of 92%.

The evaluations show furthermore that terms of significant characteristic co-occurrences are side by side or one term in between since the index terms of the corresponding methods were almost the same by co-occurrence window sizes of 3, 4 and 5 terms. The indexing with characteristic co-occurrences still remains difficult by small data sets like in this evaluation since related methods are not applicable for homonyms which have more than two possible meanings (see table 4 and 6). The analysis of context and term vectors showed that there are not enough non zero attributes to identify the meanings which could not be detected.

Acknowledgements. The four supervised disambiguation methods have been developed in the context of the EU-project NEWS (News Engine Web Services, http://www.news-project.com). Part of this work has been supported by the Rheinland-Pfalz cluster of excellence "Dependable adaptive systems and mathematical modeling" DASMOD, project ADIB (http://www.dasmod.de/twiki /bin/view/DASMOD/ADIB).

References

1. Miller, G.A., Charles, W.G.: Contextual Correlates of Semantic Similarity. Language and Cognitive Processes 6(1), 1–28 (1991)
2. Schütze, H.: Automatic Word Sense Discrimination. Computational Linguistics 24(1), 97–123 (1998)

3. Purandare, A., Pedersen, T.: Word Sense Discrimination by Clustering Contexts in Vector and Similarity Spaces. In: Proceedings of CoNLL-2004, pp. 41–48 (2004)
4. Banerjee, S., Pedersen, T.: An Adapted Lesk Algorithm for Word Sense Disambiguation using WordNet. In: Gelbukh, A. (ed.) CICLing 2002. LNCS, vol. 2276, Springer, Heidelberg, pp. 136–145 (2002)
5. Lesk, M.: Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone. In: 5th International Conference on Systems Documentation (1986)
6. Levow, G.A.: Corpus-based techniques for Word Sense Disambiguation. Technical Report AIM-1637, MIT AI Lab, 1, Cambridge (1997)
7. Bagga, A., Baldwin, B.: Entity-Based Cross-Document Coreferencing Using the Vector Space Model. In: 16th conference on Computational linguistics (1996)
8. Salton, G., Wong, A., Yang, C.S.: A Vector Space Model for Information Retrieval. Communications of the ACM 18(11), 613–620 (1975)
9. Purandare, A.: Unsupervised Word Sense Discrimination by Clustering Similar Contexts. University of Minnesota (August 2004)
10. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by Latent Semantic Analysis. Journal of the American Society for Information Science 41(6), 391–407 (1990)
11. Berry, M.W., Dumais, S.T., O'Brian, G.W.: Using Linear Algebra for Intelligent Information Retrieval. Computer Science Department, CS-94-270 (1994)
12. Kontostathis, A., Pottenger, W.M.: Detecting Patterns in the LSI Term-Term Matrix. Technical Report LU-CSE-02-010, Department of Computer Science and Engineering, Lehigh University (2002)
13. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)
14. Russel, S., Norvig, P.: Artificial Intelligence: A Modern Approach, 2nd edn. Prentice-Hall, Englewood Cliffs (2003)
15. Karov, Y., Edelman, S.: Similarity-based word sense disambiguation. Computational Linguistics, vol. 24(1) (March 1998)
16. Banko, M., Brill, E.: Scaling to very very large corpora for natural language disambiguation. In: Proceedings of the 39th Annual Meeting on Association for Computational Linguistics (2001)
17. Pedersen, T., Banerjee, S., Patwardhan, S.: Maximizing Semantic Relatedness to Perform Word Sense Disambiguation, University of Minnesota Supercomputing Institute Research Report UMSI 2005/25 (March 2005)

Enhancing Relation Extraction by Eliciting Selectional Constraint Features from Wikipedia

Gang Wang, Huajie Zhang, Haofen Wang, and Yong Yu

Department of Computer Science and Engineering,
Shanghai Jiaotong University, Shanghai, 200240, China
{gavinwang,zhjay,whfcarter,yyu}@apex.sjtu.edu.cn

Abstract. *Selectional Constraints* are usually checked for detecting semantic relations. Previous work usually defined the constraints manually based on handcrafted concept taxonomy, which is time-consuming and impractical for large scale relation extraction. Further, the determination of entity type (e.g. NER) based on the taxonomy cannot achieve sufficiently high accuracy. In this paper, we propose a novel approach to extracting relation instances using the features elicited from Wikipedia, a free online encyclopedia. The features are represented as selectional constraints and further employed to enhance the extraction of relations. We conduct case studies on the validation of the extracted instances for two common relations *hasArtist(album, artist)* and *hasDirector(film, director)*. Substantially high extraction precision (around 0.95) and validation accuracy (near 0.90) are obtained.

Keywords: selectional constraints, relation extraction, feature generation.

1 Introduction

Selectional Constraints, the semantic constraints imposed on the arguments which a predicate can take, play an important part in Natural Language Processing (NLP). These constraints are represented using a set of conceptual semantic patterns or vocabulary [7] and usually specified based upon a concept taxonomy in many practical settings [1, 6]. In Information Extraction (IE), especially Relation Extraction, selectional constraints have been vastly employed to facilitate the extraction process, where the selectional constraints for a relation predicate are generally defined based on a lexical taxonomy, a formal ontology, or a simple manually constructed Named Entity (NE) hierarchy [8-14].

However, in the application of the selectional constraints to relation extraction there have been several issues. Firstly, manual efforts are usually required to define the selectional constraints for a relation predicate. In [9, 12], arguments of relation predicates are restricted to specific nodes on a manually constructed Named Entity hierarchy. In Ontology Learning & Population [8, 10], selectional constraints are defined by the relation schema in ontology, which are usually specified by domain ontology experts. This is generally time-consuming and impractical for large scale relation extraction tasks. Secondly, the corresponding semantic type of an entity

Z. Kedad et al.(Eds.): NLDB 2007, LNCS 4592, pp. 329–340, 2007.

cannot easily be obtained with sufficiently high accuracy. For the selectional constraints defined based on a Named Entity hierarchy, the task can be thought of as Named Entity Recognition (NER), which unfortunately is one of the major sources of errors as reported in [9, 10]. In Ontology Population where semantic classes (types) for entities under consideration are already known [8, 15] (or partially known [10]), the task is simplified to some extent. Nevertheless, in most application settings, this kind of knowledge is not sufficient or even absent. Thirdly, fine-grained selectional constraints highly demanded in a certain application domain are far from sufficient. According to Sekine et al. [2] the number of NE types is quite limited (i.e. 7 types in MUC [24], 8 in IREX [25] and 5 in ACE [26] program). Sekine et al. [2] designed an extended Named Entity hierarchy containing 150 NE types. Even this hierarchy is much richer, a later application [9] based upon the hierarchy suggested that it is still not satisfying. For example, suppose we want to extract the relation *hasArtist (album, artist)*, the semantic type *album* and *artist* probably cannot be mapped onto appropriate classes in a traditional coarse-grained Named Entity hierarchy. The manually constructed taxonomy WordNet [3] does provide a fine-grained concept hierarchy whereas it is weak in the identification of proper nouns, most of which are instance-level entities. It also covers few neologisms, slang and domain-specific terms [19]. For instance, given the following sentence, '*Los Angeles was the 1980 debut album by X.*', where actually the '*Los Angeles*' is an album name and the '*X*' is a rock music band in the U.S., traditional NE recognizers probably cannot give an appropriate tag for '*X*' and may give an incorrect tag such as *Location* or *City* for '*Los Angeles*'.

In this paper, we propose a novel approach to automatically elicit the fine-grained selectional constraint features from Wikipedia (http://en.wikipedia.org), the largest online encyclopedia, and enhance relation extraction using the constraint features. The relations to be extracted are defined between pairs of entities described in Wikipedia. We represent features of a Wikipedia entity in vector spaces and perform feature selection and refinement within each type of relations. The acquired constraint features for each relation are finally used to restrict the entities in relation instances extracted using a symbolic pattern-based extraction method.

The semantic constraint features are learnt using a set of positive training examples and represented as vectors of terms which "softly" model the semantic type of entities. The term "softly" means that the semantic type is a bag of weighted words (BOW). The validation of an entity against a set of constraints is to measure the similarity between BOW of the entity and that of the constraints. The larger the similarity is, the fewer violations the entity makes with respect to the constraints. Feature terms are extracted from the descriptions of instance-level entities in Wikipedia. In this way, our approach eliminates the need for manually constructing a concept taxonomy based on which the selectional constraints are defined. Constraints with finer granularity are attained using a bag of domain specific terms and the validation of an entity against constraints becomes more effective.

We conducted two case studies on the validation of the extracted instances for two relations *hasArtist(album,artist)* and *hasDirector(film,director)*. Substantially high extraction precision (around 0.95) and validation accuracy (near 0.90) are obtained. We also conduct experiments to show the impact of different features and their combinations.

The contributions of this paper are two fold. First, we present a new method for eliciting conceptual features of encyclopedic entity entries. Second, we propose a way of representing the selectional behavior of relation predicates using the elicited conceptual features and demonstrate the effectiveness of the selectional constraint features in enhancing the extraction of relations.

The rest of the paper is organized as follows. Section 2 gives a brief description of related work. In Section 3, we describe relevant features of Wikipedia. Section 4 elaborates on the methods. In Section 5, we present the experimentation and evaluation. Finally, we conclude this paper and discuss future work in Section 6.

2 Related Work

- **Selectional Constraints and Relation Extraction**

Resnik [6] presented a formal probabilistic model to represent selectional constraints. Girju et al. [1] proposed a semi-supervised pattern-based approach to learning WordNet-based selectional constraints for detecting *part-whole* relations. They showed that better accuracy was achieved by employing selectional constraints based upon WordNet senses, especially for their generally applicable patterns. A training corpus containing both positive and negative examples is manually constructed for learning the constraints. Roth et al. [5] proposed an entity and relation recognition approach which globally took care of semantic constraints of relations and entities. Ruiz-Casado et al. [17] described an extraction pattern-based method for extracting hyperonymy/hyponymy and holonymy/meronymy relations from Wikipedia to enrich WordNet. In their work, entries in Wikipedia are mapped to WordNet senses. In contrast, we automatically learn Wikipedia-based selectional constraints based on only positive examples directly extracted from tabular infoboxes in Wikipedia. The selectional constraints are learnt based on the Wikipedia-related structured and semi-structured features instead of a semantic concept taxonomy.

- **Wikipedia-based Research Work**

Recent years have witnessed a tremendous focus on Wikipedia, a collaboratively constructed world knowledge resource. Strube et al. [21] described a method for computing word relatedness employing the category hierarchy of Wikipedia. Bunescu et al. [22] also made use of the hierarchy in their *taxonomy kernel* for Named Entity Disambiguation. Gabrilovich et al. [20] whereas proposed to treat Wikipedia as having essentially no hierarchy and they presented a novel approach to generating rich Wikipedia-based features for enhancing short text categorization. They reported great improvements over the state of the art. Their later work [19] on computing semantic relatedness of natural language texts, *ESA* represented the meaning of texts in vector spaces of Wikipedia concept entries and achieved substantial improvements. From the work of Voss [18] and Gabrilovich [19, 20], the category hierarchy is not a formal *is-a* hierarchy and multiple categorization schemes co-exist simultaneously. Consequently, the categorization hierarchy is not adapted into our approach. We also found that many entries of Wikipedia actually cannot be treated as a semantic concept nevertheless they are in fact instance-level entities such as a person, a film or a pop band, etc. Consequently, in our approach we extract the machine-readable conceptual features for Wikipedia entries, which is very different from Gabrilovich [19, 20].

3 Wikipedia

Wikipedia, as a free online encyclopedia, now is the largest knowledge repository on the Web. There are currently versions in about 200 languages and the English version of Wikipedia is the largest and now contains more than 1.5 million articles. Wikipedia is collaboratively developed by volunteers using MediaWiki software. Wikipedia outstrips all other encyclopedias in coverage and is ten times as large as its closest rival, the *Encyclopedia Britannica* (http://store.britannica.com/). Its accuracy is found to rival that of *Encyclopedia Britannica* [16].

Wikipedia is a hypertext document collection with a rich link structure. In the main articles of Wikipedia, there are three kinds of pages: the disambiguation pages, ‘List of XXX’ and ‘Lists of XXX’ like pages and normal pages about entities. Typically the title of each page is the most common name for the entity described in that article. When the name is ambiguous, it is further qualified with a parenthetical expression. For instance, the name ‘*Black Sabbath*’ corresponds to three articles titled ‘*Black Sabbath (album)*’, ‘*Black Sabbath (film)*’ and ‘*Black Sabbath (song)*’.

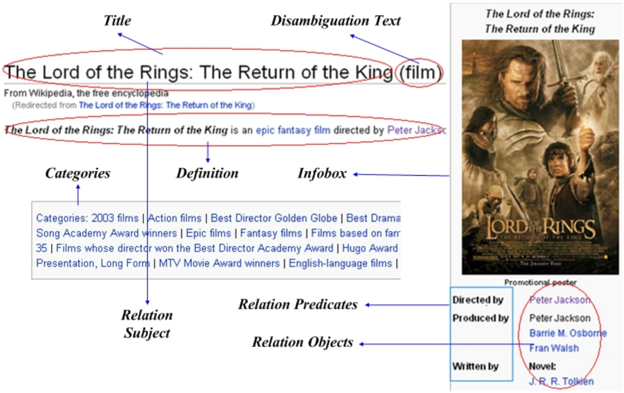

Title
Disambiguation Text
The Lord of the Rings: The Return of the King (film)
From Wikipedia, the free encyclopedia
(Redirected from The Lord of the Rings: The Return of the King)
The Lord of the Rings: The Return of the King is an epic fantasy film directed by Peter Jackso
Categories
Definition
Infobox
Categories: 2003 films | Action films | Best Director Golden Globe | Best Drama Song Academy Award winners | Epic films | Fantasy films | Films based on fan 35 | Films whose director won the Best Director Academy Award | Hugo Award Presentation, Long Form | MTV Movie Award winners | English-language films |
Relation Subject
Relation Predicates
Relation Objects
The Lord of the Rings: The Return of the King
Promotional poster
Directed by
Produced by
Written by
Peter Jackson
Peter Jackson
Barrie M. Osborne
Fran Walsh
Novel:
J. R. R. Tolkien

Fig. 1. Feature Snapshot of an entry in Wikipedia

In a normal page of Wikipedia the first sentence generally serves as the definition of an entity. Each article in Wikipedia is assigned at least one category. Articles within a category usually share the same topic. The categories form a hierarchy in which multiple categorization schemes co-exist simultaneously. In some articles, an infobox along with a picture gives a general description of an entity. In each infobox within an article there are a set of properties defined to describe the entity. Each property generally demonstrates a relation between two entities. The entity in current article can be viewed as the subject of the relations. The objects are connected by relation predicates and can be either internal links which point to other entities in Wikipedia or just literal text or external links pointing to web pages out of Wikipedia. Fig. 1 shows a feature snapshot of an article.

4 The Approach

We employ the edit-distance calculation approach described in [17] to automatically learn symbolic patterns to extract relation instances, using a set of relation seed instances randomly selected from infoboxes of Wikipedia. For example, two seeds for relation *hasDirector(film,director)*, <Titanic, James Cameron> and <King Kong (2005), Peter Jackson> are used to query the document collection of Wikipedia. Symbolic patterns are generated based on the returned text snippets. Given two snippets, ‘*Titanic was a romantic film directed by James Cameron*’ and ‘*King Kong (2005) is an American movie directed by Peter Jackson*’, it can automatically generate the following regular expression ‘** (is|was) (a|an) * (film|movie) directed by **’. The underscore for each phrase indicates that the phrase represents a Wikipedia entry associated with a corresponding hyperlink (or URL). For simplicity, we currently use hyperlinks in Wikipedia documents for entity chunking. The instances extracted by the patterns for a given relation are required to be verified by the selectional constraint features (SCF), the acquisition of which are described in the following parts of this section.

4.1 Selectional Constraint Feature Selection

As illustrated in Section 3, the definition sentence and the categories for a Wikipedia entity give much information about its semantic type. Moreover, the relation predicates which take an entity as arguments reflect the semantic type of the entity. As a result, for the semantic type of an entity, we also take into consideration the relation predicates which take the entity as argument at either the *Subject* or the *Object* position. We summarized the selectional constraint features in Table 1.

Table 1. Set of features for an entity (article *The Lord of the Rings*) in Wikipedia

Feature Name	Meaning	Example
DefinitionWords(DW)	Words in the definition sentence. (Words constructing the entity itself and stop words are excluded.)	epic, fantasy, film.
CategoryWords (CW)	The words constructing the names of categories.	film, drama, award
SubjectPredicates(SP)	Relation predicates taking the entity at *Subject* position.	director,producer, starring,writer,imdb_id
ObjectPredicates(OP)	Relation predicates taking the entity at *Object* position.	N/A.

4.1.1 Feature Subset Selection

For each relation, the instances are collected from Wikipedia infoboxes. For the subject (object) entities of the instances, feature subset selection is performed. In DW and CW, words are stemmed. For the hyperlinks in DW, each multi-word anchor text is transformed to a single term by connecting each word (token) with underscores.

Given only positive training examples, we design an unsupervised feature selection approach using the following term weighting scheme, assuming that a commonly occurring term within a set of conceptually related documents is more important.

$$weight(t) = n_r \times \log(N/n_t) \ . \tag{1}$$

where n_t denotes the number of documents in which a term occurs with respect to the entire document collection, N is the number of the documents with respect to the entire document collection, n_r represents the number of documents in which a term occurs with respect to a subset of document collection constructed by the documents for the subjects (objects) of relation instances.

For each feature set, we collect top 10 terms with highest weights to construct a feature vector. The term weights are re-calculated using traditional TFIDF scheme.

Finally for each relation, each Wikipedia entity extracted or used for training at *Subject* (*Object*) position is represented as four feature vectors, $\mathbf{v}_{DW}, \mathbf{v}_{CW}, \mathbf{v}_{SP}, \mathbf{v}_{OP}$.

4.2 Selectional Constraint Feature Generation

From the observations, we found that subjects (objects) of instances for a relation are not necessarily instances of a single unified concept. This is intuitive and as an example, the relation *hasArtist(album, artist)* takes objects such as singer, music band and musician, etc. In order to discover more coherent and consistent selectional constraint features, clustering is employed to partition the collection of features into several clusters where each represents a virtual concept.

As described in Section 3 and shown in Fig. 1, the disambiguation text in an article title can usually be considered a concept for that disambiguated entity. For example, the objects of *hasArtist(album, artist)* have the following disambiguation text, 'band', 'singer', 'musician', 'guitarist', 'rapper', 'entertainer', etc. Herein we refer to them as labels. Excluding the labels with very low counts, we obtain a list of popular labels, which provide us certain degree of knowledge about the entities. Up to now, we can impose constraints on the process of clustering by defining a *must-link* constraint (two instances have to be together in the same cluster) and the clustering becomes a semi-supervised one [23]. The *must-link* constraint is defined to be that two entities with the same label must be put into the same cluster.

However, the labels cannot be guaranteed to be complete and thus the number of clusters cannot be pre-determined. Consequently, we perform agglomerative hierarchical clustering on the dataset. The similarity between any two entities is defined as similarity between the corresponding selectional constraint features of the two, which is shown in Equation (2).

$$sim(e_1, e_2) = sim(\mathbf{V}^1, \mathbf{V}^2) = \begin{cases} 1 & \text{if } e_1 \text{ and } e_2 \text{ have the same label,} \\ \dfrac{1}{\sum_i \mathbf{w}_i} * \sum_i \cos(\mathbf{v}_i^1, \mathbf{v}_i^2) * \mathbf{w}_i & \text{otherwise.} \end{cases} \tag{2}$$

where $\mathrm{i} \in$ {common features of e_1 and e_2 in {DW,CW,SP,OP}}.

In Equation (2), we use the *cosine similarity* to measure the similarity between two entities. Note that among the four feature sets, the subject predicates and the object predicates are considered semantic features since they are extracted from the relation instances instead of the unstructured article contents. The two are not always present for an entity in which case we do not count them in and normalize the similarity score via the total weights used.

In the following, we illustrate the clustering process. Firstly, we define the Average Pairwise Similarity (*APS*) of a cluster C in Equation (3).

$$APS(C) = \begin{cases} \dfrac{1}{|C| \times (|C|-1)/2} \times \sum\limits_{\substack{\mathbf{V}_i, \mathbf{V}_j \in C, \\ \mathbf{V}_i \neq \mathbf{V}_j, i<j}} sim(\mathbf{V}_i, \mathbf{V}_j) & \text{if } |C| > 1, \\ 0 & \text{if } |C| = 1. \end{cases} \tag{3}$$

The Cohesion of a cluster C is defined as,

$$Cohesion(C) = |C| \times APS(C) \ . \tag{4}$$

Our clustering process is to optimize the overall cohesion of clusters based upon the following Internal Cohesion IC function (*k* is the number of clusters),

$$\text{maximize } IC = \sum_{i=1}^{k} Cohesion(C_i) \ . \tag{5}$$

Greedy strategy is employed to approximately optimize the IC, two clusters C_i and C_j are merged if the resulting Internal Cohesion Gain (ICG) is maximal, which is to maximize

$$ICG(C_i, C_j) = Cohesion(C_i \cup C_j) - (Cohesion(C_i) + Cohesion(C_j)) \ . \tag{6}$$

The clustering process stops if the maximal ICG is still below a threshold τ_c. The final output of the clustering is a set of clusters $\mathrm{X} = \{C_1, C_2, ..., C_k\}$.

4.3 Relation Validation with Selectional Constraint Features

The extraction process is now straightforward. Firstly, the learnt patterns are matched against the entire free text collection. The subject and object extracted from each matched sentence are verified using corresponding selectional constraint features. In this stage, the subject and the object each should be associated with a unique entry in Wikipedia. The corresponding entry identifier can be retrieved in indexes. Up to now, each subject or object corresponds to an entity which is associated with a set of selectional constraint features represented by vector $\mathbf{V}_q$. We create a cluster $C_q = \{\mathbf{V_q}\}$ containing only $\mathbf{V}_q$ and find a cluster C_e with which maximal Internal Cohesion Gain is obtained. It is defined as:

$$C_e = \underset{C_i \in \mathrm{X}}{\arg\max} \ ICG(C_q, C_i) \ . \tag{7}$$

The entity is rejected if the maximal ICG falls below a threshold τ_q and is accepted, otherwise. Formally put, an *accept* function is defined as follows,

$$accept(q) = \begin{cases} true & \text{if } ICG(C_q, C_e) \geq \tau_q, \\ false & \text{otherwise.} \end{cases} \ . \tag{8}$$

An extracted relation instance is accepted if both the subject and the object are accepted by the function.

5 Experimentation

For the experiments in this paper, we used data from the *Wikipedia XML* corpus [27], which are a set of XML collections based on Wikipedia in early 2006. We used only the main English collection of XML files. We parsed the XML collection, out of 644,577 normal pages, 15 percent are associated with at least one infobox. There are 953,550 relation seeds (#Subject: 89,406, #Objects: 109,868, #Predicates: 9,197). Each article is spitted into sentences using OpenNLP (http://opennlp.sourceforge.net/) and then indexed using Lucene (http://lucene.apache.org/). The hyperlinks in each article are kept in the indexes.

We conducted two case studies on extracting relations *hasArtist(album,artist)* and *hasDirector(film,director)*. The evaluation focuses on the impact of the selectional constraint features on the validation of extracted instances. We also demonstrate the impact of various features, their combinations, and the feature clustering.

5.1 Effect of Selectional Constraint Features on Extraction

Patterns shown in Table 2 are used to extract relation instances, from which, around 100 instances are randomly selected for evaluation (We only keep the instances which do not appear in the initial training seeds.) The subject and the object of an instance are required to be an entry in Wikipedia. Three human subjects are asked to judge each relation instance and the result which has more votes is chosen as the standard answer. The subject and the object are judged separately. For relation *hasDirector(film,director)*, the subject should be a film and the object should be a director. If any of the subject and the object do not conform to the constraints, the relation instance is marked false. Otherwise, the relation instance still needs further check to prove its validity. Table 3 gives examples extracted by pattern Pd2.

Table 2. Patterns. (<S> and <O> represent placeholders for subject and object.)

Relation	Pattern Name	Pattern Expression
hasDirector(film,director)	Pd1	<O> ' s (\S+)? (film|movie)? <S>
	Pd2	<S> (is|was)? directed by <O>
hasArtist(album,artist)	Pa1	<O> ' s (album)? ,? <S>
	Pa2	<O> (released|releases) (a|an|the)? (\S+)? (album)? <S>

Table 3. Relation Instances Extracted by Pattern Pd2 for relation *hasDirector(film,director)* and judgements made by human and SCF

Subject	Object	Judgement by Human	Judgement by SCF
Golden Age (film)	Shekhar Kapur	true	true
Stargate	Roland Emmerich	true	false
Batman Forever	Joel Schumacher	true	true
A.I. (film)	Steven Spielberg	true	true
University of Arizona	William Rathje	false	false
Animation	Nelson Shin	false	true
Death of a Salesman	Elia Kazan	true	true
The Chipmunk Adventure	*Janice Karman*	true	false
...	......	......	......

Table 4 gives the results, in which the precision in first column means the precision of a pattern without applying selectional constraint features. The *P, R, A* represent precision, recall and accuracy of selectional constraint features respectively and are defined in Equation (9). (SCF represents Selectional Constraint Features.)

$$P = \frac{|\,\text{items marked true by both human and SCF}\,|}{|\,\text{items marked true by SCF}\,|}$$
$$R = \frac{|\,\text{items marked true by both human and SCF}\,|}{|\,\text{items marked true by human}\,|} \qquad (9)$$
$$A = \frac{|\,\text{items with same mark by human and SCF}\,|}{|\,\text{all items}\,|}$$

Table 4. Effect of Selectional Constraint Features on Relation Extraction. (CL denotes Clustering while NCL denotes Non-Clustering. Average weighting on 4 features.)

Pattern (Accuracy)		**Subject**			**Object**			**Relation**		
		P	**R**	**A**	**P**	**R**	**A**	**P**	**R**	**A**
Pd1(**.174**)	CL	.964	.915	.929	.988	.907	.955	**.977**	.894	**.879**
	NCL	.952	.627	.778	.979	.630	.798	**.944**	.596	**.667**
Pd2(**.495**)	CL	.960	.989	.977	.986	.955	.959	**.865**	.938	**.876**
	NCL	.957	.937	.943	.985	.892	.902	**.913**	.875	**.835**
Pa1 (**.260**)	CL	.982	.964	.970	.976	.973	.953	**.981**	.964	**.950**
	NCL	.981	.928	.950	.958	.693	.744	**.967**	.732	**.720**
Pa2(**.444**)	CL	.988	.970	.965	.979	.998	.988	**.990**	.949	**.970**
	NCL	.985	.943	.930	.960	.905	.901	**.980**	.931	**.949**

In the results matched by pattern Pd2, at *Subject* position, sentence '<*Elevated (movie)*> is a <*1997*> <*Short subject*> directed by <*Vincenzo Natali*>' is matched and <*Short subject*> is extracted as subject, which is marked false by human judge. However it is incorrectly marked true by SCF. Looking into the features of the entity <*Short subject*>, it has such definition '*Short subject is an American film industry term that historically has referred to any film in …*' and categories '*short films*' and '*Portal:Art/Categories*' are not associated with any predicates, since SCF gives high weights for the term '*film*', <*Short subject*> is incorrectly accepted. The entity <*Animation*> is also incorrectly accepted as the subject as it has a category named '*film*' and is associated with the object predicates '*industry*' and '*movie_name*'. The two entities are general concepts in film domain and thus share many features with other film instances. For the coverage, at the *Object* position, for sentence '<*The Chipmunk Adventure*> was directed by <*Janice Karman*>' matched by pattern Pd2, SCF incorrectly rejected the entity <*Janice Karman*>, which has definition '*Janice Karman is an American film_producer, record_producer, singer, and voice artist.*' In which the term '*film_producer*' has very low weight in DW features. Furthermore, the entity shares no relevant feature terms with SCF in the CW, SP, OP feature set.

In the sentence '<*Stargate*> directed by <*Roland Emmerich*>' matched by pattern Pd2, although there is actually a film named '*Stargate*' which is directed by the director <*Roland Emmerich*>, the hyperlink for <*Stargate*> actually points to a general concept which has such definition '*Stargate collectively refers to the fictional universe started with the 1994 science fiction feature film Stargate, …*'. It is rejected by SCF while it is accepted by the human subjects. There is actually an entry '*Stargate (film)*' which is talking about the Stargate film, therefore we considered that the hyperlink is incorrectly marked in Wikipedia.

5.2 Effect of Selectional Constraint Feature Clustering

Table 4 shows that clustering can improve the coverage with insignificant changes of precision. Since the subjects/objects of instances for a relation are not necessarily instances of a single unified concept, without clustering, the cohesion of SCF is low and it tends to reject the entities with low weight rare features. As an example, the object <*Carl Orff*>, which is a music composer and matched by pattern Pa1, is rejected by NCL version of SCF. However, there is a small amount of such kind of entities in the initial training instances at the *Object* position, therefore it should be accepted and is actually accepted by SCF using clustering. In NCL SCF, the average pair-wise similarity is low within a single cluster and the threshold used in *accept* function defined in Equation (8) cannot be well-determined and it can just be either too loose or too strict. In the experiment settings, an empirical threshold is chosen in order to maximize the accuracy of the SCF. Without clustering, SCF for relation *hasArtist(album, artist)* at the *Object* position rejected some correct entities such as <*Richard Wagner*> (a composer, conductor), <*Damon Albarn*> (a vocalist and keyboardist), etc. Meanwhile, NCL version of SCF at the *Subject* position of *hasArtist(album, artist)* did not give a significant decrease in coverage due to the fact that the subjects of the relation coherently represent one unified concept, album. However, clustering achieved significant improvement on coverage when the relation subjects/objects are diverse, as shown in Table 5.

5.3 Effect of Different Selectional Constraint Features and Combinations

In Table 5, DW, CW and their combination achieved higher coverage and overall accuracy while SP, OP and the combination resulted in higher precision. Firstly, SP and OP provide more semantic information about an entity thus bear stronger features with respect to the semantic type of an entity and this leads to higher precision. Secondly, SP and OP are not always present for an entity, in which case, the coverage falls down. In order to achieve maximal overall accuracy, we give higher weights to DW and CW, we also prefer SP to OP. The combination in the last row led to the highest overall accuracy.

Table 5. Effect of Different Selectional Constraint Features and Combinations (Data calculated from instances matched by pattern Pd1 and Pd2 for relation *hasDirector(film,director)*, orignal accuracy/precision=.345)

Features	**Subject**			**Object**			**Relation**		
	P	**R**	**A**	**P**	**R**	**A**	**P**	**R**	**A**
DW	.851	.906	.856	.955	.815	.832	**.835**	.747	**.714**
CW	.877	.934	.889	.869	.969	.867	**.786**	.926	**.745**
SP	.995	.486	.727	.964	.615	.699	**.973**	.379	**.520**
OP	.912	.682	.782	.972	.815	.845	**.908**	.621	**.663**
0.5DW+0.5CW	.918	.944	.920	.900	.969	.896	**.807**	.926	**.806**
0.5SP+0.5OP	.987	.692	.819	.981	.815	.856	**.984**	.653	**.709**
0.3DW+0.3CW +0.25SP+0.15OP	.962	.953	.952	.990	.931	.953	**.916**	.916	**.878**

6 Conclusions and Future Work

In this paper we presented a novel approach to elicit Wikipedia-based selectional constraint features and employ the features in the extraction of relations from the natural free text of Wikipedia. In the experiments, we investigated the impact of various selectional constraint features and their combinations on relation extraction. It showed that the extracted selectional constraint features can substantially improve the precision of traditional relation extraction. We also demonstrate the effect of feature clustering. Our method can also result in improvements in the coverage of relation extraction indirectly because many general extraction patterns such as ‘A’s B’, ‘B of A’ etc. can be still applied in the presence of the selectional constraint features.

Future research directions are: a) to design more formal models to represent the selectional constraints; b) to bootstrap using the newly learnt relation instances.

Acknowledgments. The work is funded by IBM China Research Lab.

References

1. Girju, R., Badulescu, A., Moldovan, D.: Learning semantic constraints for the automatic discovery of part-whole relations. In: Proceedings of HLT-NAACL (2003)
2. Sekine, S., Sudo, K., Nobata, C.: Extended Named Entity Hierarchy. In: Proceedings of the LREC-2002 Conference (2002)
3. Fellbaum, C.: WordNet: An Electronic Lexical Database. MIT Press, Cambridge (1998)
4. Stevenson, M., Greenwood, M.A.: A Semantic Approach to IE Pattern Induction. In: Proceedings of the 43rd Annual Meeting of the ACL, pp. 379–386 (2005)
5. Roth, D., Yih, W.: Probabilistic Reasoning for Entity & Relation Recognition. In: Proceedings of 19th International Conference on Computational Linguistics (2002)
6. Resnik, P.: Selectional constraints: an information-theoretic model and its computational realization. Cognition (1996)
7. Karambelkar, S.: Acquisition of selectional constraints in natural language processing. Master thesis. University of Sheffield (2001)

8. Schutz, A., Buitelaar, P.: RelExt: A Tool for Relation Extraction from Text in Ontology Extension. In: Proceedings of the 4th International Semantic Web Conference (2005)
9. Sekine, S.: On-Demand Information Extraction. In: Proceedings of COLING (2006)
10. Boer, V., Someren, M., Wielinga, B.J.: Extracting Instances of Relations from Web Documents using Redundancy. In: Sure, Y., Domingue, J. (eds.) ESWC 2006. LNCS, vol. 4011, Springer, Heidelberg (2006)
11. Stevenson, M.: An Unsupervised WordNet-based Algorithm for Relation Extraction. In: 4th LREC Workshop Beyond Named Entity: Semantic Labeling for NLP Tasks (2004)
12. Choi, Y., Cardie, C., Riloff, E., Patwardhan, S.: Identifying sources of opinions with CRFs and extraction patterns. In: Proceedings of HLT/EMNLP, pp. 355–362 (2005)
13. Agichtein, E., Gravano, L.: Snowball: Extracting Relations from Large Plain-text Collections. In: Proceedings of Digital Libraries (2000)
14. Culotta, A., Sorensen, J.: Dependency tree kernels for relation extraction. In: Proceedings of the 42nd Annual Meeting of the ACL (2004)
15. Geleijnse, G., Korst, J.: Automatic Ontology Population by Googling. In: Proceedings of the 17th BNAIC, pp. 120–126 (2005)
16. Giles, J.: Internet Encyclopaedias Go Head to Head. Nature 438, 900–901 (2005)
17. Ruiz-Casado, M., Alfonseca, E., Castells, P.: Automatic extraction of semantic relationships for WordNet by means of pattern learning from Wikipedia. In: Montoyo, A., Muńoz, R., Métais, E. (eds.) NLDB 2005. LNCS, vol. 3513, Springer, Heidelberg (2005)
18. Voss, J.: Collaborative Thesaurus Tagging the Wikipedia Way. Available at http://arxiv.org/abs/cs/0604036
19. Evgeniy, G., Shaul, M.: Computing Semantic Relatedness using Wikipedia-Based Explicit Semantic Analysis. In: Proceedings of IJCAI'07 (2007)
20. Evgeniy, G., Shaul M.: Overcoming the Brittleness Bottleneck using Wikipedia: Enhancing Text Categorization with Encyclopedic Knowledge. In: Proceedings of AAAI'06, pp. 1301–1306 (2006)
21. Strube, M., Ponzetto, S.: WikiRelate! Computing Semantic Relatedness Using Wikipedia. In: Proceedings of AAAI'06 (2006)
22. Bunescu, R., Pasca, M.: Using Encyclopedic Knowledge for Named Entity Disambiguation. In: Proceedings of EACL'06 (2006)
23. Basu, S., Banerjee, A., Mooney, R.: Semi-Supervised Clustering by Seeding. In: Proceedings of ICML'02 (2002)
24. MUC: Voorhees, E.: Introduction to Information Extraction and Message Understanding Conferences, http://www.itl.nist.gov/iaui/894.02/related_projects/muc/
25. IREX: http://www.cs.nyu.edu/cs/project/proteus/irex
26. ACE: http://www.nist.gov/speech/tests/ace/
27. Denoyer, L.: The Wikipedia XML Corpus. SIGIR Forum (2006)

A Computer Science Electronic Dictionary for NOOJ

Farida Aoughlis

Faculté de Génie électrique et informatique
Université Mouloud Mammeri,
Tizi Ouzou, Algérie
fariyamo@yahoo.fr

Abstract. An automatic text analysis system cannot lexically recognize a word unless it already exists in the electronic dictionary. Our works applies to the NOOJ system. Work remains to be made to build terminological dictionaries. To build the terms dictionary, it is necessary to identify the terms in the texts, to count and to acquire them, to locate the terms by deciding if a word group constitutes or not an entry in the dictionary, and for all that the problem is to find corpora.

Keywords: Terminology, terminology extraction, terminology acquisition, electronic dictionary, compound noun, local grammar, NOOJ.

1 Introduction

Automatic translating software needs increasingly significant and varied terminological resources. They can be simple lists of terms that are more or less structured (structured indices, thesaurus, lexical networks) used by automatic indexing systems or for information retrieval or more documented terminological reference frames. [54]

The computer science compound words dictionary INFO_COMP that we build will allow the analysis and the automatic computer science texts indexing. The context of works is NOOJ was developed by Silberztein M. NOOJ is a new linguistic development environment, issued of author 12 years experience, as the INTEX user and designer at the LADL [48].

For the construction of our dictionary, we extract manually terminology from texts. We also extract terms automatically with NOOJ but here the terms extracted are examined if they are possible entries in the dictionary.

2 Terminology, Term and Specialized Language

The computer science texts are technical texts and the language is specialized. What do we mean by terminology? ISO, 1990 defines the terminology as the scientific study of the notions and the terms used in the specialist's languages. A specialized language [24], [25] is the use of a language that makes it possible to give an account

Z. Kedad et al.(Eds.): NLDB 2007, LNCS 4592, pp. 341–351, 2007.

technically of specialized knowledge. This specialized knowledge is linguistically called by terms that are words, word groups. In a general language, the lexemes constituting the inputs of the dictionary are words; in the case of a specialized language, the lexemes are terms. In the linguistics dictionary [22], any discipline, and with stronger reason any science needs a set of terms, rigorously defined, by which it indicates the concepts which are useful for him: this set of terms constitutes its terminology. In what follows a term relates to a compound noun. One will find in [36] work an examination of the techniques of extraction of the terminological data and their impact on the terminologist work.

3 Tools for Automatic Extraction of Terminology

A census of the various existing systems is made in [16].They make it possible to extract new terms starting from texts or corpus. Here also the various linguistic, statistical or mixed methods exist and are used to develop the tools for automatic extraction. Some extraction software evaluation criteria are examined in [35]. In [19], standard techniques for terms extraction are examined. In the thesis of [18] and in [21], specialized lexical pivots are used for the automatic acquisition of terms. In [20] a non-technical corpus is used for the extraction of terminology. An evaluation of acquisition tools for information extraction is made in [43]. In [31] we find studies about terminological variation. The majority of the authors [17] consider it essential to maintain a human activity in the acquisition systems, with an acceptance or a refusal of the results but also as far as being the acquisition centre, the computer processing will thus be reduced to a presentation and data-recording tool. Studies based on a semantic approach and terminology acquisition can be found in [40] and [16]. In the framework of the corpora specialized in terminology and the automatic acquisition of collocations, various works in progress [33], [41] and [45] are presented. A study about collocation can be found in [44]. In [38], Meilland J.C presents an automatic extraction terminology from short textual words. Smadja et al., [53] used statistical methods for translating collocations for bilingual lexicons.

3.1 Tools Using the Linguistic Methods

They are based on a syntactic analysis of the texts. A morpho-syntactic tagging locates all he noun syntagms (candidate terms). The expert retains the terms. Four systems are listed in [54]; we have TERMINO [15], LEXTER [8], [9], XTERM developed in 1999 by Cerbah, F. and FASTER [31]. Another system SINTESI (Systema INtegrato per TESti in Italiano), is described in [24], the extraction of terminology is used to generate search keys for a database. To acquire English computer science terms the system LEXPRO [46] uses the system INTEX [48]. NOOJ [50] allows terms extraction.

3.2 Statistical Methods

System ANA [23] automatically extracts concepts from texts to produce a semantic network. MANTEX developed by [42] is founded on the repeated segments principle. In his doctoral thesis, [4] evaluates the statistical approaches capital in locating complex lexical units. MANTEX [45] extracts collocations.

In these methods, pertinent words having only one occurrence are not acquired (silence).

3.3 Mixed Methods

System like ACABIT [14] allows curing the problems of noise which arises in the linguistic methods. XTRACT [52] is a generic tool for location of collocations and not only of the terms.

4 Compound Nouns

In order to constitute our dictionary, a manual collection of the terms is carried out. It is significant to know if a compound noun can constitute an input in the dictionary. We are interested in the linguistic aspects of the terminology and particularly in the compound nouns composition.

4.1 Concept of Composition and Compound Nouns

All the completed research tasks try to define the concept of a compound noun composition but do not provide a magic recipe for their recognition. There is not a single definition of a compound noun but a certain number of common properties. The concept of nominal composition, very much discussed was approached by [10], [11], [29], [5], [6], [7] and [34] who admit that it is not possible to distinguish the compound nouns from free sequences. We will quote other work [37] and [39] for the syntax of the compound nouns, [32] on the lexicon of the compound nouns and [30] for syntactic and semantic automatic processing. Work on compounds N “of” N was studied without giving operational definition of “freezing” [1]. French compound nouns “freezing degree” notions for NN (Noun Noun) and NDN (Noun “of” Noun) categories are defined in [27].

4.2 How to Decide if a Compound Noun Is an Entry of the Dictionary?

A term can be simple if it contains one word or compound if it contains more than one. A compound word is built starting from simple words. Silberztein M. defines a compound noun as a consecutive sequence of at least two simple forms and blocks of separators. A simple form is a nonempty consecutive sequence of characters of the alphabet appearing between two separators. A simple word is a simple form that constitutes an input of dictionary. We will use indifferently term or compound noun to indicate the same concept within the selected technical language (computer science).

A terminological bank is essentially composed of compound nouns. Each linguist uses his own terminology to define the compound words and proposes his own criteria. In our case the simple words are partly listed in the electronic simple words dictionary, the DELAS. The compound words are listed in the DELAC, but the terms concerning the specialized vocabularies are there in a very small number. Our task is to set up the computer science French terms dictionary, this dictionary will be thus a specialized dictionary. Will all the compound nouns found in the texts constitute an input in the dictionary? In order to decide if a word is or not an input of the dictionary we have to know what is a possible lexical entry? In [27] work morpho-syntactic, or semantic criteria are defined to make the distinction between free nominal group and compound noun. In [48] criteria defined a compound word and in [49] work, semantic is used and productive nominal groups and the lexicalised compound nouns are presented. We can find in [26] a study about "frozen sequences" and the semantic factors.

5 The Electronic Dictionary of Computer Science: INFO_COMP

NOOJ [51] is a development environment used to construct large-coverage formalized descriptions of natural languages, and apply them to large corpora, in real time. The description of natural languages are formalized as electronic dictionaries, and grammars represented by organized sets of graphs.

For our specialized dictionary, at first we collected terms from texts, we studied the shape of an input, the compound nouns gender and number, the possible determinants, the particular inflection of the compound nouns and their local grammars. In [47], we can find the plural of some compound nouns. We defined the compound words classes for computer science terminology in [3], [2].

Our dictionary contains about 10 000 compound words. 30 000 compounds are extracted manually and will be added to the dictionary. With NOOJ, we also extract automatically candidate compounds, from texts and corpora. We add manually the terms retained in the dictionary with codifying the entries, it is the terminology acquisition.

5.1 NOOJ Grammars

A NOOJ grammar makes it possible to gather the terms by family; here, a family is the principal word (noun) in the compound, called the "head". A grammar can contain several graphs and can be used for removing ambiguous forms. For the term card we have: card to band, card to card, card to disk…We give the local grammar "carte" (card) in the Fig. 1:

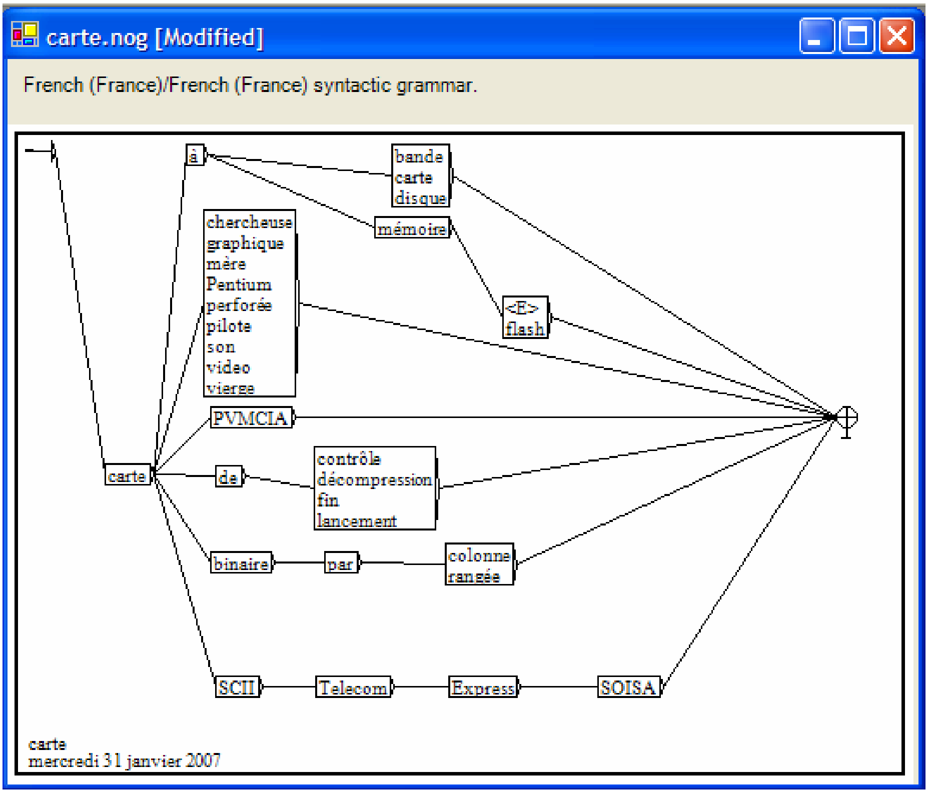


Fig. 1. A local grammar for the term “carte” (card)

5.2 NOOJ Dictionaries

NOOJ dictionaries [51] are a great enhancement over INTEX DELA-type dictionaries [12] as well as lexicon grammars. NOOJ dictionaries can represent spelling and terminological variants. DELAC is the dictionary of the compound nouns. Courtois B. studied binary compounds [13], Gross M. ternary compound nouns [28] and longest (4, 5 and 6 full words), only for French natural language.

In NOOJ, the INTEX dictionaries are represented in one unique format [50] the full description of the inflexion and derivation is encoded inside NOOJ dictionaries for the entries.

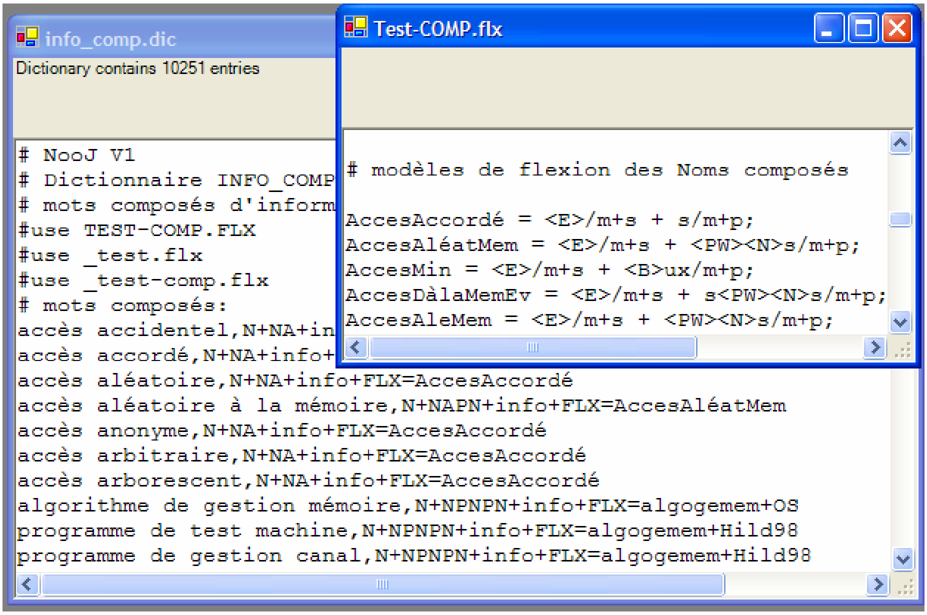
5.3 INFO_COMP Dictionary and Flexional Models .FLX

In the Fig. 2, For the term “accès aléatoire” (random access), we have the entry :

accès aléatoire,N+NA+info+FLX=AccesAccordé

accès aléatoire is a term, for this dictionary entry, we have the category N or NA:

N+NA; info is a semantic information: computer science term;



```
# NooJ V1
# Dictionnaire INFO_COMP
# mots composés d'inform
#use TEST-COMP.FLX
#use _test.flx
#use _test-comp.flx
# mots composés:
accès accidentel,N+NA+in
accès accordé,N+NA+info+
accès aléatoire,N+NA+info+FLX=AccesAccordé
accès aléatoire à la mémoire,N+NAPN+info+FLX=AccesAléatMem
accès anonyme,N+NA+info+FLX=AccesAccordé
accès arbitraire,N+NA+info+FLX=AccesAccordé
accès arborescent,N+NA+info+FLX=AccesAccordé
algorithme de gestion mémoire,N+NPNPN+info+FLX=algogemem+OS
programme de test machine,N+NPNPN+info+FLX=algogemem+Hild98
programme de gestion canal,N+NPNPN+info+FLX=algogemem+Hild98
```

```
# modèles de flexion des Noms composés

AccesAccordé = <E>/m+s + s/m+p;
AccesAléatMem = <E>/m+s + <PW><N>s/m+p;
AccesMin = <E>/m+s + <B>ux/m+p;
AccesDàlaMemEv = <E>/m+s + s<PW><N>s/m+p;
AccesAleMem = <E>/m+s + <PW><N>s/m+p;
```

Fig. 2. Extracts from the dictionary INFO_COMP; the inflectional description Test-COMP

FLX gives the name of the flexional model here AccesAccordé, defined in the inflexional description file Test-COMP.flx:

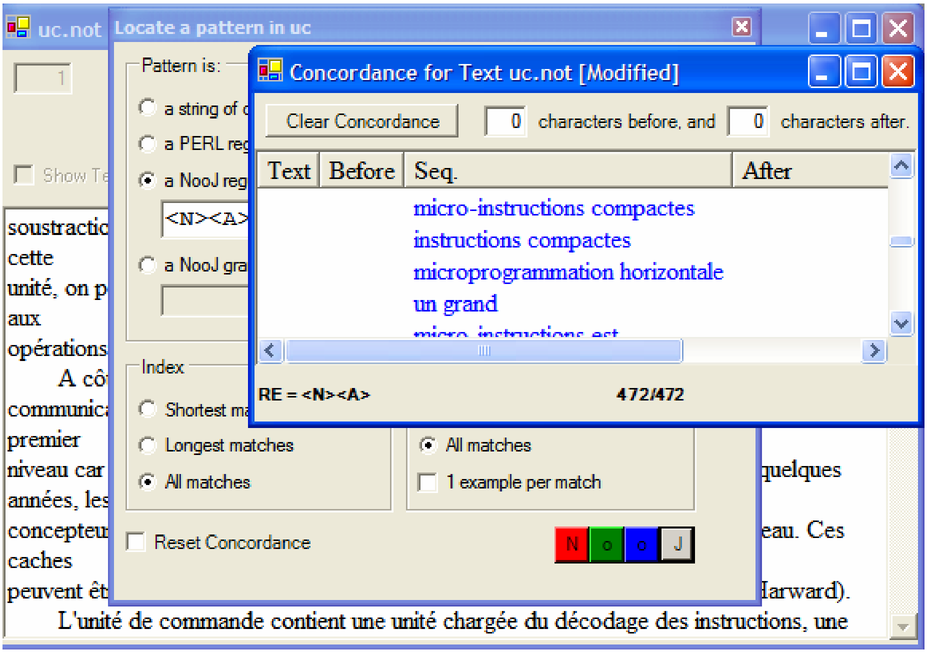
AccesAccordé = <E>/m+s + s/m+p;

The lemma “accès aléatoire” has two forms:

m+s =>	accès aléatoire	masculine (m)	singular (s)
m+p =>	accès aléatoires	masculine (m)	plural (p)

5.4 Automatic Extraction of Compound Terms with NOOJ

A pattern is a NOOJ expression and we can locate any pattern in the text. We use “ **locate a pattern**” to extract compound nouns from texts or corpus, for example, in Fig. 3, a linguistic analysis of the text uc.not is made with NOOJ, the option locate a pattern is used to find the <N><A> (Noun Adjective) terms in the text. The list of concordance (candidate terms) is given, here 472. From the concordance for the text, the linguist selects the terms which are entries for the INFO_COMP dictionary and adds them manually to it (acquisition) with the format given at 5.3. We can locate any pattern we want [51].



Pattern is:
○ a string of c
○ a PERL reg
● a NooJ reg
<N><A>
○ a NooJ gra

Index
○ Shortest ma
○ Longest matches
● All matches

● All matches
□ 1 example per match

□ Reset Concordance

0 characters before, and 0 characters after.

Text	Before	Seq.	After
		micro-instructions compactes	
		instructions compactes	
		microprogrammation horizontale	
		un grand	
		micro-instructions est	

RE = <N><A> 472/472

soustractio
cette
unité, on p
aux
opérations
A cô
communica
premier
niveau car
années, les
concepteu
caches
peuvent êt
quelques
eau. Ces
Harward).
L'unité de commande contient une unité chargée du décodage des instructions, une

Fig. 3. Locating terms with a pattern here <N><A>

6 Conclusion

We started with newspapers terms were collected. These newspapers made it possible to collect terms but not much (3000), in spite of the size of the texts. Specialized books of computer science were taken as corpus and the number of terms clearly increased, because these books are of the studied speciality. Actually we collect manually terms from Hildebert dictionary and automatically from texts with NOOJ. We are building a big corpus of computer science with different texts of computer science from PDF, word and text files. The corpus contains actually 1071 text files; its size is 90 Mo. It remains to test terminology extraction from this corpus.

More than 10 000 terms were extracted and listed manually and added (terminology acquisition) to the INFO_COMP dictionary. 30 000 terms are collected and will be added to the dictionary. The manual collection of the terms is long and setting up a corpus for French computer science terminology is long and not easy. Manual extraction is long and needs to read big corpus. Semantic is often used to decide if a compound is a term. Others information (conceptual, syntactic, synonymous, links, translation in English…) can be added to the entry of the dictionary according to the use that one will make of it.

The elaborate dictionary INFO_COMP and the local grammars will make it possible to analyse computer science corpus, texts. With NOOJ, one will be able with this new dictionary of terminology to treat computer science technical texts and to use them in various applicability such are the automatic indexing, the information retrieval, the machine analysis of texts, and machine translation. A translation in English will be added for each term.

It remains to finalize the coding of all the terms of the dictionary, to set up all the other grammars. Tests are then designed to analyse computer science texts, to index them. We expected to compare between NOOJ linguistic method of indexing and extraction with statistics methods like ANA.

Acknowledgements

I would like to thank Max Silberztein, from the LASELDI laboratory at the Franche-Comté University for his help and Elisabeth Metais from the CEDRIC Laboratory at the CNAM of Paris for all her remarks and help.

References

1. Ancombre, J.C-L.: Pourquoi un moulin à vent n'est pas un ventilateur. In: Langue française 86, Sur les compléments circonstanciels, Paris (1990)
2. Aoughlis, F.: Building an Electronic Dictionary of Computer Science. In: 8th INTEX NOOJ workshop, Besançon (2005), Presentation at http://http://www.nooj4nlp.net
3. Aoughlis, F., Métais, E.: Computer Science Terminology Extraction and Acquisition. WSEAS transactions on Computers 5(10), 2472–2478 (2006)
4. Balvet, A.: Approches catégoriques et non catégoriques en linguistique des corpus spécialisés, application à un système de filtrage d'information, doctorat thesis, université Paris X, Nanterre (2002)
5. Bauer, L.: English Word Formation. Cambridge University Press, Cambridge (1983)
6. Bauer, L.: Introducing linguistic morphology. Edinburgh university press, Edinburgh (1988)
7. Benveniste, E.: Fondements syntaxiques de la composition nominale » et « formes nouvelles de la composition nominale. In: Problèmes de la linguistique générale, Gallimard, Paris, vol. 2, pp. 145–176 (1974)
8. Bourigault, D.: LEXTER, un logiciel d'extraction de terminologie. Application à l'extraction de connaissances à partir de textes, doctorat thesis, Paris, Ecole des hautes Etudes en Sciences Sociales (1994)
9. Bourigault, D., Lepine, P.: Utilisation d'un logiciel d'extraction de terminologie (LEXTER) en acquisition de connaissances. In: Acquisition et ingénierie des connaissances: tendances actuelles, Cepaduès (1995)
10. Cadio, P.: À entre deux noms : vers la composition nominale. In: Lexique 1, PUL pp. 193–240 (1992)
11. Corbin, D.: Hypothèses sur les frontières de la composition nominale. In: Cahiers de grammaire, 17, Novembre 1992, le MIR, université de Toulouse (1992)
12. Courtois, B., Silberztein, M.: Dictionnaires électroniques du français. In: Langue Française, vol. 87, Larousse, Paris (1990)

13. Courtois, B., Garrigues, M., Gross, M., Gross, G., Rung, M., Mathieu-Colas, Silberztein, M., Vives, R.: Dictionnaire électronique des noms composés DELAC : les composants NA et NN, version 4.0, technical report LADL N° 55 (1997)
14. Daille, B.: Approche mixte pour l'extraction automatique de terminologie: statistique lexicale et filtres linguistiques, Thèse de doctorat en informatique fondamentale, université de Paris7 (1994)
15. David, S., Plante, P.: De la nécessité d'une approche morpho-syntaxique en analyse de textes. In: OICO, vol. 2(3), Québec, pp. 140–155 (1990)
16. De Chalendar, G.: STEVLAN: un système de structuration du lexique guidé par la détermination automatique du contexte thématique ».Thèse de docteur de l'université de Paris XI, Orsay, (2001)
17. De Chalendar, G.: Les systèmes d'acquisition de connaissances à partir de textes, (2002) http : // www.limsi.fr/individu/gael/manuscritThèse/html/node64.html
18. Drouin, P.: Acquisition automatique de termes : l'utilisation des pivots lexicaux spécialisés, Thèse de doctorat présenté à l'Université de Montréal, Montréal (2002)
19. Drouin, P.: Extraction de termes : techniques courantes. In: Ateliers sur les corpus spécialisés en terminologie, (2003) http://www.ling.umontreal.ca/lhomme/cgi-bin/PD.zip
20. Drouin, P.: Term extraction using non technical corpora as a point of leverage. Terminology 9(1), 99–117 (2003)
21. Drouin, P.: Acquisition des termes simples fondé sur les pivots lexicaux spécialisés ». In: Actes des cinquièmes rencontres Terminologie et intelligence artificielle (TIA), Strasbourg, pp. 183–186 (2003)
22. Dubois, G., et al.: Dictionnaire de linguistique, édition Larousse (1973)
23. Enguehard, C.: Acquisition naturelle automatique d'un réseau sémantique, thèse de doctorat de l'UTC, Compiègne (1992)
24. Fotopoulou, A.: Analyse automatique des textes: Dictionnaires et thésaurus électroniques des termes des télécommunications, Actes de la 5-ème journée ERLA-GLAT Brest, université de Bretagne occidentale (1994)
25. Fotopoulou, A.: Elaboration d'un thésaurus des termes des télécommunications, rapport, Institut National des télécommunications, Program Human Capital & Mobility (1995)
26. Gonzales Rey, M.I., Lopez Diaz, M.: De l'opacité des séquences figées comme exception sémantique. In: Actes-Exception-2003 (2003) http://www.cavi.univ-paris3.fr/ilpga/Actes-Exception-2003/Auteurs/lopezdiaz-gonzalezrey/texte.pdf
27. Gross, G.: Définition des noms composés dans un lexique-grammaire. In: Langue Française, septembre 1990, vol. 87 (1990)
28. Gross, M., Courtois, B.: Dictionnaire électronique DELAC: les noms composés ternaires et à plus de trois constituants, LADL (Décembre 1998)
29. Habert, B.B., Jacquemin, Ch.: Noms composés, termes, dénominations complexes: problématiques linguistiques et traitements automatiques. In: traitement automatique des langues, vol. 2, Traitement automatique de la composition nominale (1993)
30. Habert, B., Jacquemin, C.: Constructions nominales à contraintes fortes et grammaires d'unifications. linguisticae Investigationes XIX-2, 401–427 (1995)
31. Jacquemin, Ch.: Variation terminologique et acquisition automatique de termes et leurs variantes en corpus. Habilitation à diriger des recherches en informatique, IRIN, université de Nantes (1997)
32. Jung, R.: Remarques sur la constitution du lexique des noms composés. In: Langue française, Paris, vol. 87 (1990)

33. Lemay, C.: Utilisation de corpus de référence pour dégager la terminologie d'un corpus ». In: Ateliers sur les corpus spécialisés en terminologie, Downloadable presentation. (2003), http://www.ling.umontreal.ca/lhomme/cgi-bin/CL2.zip
34. Levi, J.: The syntax and semantics of complex nominals. Academic Press, London, New York (1978)
35. L'homme, M.C.: Évaluation de logiciels d'extraction de terminologie : examen de quelques critères ». In: JIAMCATT, Réunion Inter institutions sur la terminologie et la traduction assistées par ordinateur, Office des Nations Unies, Vienne (Autriche) (2000)
36. L'homme, M.C.: Nouvelles technologies et recherche terminologique. Techniques d'extraction des données terminologiques et leur impact sur le travail du terminographe. L'Impact des nouvelles technologies sur la gestion terminologique, (2001) (Downloadable text)
37. Mathieu Colas, M.: Typologie des noms composés. In: Rapport de recherches n° 7 du LLI, Université de Paris 13 (1989)
38. Meilland, J.C., Bellot, P.: Extraction de terminologie à partir de libellés textuels courts. In: P.U.R., (Presses Universitaires de Rennes), Linguistique de corpus (2003)
39. Monceaux, A.: La formation des noms composés de structure Nom Adj, élaboration d'un lexique électronique, Doctorat thesis, université de MLV, Noisy le Grand (1994)
40. Morin, E.: Extraction de liens sémantiques entre termes à partir de corpus de textes techniques, Doctorat thesis, IRIN, Nantes (1999)
41. Orliac, B.: Acquisition de collocations verbe + nom à partir de représentations syntaxiques. In: Ateliers sur les corpus spécialisés en terminologie (2003) Downloadable version at http://www.ling.umontreal.ca/lhomme/cgi-bin/BO.zip
42. Oueslati, R.: Aide à l'acquisition de connaissances à partir de corpus. Thèse de doctorat, Université Louis Pasteur Strasbourg (1999)
43. Poibeau, T., Dutoit, D., Bizouar, S.: Evaluating Resource Acquisition Tools for Information Extraction. In: Proceeding of the International language Resource and Evaluation Conference, (LRERC 2002),Las Palmas, Les canaries (2002)
44. Reymond, D.: La co-occurence en T.A.L: dis moi qui tu fréquentes et je te dirai qui tu es ». In: L'Ecriture dans tous ses états. Approche en sciences cognitives. Colloque 20 et 21 mai 2003, Université d'Aix en Provence (2003)
45. Rochibeau, B.: Extraction de collocations fondée sur les méthodes statistiques, in Ateliers sur les corpus spécialisés en terminologie (2003) downloadable version at http://www.ling.umontreal.ca/lhomme/cgi-bin/BR.zip
46. Savary, A.: Recensement et description des mots composés- Méthodes et applications. Doctorat thesis, université de Paris7 (2000)
47. Silberztein, M.: Le dictionnaire électronique des mots composés ». In: Langue Française, septembre 1990, vol. 87 (1990)
48. Silberztein, M.: Dictionnaires électroniques et analyse automatique de textes: le système INTEX, Edition Masson, Paris (1993)
49. Silberztein, M.: Les groupes nominaux productifs et les noms composés lexicalisés. In: Lingvisticae Investigationes XVII: 2, John Benjamins B.V, Amsterdam (1993)
50. Silberztein, M.: NOOJ's dictionaries. In: The Proceedings of the LTC (2005)
51. Silberztein, M.: NOOJ'manual, (2006), http://www.nooj4nlp.net
52. Smadja, F.: XTRACT: an overview. Computers and Humanities 26, 399–413 (1993)
53. Smadja, F., McKeown, K., Hatzivassiloglou, V.: Translating collocations for bilingual lexicons: A statistical approach. Computational Linguistics 22(1), 1–38 (1996)
54. Term: Constitution de corpus à partir du Web pour l'acquisition automatique de terminologie: une expérience, (2003) www.poleia.lip6.fr/ slodzian/sberland/

Appendix: INFO_COMP Dictionary Extract for the Term “Accès”

accès accidentel
accès accordé
accès aléatoire
accès aléatoire à la mémoire
accès anonyme
accès arbitraire
accès arborescent
accès au destinataire
accès au support technique
accès autorisé
accès aux bases de données
accès aux données
accès aux services supplémentaires
accès califourchon
accès complet
accès complet au réseau
accès concurrentiel
accès conflictuel
accès d’arrivée
accès d’objet
accès de base
accès de départ
accès de recherche
accès de test
accès désiré
accès direct   glossaire
accès direct à la mémoire
accès direct à la mémoire évolué
accès direct au programme
accès direct aux mémoires
accès direct en mémoire
accès direct mémoire
accès discret
accès disqu

Applying Wikipedia's Multilingual Knowledge to Cross–Lingual Question Answering[*]

Sergio Ferrández, Antonio Toral, Óscar Ferrández, Antonio Ferrández, and Rafael Muñoz

Natural Language Processing and Information Systems Group
Department of Software and Computing Systems
University of Alicante, Spain
{sferrandez,atoral,ofe,antonio,rafael}@dlsi.ua.es

Abstract. The application of the multilingual knowledge encoded in Wikipedia to an open–domain Cross–Lingual Question Answering system based on the Inter Lingual Index (ILI) module of EuroWordNet is proposed and evaluated. This strategy overcomes the problems due to ILI's low coverage on proper nouns (Named Entities). Moreover, as these are open class words (highly changing), using a community–based up–to–date resource avoids the tedious maintenance of hand–coded bilingual dictionaries. A study reveals the importance to translate Named Entities in CL–QA and the advantages of relying on Wikipedia over ILI for doing this. Tests on questions from the Cross–Language Evaluation Forum (CLEF) justify our approach (20% of these are correctly answered thanks to Wikipedia's Multilingual Knowledge).

1 Introduction

Currently, the exponential growth of digital information requires processes capable of searching, filtering, retrieving and classifying this information. Moreover, the information required by the users might be in different languages. Nowadays, one of the most demanded way of accessing multilingual information is to obtain information from sources written in different languages than that of input queries. Obviously, multilinguality is one of the main difficulties that impedes the right acquisition of information.

For this purpose, Computational Linguistics applications such as Information Retrieval (IR) and Question Answering (QA) are used. IR is the science of searching for documents that contain the information required by the user, whereas QA can be defined as the task consisting of answering precise and arbitrary questions formulated by the user. The aim of a QA system is to find the correct answer to user questions in a non-structured collection of documents. In Cross–Lingual (CL) environments, the question is formulated in a different

[*] This work has been developed in the framework of the project QALL-ME, which is a 6th Framework Research Programme of the European Union (EU), contract number: FP6-IST-033860.

Z. Kedad et al.(Eds.): NLDB 2007, LNCS 4592, pp. 352–363, 2007.

language from that of the documents, which increases the difficulty. As it was revealed in the Cross-Language Evaluation Forum (CLEF) 2006 [15], multilingual tracks of IR and QA tasks have been recognized as an important issue in information access.

In this paper, we present a novel approach for solving the CL–QA task. Our strategy consists of a CL–QA system [9], which performs the references between words in different languages using the Inter Lingual Index (ILI) module of EuroWordNet (EWN) [22] as well as the multilingual relations encoded in Wikipedia[1]. The original contribution of this research consists of the application of Wikipedia's Multilingual Knowledge (WMK) in order to overcome ILI's low recall regarding proper nouns and as an affordable alternative to other approaches that rely on hand–coded dictionaries of proper nouns and therefore avoiding maintenance. Besides, a detailed study justifying the need to translate this kind of nouns within CL–QA is included.

The rest of the paper is organized as follows: section 2 describes the background of current CL–QA systems Afterwards, our CL–QA system based in ILI is presented. This is followed by a detailed description about the integration of WMK. Next, section 5 illustrates a study about the need for translating Named Entities in CL–QA. In section 6, an evaluation regarding CLEF official questions is presented. Finally, section 7 wraps up the paper with our conclusions and future work proposals.

2 Background

The overall accuracy of CL–QA systems is directly affected by their ability to correctly analyze and translate the question that is received as input. An imperfect or fuzzy translation of the question causes a negative impact on the overall accuracy of the systems. According to [17], the Question Analysis phase is responsible for 36.4% of the total of number of errors in open–domain QA.

The last edition of CLEF (2006) [15] has confirmed that most of the implementations of current CL-QA systems [5,13,18,20,21]are based on the use of on-line translation services. However, a recent research [8] presents a study detailing the common errors produced by Machine Translation (MT) based systems and proposes an alternative approach to overcome such errors.

This revision of the state of the art focuses on the bilingual English–Spanish QA task, because the CL–QA system used for the evaluation works in these languages. In CLEF 2006, three different approaches have been presented by CL-QA systems as solutions for the bilingual English–Spanish task.

The first one [6] translates entire documents into the language in which the question is formulated. This system uses a statistical MT system that has been trained using the European Parliament Proceedings Parallel Corpus 1996–2003 (EUROPARL).

The second system [23] uses an automatic MT tool to translate the question into the language in which the documents are written. This strategy is

[1] www.wikipedia.org

the simplest technique available. In this case, when comparing to the Spanish monolingual task, the system loses about 55% of this precision in the CL task.

The third system [12] translates the question using different on-line machine translators and some heuristics. This technique consults several web services in order to obtain an acceptable translation.

The previously described strategies are based on the use of MT in order to carry out the bilingual English–Spanish task, and all of them try to correct the translation errors through different heuristics.

The translations are often inexact and quite fuzzy. Besides, the MT systems resolve the ambiguity by means of only giving one translation per word. These facts cause an important negative impact on the precision of the systems. This can be checked on the last edition of CLEF 2006 where the cross lingual systems obtained less than 50% of correct answers compared to the monolingual task.

For instance, MT systems generate errors [8] such as translations of names that should be left untranslated, translations of polysemous words where the sense translated is not the correct one, syntactic errors in the translation, wrong translations of interrogative particles, incorrect lexical-syntactic category of the translated words and unknown words by the MT and thus left untranslated. The impact of this kind of mistakes should be controlled and evaluated.

In the next sections, our strategy of CL–QA system and the integration of WMK in order to control the references between languages are detailed.

3 System Description

In this section, the architecture and functionality of our method to open domain CL-QA [7] are detailed. A graphic depicting the overall architecture of the system is shown in figure 1.

The system is designed to localize answers from documents, where both answers and documents are written in different languages. The system is based on complex syntactic pattern matching using Computational Linguistics tools [1,14,19]. Also, a new proposal of Word Sense Disambiguation (WSD) for nouns (presented in [11]) is applied to improve the precision of the system.

The fundamental and original characteristic of our approach is the strategy used for the Inter Lingual Reference (ILR) Module in which the ILI Module of EuroWordNet (EWN) [22] is used with the aim of reducing the negative effect of question translation on the overall accuracy. This multilingual knowledge source is used to reference verbs, common nouns and proper nouns (named entities).

Named Entities (NEs) contained in the input questions are identified and classified by the Named Entity Recognition (NER) NERUA system. Four entity types are considered: person (PER), location (LOC), organization (ORG) and miscellaneous[2] (MISC). The recognition of NEs makes the ILR module capable of carrying out a customized treatment for each entity type.

The strategy followed by the ILR module introduces two improvements:

[2] This entity type is assigned when a detected entity cannot be enclosed in any of the remaining ones. E.g. Maastricht treaty (in question 13 of QA–CLEF 2006).

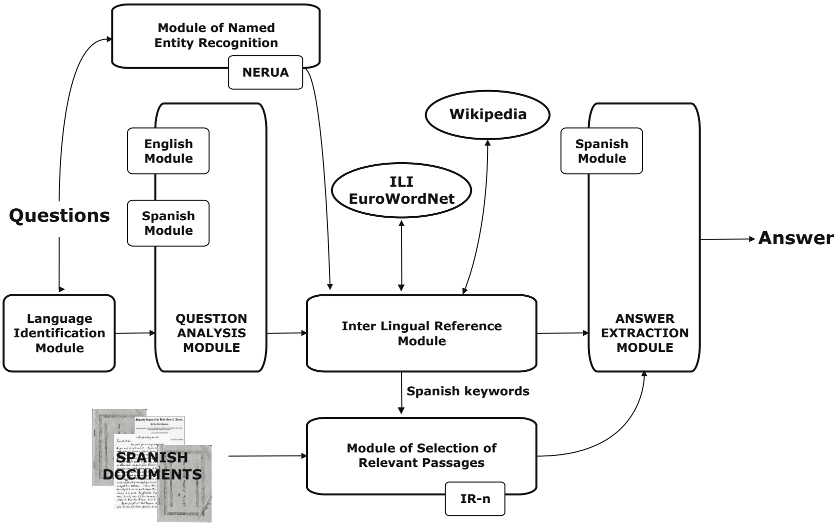


Fig. 1. Architecture of the system

(1) The consideration of more than one translation per word by means of using the different synsets of each word in the ILI module of EWN. Figure 2 shows the references provided by the ILI module for the input word "*president*" in English when the target language is Spanish.

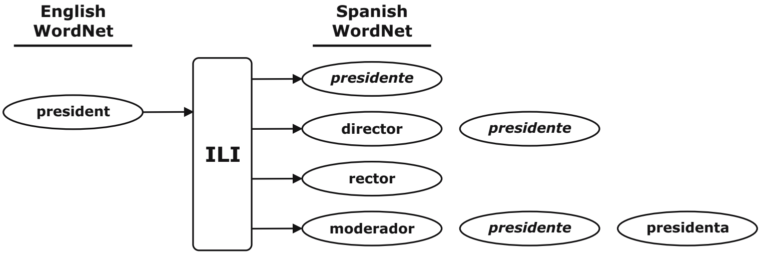


Fig. 2. Links to the word "*president*"

As can be seen in figure 2, in some cases the ILR module obtains more than one Spanish equivalent for each English word. The current strategy employed to get the best translation consists of assigning a weight depending on the frequency of each word in ILI. In this case, the most weighted Spanish word is *"presidente"*. This strategy improves the method commonly followed by MT services in which only one possible translation is given for each word.
(2) Unlike the current bilingual English–Spanish QA systems, the question analysis is developed in the original language without any translation. The

system develops two main tasks in the question analysis phase using a set of syntactic patterns:

- The detection of the expected answer type. The system detects the type of information that the answer has to satisfy to be a candidate of an answer (proper nouns, quantity, date, ...).
- The identification of the main Syntactic Blocks (SB) of the question. The system extracts the SB that are necessary to find the answers.

In order to show the complete process, an example of a question at CLEF 2006 is provided:

- **Question 107 at CLEF 2006:** *How many soldiers does Spain have?*
- **SB:**
  [Noun Phrase *soldier*]
  [Verb Phrase *to have*]
  [Noun Phrase *Spain*]
- **Type:** entity-amount
- **Keywords to be referenced with ILI:** soldier have Spain
  soldier ↦ soldado
  have ↦ estar-enfermo *tener* padecer sufrir causar inducir hacer consumir tomar ingerir experimentar *tener* poseer *tener* recibir aceptar querer constar figurar existir
  Spain ↦ España

On the other hand, the verbs and common nouns that are not referenced in ILI are translated into Spanish using an on-line Spanish Dictionary[3]. Moreover, in order to decrease the effect of incorrect translation of the proper nouns, the matches using these words in the search of the answer are realized using the set of translated words and the original word of the question. The matches found using the original English word are valued at 20% less.

The final step of the CL–QA process is the Extraction of the Answer. The system uses the syntactic blocks of the question and different sets of syntactic patterns (according to the type of the question) with lexical, syntactic and semantic information to find out the correct answer.

In the next section, our novel strategy which integrates multilingual knowledge from Wikipedia in order to translate named entities is presented.

4 Integrating Wikipedia's Multilingual Knowledge in CL-QA

The main drawback of using ILI is that it contains very few proper nouns.[4] In fact, according to [16], WordNet 1.6 contains 3,876 proper nouns. This word

[3] http://www.wordreference.com

[4] The word class corresponding to the NE types considered: person, location, organization and miscellaneous.

class is highly evolving, meaning that new proper nouns appear continuously. As ILI is a hand-tagged resource developed by a small number of linguist experts, it becomes obvious that it would be tedious and time-consuming to maintain a considerable amount of proper nouns within its infrastructure.

Exploiting Wikipedia is an appropriate way in order to fill this gap. Wikipedia is an encyclopedia written in a collaborative way[5] that contains a huge amount of proper nouns,[6] and like this word class, this resource is continuously updated. Moreover, it has multilingual links that reference entries in an input language with their equivalents in other languages.

Wikipedia has been already employed within monolingual QA [4]. However, although their multilingual capabilities have been used for tasks such as multilingual corpora creation [3] and discovery of related entries of Wikipedia in different languages [2], to our knowledge, they have not been applied within the CL–QA environment. The following example shows how CL–QA can benefit from the incorporation of this knowledge.

- **Question 186 at CLEF 2006:** *In which town in Zeeland did Jan Toorop spend several weeks every year between 1903 and 1924? ("¿En qué ciudad de Zelanda pasaba varias semanas al año Jan Toorop entre 1903 y 1924?")*

  The question contains two proper nouns: "*Zeeland*" and "*Jan Toorop*". None of them is referenced in ILI. However, both have an entry in the English version of Wikipedia, and both entries contain a reference to their Spanish equivalents: "*Zelanda*" and "*Jan Toorop*" respectively. Furthermore, if this question would have been translated by a MT service, the string "*Jan Toorop*" would have been converted to "*Enero Toorop*" interpreting that Jan states for January.

To incorporate WMK into our CL–QA system, the ILR module performs a special treatment of NEs that depends on the entity type (this decision will be justified in the study presented in section 5). Person entities are directly translated by WMK whereas the remaining entity types are translated by ILI, and if no translation is found in this resource, WMK is used. The hypothesis is that both resources contain complementary information and therefore a combination of them could achieve better CL–QA performance.

In order to include WMK in our system, database dumps[7] provided by the Wikimedia Foundation were downloaded and tailored for our specific needs as well as for efficiency reasons. Besides, an API to access this database and gather the required information was developed. Both this API and utilities to download, import and tailor Wikimedia database dumps are part of the software wiki_db_access, which has been released with a free license with the aim that it could be useful for research purposes.[8]

[5] On 2007/01/16 the English version had 3, 247, 299 registered users.

[6] The dump used contains 1, 496, 097 encyclopedic entries.

[7] Available at `http://download.wikimedia.org`

[8] Available at `http://www.dlsi.ua.es/~atoral/`

Table 1. Percentage of questions containing NEs and percentage of NEs that should be translated

Dataset	Questions					
		overall	PER	LOC	ORG	MISC
Questions CLEF 2004	with NEs	81%	23.5%	28%	15%	20.5%
	NEs should be translated	44.89%	2.1%	60.7%	56.7%	48.8%
Questions CLEF 2005	with NEs	93%	34%	25.5%	24%	13.5%
	NEs should be translated	36%	10.3%	50.9%	39.6%	55.5%
Questions CLEF 2006	with NEs	89%	31%	24.5%	22.5%	24%
	NEs should be translated	42.69%	3.2%	65.3%	40%	50%
Average	with NEs	87.7%	29.5%	26%	20.5%	19.3%
	NEs should be translated	41.2%	5.2%	59%	45.4%	51.4%

5 The Need for Translating NEs in CL–QA

This section presents a minute study on the need for translating NEs in CL–QA. The dataset used has been the official 600 English questions of CLEF 2004, 2005 and 2006. The aim of this study is to find out solutions in order to overcome the errors in the translation of NEs between different languages. We provide results on how important it is to translate NEs in CL–QA and how they can be successfully translated.

Table 1 presents the results on our study to find out the percentage of questions that contain NEs and the percentage of these NEs that need to be translated. The percentage of questions with NEs is quite high (81% for 2004, 93% for 2005 and 89% for 2006, i.e. 87.7% on average). From these entities, nearly half of them should be translated (44.89% for 2004, 36% for 2005 and 42.69% for 2006, i.e. 41.2% on average). The remaining percentage of NEs should not be translated (for all of these NEs, no reference is found in ILI[9] while most of them are present in Wikipedia but their name both in the input and target language is the same). Regarding the entity types, it can be seen that it is very important for CL–QA to translate locations, organizations and miscellaneous entities while the impact of not treating person entities would be low.

We have discovered that most of the mistakes regarding wrong ILI references are caused by trying to translate a word that should not be translated (e.g. a person name). Being person entities those with a lower need to be translated, and being ILI a resource with low recall regarding proper nouns, it is for this entity type that ILI obtains the worst performance. Table 2 shows the percentage of person entities that is wrongly translated by ILI in the question sets. Roughly, 30% of person entities are wrongly translated, which has a considerable impact for the CL–QA process.

[9] Even if any of these NEs would be incorrectly translated by ILI, our CL–QA system takes into account as well as the translated NEs, these NEs in the original language.

Table 2. Percentage of wrong translation of type person using the ILI mdule

Dataset	Wrong Translation
Person Entities from CLEF 04	28.6%
Person Entities from CLEF 05	27.6%
Person Entities from CLEF 06	30.3%
Average	28.8%

The following example (see Table 3) shows a case in which ILI fails to translate person NEs whereas WMK provides the correct reference in the target language. This justifies our decision to directly translate person entities by means of using WMK. In this example, the proper noun "*Jan*" is confused with the abbreviation of the month "*January*" by the ILI module of EuroWordNet while WMK provides the correct reference in Spanish. In these cases, the need for some kind of treatment such as NER is clear in order to classify entities and therefore to perform a specialized treatment of NEs depending on the entity type. This will be discussed in the following section.

Table 3. Question 184 CLEF 2006

Language	Question 184 CLEF 2006
English	Who is Jan Tinbergen?
Spanish	Quién es Jan Tinbergen?
Translated Keywords	
using ILI	**enero** Tinbergen
using WMK	Jan Tinbergen

In a nutshell, the study has proved that it is important to translate NEs in CL–QA. It has also been revealed that a specialized treatment should be carried out depending on the entity type. Concretely, ILI's performance for person entities is very low. In fact, the CL–QA system obtains better results if person entities are not translated at all than if they are translated by ILI. However, the idiosyncracies of WMK provide a treatment of person entities that overcome ILI's limitations.

6 Experimental Results

6.1 Evaluation Environment

For carrying out this evaluation, the CLEF 2004, 2005 and 2006 sets of 600 English and Spanish questions and the EFE 1994–1995 Spanish corpora are used. These corpora provide a suitable framework in order to check the CL–QA system precision.

The set of questions is composed of "*factoid questions*" and "*definition questions*". The factoid questions are fact–based questions, asking for the name of a person, a location, the extent of something, the day on which something happened, etc.

Furthermore, with regard to the corpora created for training the NE recognizer, we have carried out the following strategy. We have manually annotated all the question datasets (2004, 2005 and 2006) and in order to apply NER to the 2006 question set, we have used as a training corpora the question sets belonging to 2004 and 2005 editions. For the 2005 question set, the 2004 and 2006 datasets were used as a train, and finally, for the 2004 question set we have merged the 2005 and 2006 question sets in order to create the training corpus.

Regarding WMK, we have used the English database dump provided by Wikimedia (enwiki-20061104) and specifically the page, pagelinks and langlinks data.

6.2 Result Analysis

The aim of these experiments is to evaluate the impact of applying WMK to our CL–QA system. We show the recall performance for translating entities obtained by both ILI and WMK. Besides, we provide the precision of our CL–QA system and compare it with the precision of our monolingual system.

From the NEs that should be translated, table 4 shows the percentage that are translated by using ILI and from the NEs not translated by ILI it shows the percentage that is translated by WMK. Although ILI is able to translate barely half of the NEs (57,3% for LOC, 39,8% for ORG and 59,6% for MISC, i.e. 39.1%

Table 4. NEs translated by ILI and WMK

Dataset		**ILI**	**WMK**
CLEF'04	PER	-	100%
	LOC	54.5%	90%
	ORG	29.4%	75%
	MISC	85%	100%
CLEF'05	PER	-	100%
	LOC	54.5%	93.3%
	ORG	10.5%	94.1%
	MISC	31.3%	90.9%
CLEF'06	PER	-	100%
	LOC	62.8%	84%
	ORG	50%	88.8%
	MISC	62.5%	88.8%
Average	PER	-	100%
	LOC	57.3%	89.1%
	ORG	39.8%	86%
	MISC	59.6%	93.2%
	TOTAL	39.2%	92.1%

in TOTAL), this is overcome by applying WMK (100% for PER, 89,1% for LOC, 86% for ORG and 93,2% for MISC, i.e. 92.1% in TOTAL).

Table 5 shows the precision of our system[10] in the CL scenario (questions in English and documents in Spanish) compared with the monolingual one (questions and documents in Spanish) for the questions sets of CLEF 2004, 2005 and 2006. Regarding the CL scenario, we not only show the total precision, but also provide the percentage of precision that is obtained thanks to the use of WMK (see second row of table 5). The importance of applying WMK to the CL–QA system is corroborated by these results as around 20% of the questions are correctly answered because of the incorporation of this module (18% for 2004, 23.5% for 2005 and 16% for 2006).

Table 5. QA system evaluation

Dataset		**Prec.**
English Questions (CL, % total answered)	CLEF'06	44%
	CLEF'05	42.5%
	CLEF'04	33.5%
English Questions (CL, % answered by using WMK)	CLEF'06	16%
	CLEF'05	23.5%
	CLEF'04	18%
Spanish Questions (monolingual, % total answered)	CLEF'06	50.5%
	CLEF'05	51.5%
	CLEF'04	41.5%

Compared to other state–of–the–art CL–QA systems, our approach obtains better results [10]. In fact, our precision loss of CL with respect to the monolingual run is around 17% whereas in the English–Spanish QA task at CLEF 2006 [15] the precision on English–Spanish CL–QA task was approximately 50% lower than for the monolingual Spanish task.

7 Conclusions

This paper has presented a novel approach that consists of applying multilingual knowledge encoded in Wikipedia to a CL–QA system based on the ILI module of EWN in order to improve the translation of NEs contained in the input questions. This original strategy to use WMK within CL–QA is motivated by two reasons that are proved by the evaluation results presented in the current paper: (i) the small percentage of NEs referenced in ILI (39.2% of NEs that should be translated in CLEF 2004, 2005 and 2006 questions) and (ii) the need to translate NEs in CL–QA environments (41.2%). A study that demonstrates the latter hypotheses has been presented and discussed.

[10] To calculate this value, both correct and the inexact answers that contain more information than that required by the query are considered.

The proposed approach has been evaluated on CLEF 2004, 2005 and 2006 English–Spanish CL–QA questions. For each year question set we provide the percentage of NEs that is translated by using ILI and, from the remaining NEs (those that ILI does not translate), the percentage that gets translated by applying WMK. The results prove that although ILI leaves a considerable percentage of NEs untranslated (ILI successfully translates between 39,8% and 59,6% of the entities), WMK succeeds to translate on average between 86% and 100% of these NEs depending on the entity type. Moreover, around 20% of the input questions are correctly answered by the CL–QA system as a consequence of using WMK. Besides, our CL–QA system has been evaluated by comparing the precision obtained at both CL and monolingual scenarios. The precision loss remains lower (around 17%) than for other state-of-the-art systems (around 50%).

Another contribution of this paper is the release as free software of the software tools used to process and gather information from Wikimedia database dumps.

Finally, as a future work proposal, we would like to take advantage of the knowledge that can be acquired by employing both multilingual resources incorporated in our system (ILI and WMK). In order to do this we plan to study strategies to combine in different ways the knowledge present in both resources.

References

1. Acebo, S., Ageno, A., Climent, S., Farreres, J., Padró, L., Placer, R., Rodriguez, H., Taulé, M., Turno, J.: MACO: Morphological Analyzer Corpus-Oriented. ESPRIT BRA-7315 Aquilex II, Working Paper 31 (1994)
2. Adafre, S., Jijkoun, V., Rijke, M.: The University of Amsterdam at WiQA 2006. In: Workshop of Cross-Language Evaluation Forum (CLEF), Alicante, Spain (2006)
3. Adafre, S., Rijke, M.: Finding similar sentences across multiple languages in wikipedia. In: EACL 2006 Workshop on New Text, Wikis and Blogs and Other Dynamic Text Sources, Trento, Italy (2006)
4. Ahn, D., Jijkoun, V., Mishne, G., Müller, K., de Rijke, M., Schlobachz, S.: Using Wikipedia at the TREC QA Track. In: Proceedings of the Thirteenth Text Retrieval Conference (TREC 2004) (2005)
5. Bos, J., Nissim, M.: Cross-Lingual Question Answering by Answer Translation. In: Workshop of Cross-Language Evaluation Forum (CLEF) (2006)
6. Bowden, M., Olteanu, M., Suriyentrakorn, P., Clark, J., Moldovan, D.: LCC's PowerAnswer at QA@CLEF 2006. In: Workshop of Cross-Language Evaluation Forum (CLEF) (2006)
7. Ferrández, S., Ferrández, A.: Cross-lingual question answering using inter lingual index module of eurowordnet. Advances in Natural Language Processing. Research in Computing Science 18, 177–182 (2006) ISSN: 1665-9899 18
8. Ferrández, S., Ferrández, A.: The Negative Effect of Machine Translation on Cross-Lingual Question Answering. In: Gelbukh, A. (ed.) Computational Linguistics and Intelligent Text Processing. LNCS, vol. 4394, pp. 494–505. Springer, Heidelberg (2007)
9. Ferrández, S., Ferrández, A., Roger, S., López-Moreno, P., Peral, J.: BRILI, an English-Spanish Question Answering System. In: Proceedings of the International Multiconference on Computer Science and Information Technology. 1st International Symposium Advances in Artificial Intelligences and Applications (AAIA '06), pp. 23–29 (November 2006) ISSN 1896-7094

10. Ferrández, S., López-Moreno, P., Roger, S., Ferrández, A., Peral, J., Alavarado, X., Noguera, E., Llopis, F.: AliQAn and BRILI QA System at CLEF-2006. In: Workshop of Cross-Language Evaluation Forum (CLEF) (September 2006)
11. Ferrández, S., Roger, S., Ferrández, A., Aguilar, A., López-Moreno, P.: A new proposal of word sense disambiguation for nouns on a question answering system. Advances in Natural Language Processing. Research in Computing Science 18, 83–92 (2006) ISSN: 1665-9899
12. García-Cumbreres, M.A., Ureña-López, L.A., Martínez-Santiago, F., Perea-Ortega, J.M.: BRUJA System. The University of Jaén at the Spanish task of CLEFQA 2006. In: Workshop of Cross-Language Evaluation Forum (CLEF) (September 2006)
13. Gillard, L., Sitbon, L., Blaudez, E., Bellot, P., El-Béze, M.: The LIA at QA@CLEF-2006. In: Workshop of Cross-Language Evaluation Forum (CLEF) (September 2006)
14. Llopis, F., Vicedo, J.L.: Ir-n, a passage retrieval system. In: Workshop of Cross-Language Evaluation Forum (CLEF) (2001)
15. Magnini, B., Giampiccolo, D., Forner, P., Ayache, C., Jijkoun, V., Osevona, P., Peñas, A., Rocha, P., Sacaleanu, B., Sutcliffe, R.: Overview of the CLEF 2006 Multilingual Question Answering Track. In: Workshop of Cross-Language Evaluation Forum (CLEF) (September 2006)
16. Magnini, B., Negri, M., Preete, R., Tanev, H.: A wordnet-based approach to named entities recognition. In: Proceedings of SemaNet '02: Building and Using Semantic Networks, Taipei, Taiwan, pp. 38–44 (2002)
17. Moldovan, D.I., Pasca, M., Harabagiu, S.M., Surdeanu, M.: Performance issues and error analysis in an open-domain question answering system. ACM Trans. Inf. Syst 21, 133–154 (2003)
18. Sacaleanu, B., Neumann, G.: DFKI-LT at the CLEF 2006 Multiple Language Question Answering Track. In: Workshop of Cross-Language Evaluation Forum (CLEF) (September 2006)
19. Schmid, H.: TreeTagger — a language independent part-of-speech tagger. Institut fur Maschinelle Sprachverarbeitung, Universitat Stuttgart (1995)
20. Sutcliffe, R.F.E., White, K., Slattery, D., Gabbay, I., Mulcanhy, M.: Cross-language French-English Question Answering using the DLT System at CLEF 2006. In: Workshop of Cross-Language Evaluation Forum (CLEF) (September 2006)
21. Tomás, D., Vicedo, J.L, Bisbal, E., Moreno, L.: Experiments with LSA for Passage Re-Ranking in Quesion Answering. In: Workshop of Cross-Language Evaluation Forum (CLEF) (September 2006)
22. Vossen, P.: Introduction to eurowordnet. Computers and the Humanities 32, 73–89 (1998)
23. Whittaker, E.W.D., Novak, J.R., Chatain, P., Dixon, P.R., Heie, M.H., Furui, S.: CLEF2005 Question Answering Experiments at Tokyo Institute of Technology. In: Workshop of Cross-Language Evaluation Forum (CLEF) (September 2006)

Zero Anaphora Resolution in Chinese and Its Application in Chinese-English Machine Translation

Jing Peng and Kenji Araki

Language Media Laboratory, Hokkaido University
Kita-14, Nishi-9, Kita-ku, Sapporo, Japan
{pj,araki}@media.eng.hokuai.ac.jp

Abstract. In this paper, we propose a learning classifier based on maximum entropy (ME) for resolving ZA in Chinese. Besides regular grammatical, lexical, positional and semantic features, we develop two innovative Web-based features for extracting additional semantic information of ZA from the Web. Our study shows the Web as a knowledge source can be incorporated effectively in the learning framework and significantly improves its performance. In the application of ZA resolution in MT, it is viewed as a pre-processing module that is detachable and MT-independent. The experiment results demonstrate a significant improvement on BLEU/NIST scores after the ZA resolution is employed.

Keywords: zero anaphora resolution, Web-based features, ME-based classifier, machine translation.

1 Introduction

In many natural languages, grammatical components that can be understood contextually by a reader are frequently unexpressed for discourse fluency. This is especially the case in many Asian languages [1], such as Chinese, Japanese and Korean, where a kind of anaphoric expression is frequently eliminated. This phenomenon is called zero-anaphora (ZA) in Natural Language Processing (NLP) [2].

For example, in sentence (1)[1], ϕ_1 and ϕ_2 in the subject position refer to the subject “jie jie (my sister)” and the object “ping guo (apples) in their previous sentences respectively, and they can be recovered as “ta (she)” and “ta men (they)”. ϕ_3 in sentence (2) refers to the object of the preposition phrase, “xie zi (the matting)”, and it can be recovered as “ta (it)”. In sentence (3), the grammatical role of ϕ_4 is the object. It refers to “wen ti (the problem)”.

[1] In the examples, we use ϕ_i to denote a ZA, where the subscript i is the index of its antecedent. The noun phrase attached with the same subscript is the antecedent of the ZA. Below the examples, we also give English translation, in which the unexpressed elements are recovered and identified by ( ).

Z. Kedad et al.(Eds.): NLDB 2007, LNCS 4592, pp. 364–375, 2007.

(1) [jie jie]$_1$ lai kan wo, ϕ_1 gei le wo ji ge [ping guo]$_2$, ϕ_2 hen hao chi.
My sister visited me. (She) gave me several apples. (They) were delicious.

(2) wo jiu shui zai yi jiao de [xie zi]$_3$ shang, ϕ_3 yi jing kao de na me re.
I was sleeping on a corner of the matting. (It) was heated up very hot.

(3) [zhe ge wen ti]$_4$ kan shang qu hen rong yi, wo men hai mei you jie jue ϕ_4.
The problem seems to be very easy, but we have not resolved (it) yet.

ZA resolution is important in many natural language processing tasks. One example is machine translation (MT). For target languages such as English that cannot be adequately generated with omitted expressions, here the subject and object, the antecedent of the ZA in the source language must be identified and made explicit.

Observe the translation outputs for sentence (1) produced by some existing Chinese-English MT systems.

[Reference translation]: My elder sister visited me. (She) gave me several apples. (They) were delicious.

[MT1]: The elder sister comes to see me, gives me several apples, is delicious.

[MT2]: Sister come to see me, gave me a few apples, good to eat.

[MT3]: The elder sister comes to see me, giving me a few apples, delicious.

From these translation outputs, it is obvious that the three MT systems cannot deal with ZA resolution. That is they cannot perform to detect ZAs, identify their antecedents and generate the grammatical English in the further. Besides the systems above, the majority of current MT systems only handle one-sentence inputs, rather than full discourses. Thus they usually cannot solve ZA problem, which should be operated beyond the sentence level [3]. Our study is motivated by improving the quality of MT by resolving ZA problem.

In this paper, we assume that the task of ZA detection has been performed by some other module, such as a shallow parser [4]. Our work only focuses on identifying the antecedent of ZA. We construct a maximum entropy (ME) classifier to check whether a candidate is the correct antecedent or not. In training and applying the classifier, we first employed a set of 13 regular features to capture the context information in discourses. From the original experiment results, we found that the classifier using the regular features mainly suffered from the problem of insufficient semantic knowledge. Therefore, we improved the classifier by developing two Web-based features to obtain additional semantic information from the Web. The values of the two features can be estimated by querying the Web. We combine the two features with the regular feature set and retrain the advanced ME-based classifier. In this way, we effectively incorporate the Web into our classifier and experimentally show that the Web as a knowledge source can improve the performance of our approach. In the application of our ZA resolution in MT, we propose a frame combining our resolution and MT using a detachable and MT-independent pre-processing module. BLEU/NIST measures are used to evaluate the performance of three Chinese-English MT systems without and with the ZA resolution respectively. The experiment results show that the ZA resolution can improve the quality of MT significantly.

2 Previous Research About ZA Resolution

For identifying anaphoric relations, the existing research can be categorized into 3 classes: rule-based, statistical or machine learning models, and hybrid approaches.

In the first class, most approaches have been using heuristic rules to eliminate unacceptable candidates until the most likely antecedent is obtained. These hand-crafted rules were developed based on pragmatics and a set of linguistic factors, which include gender, number agreement, semantic consistency, syntactic parallelism, salience, proximity etc [4] [5] [6]. However, writing a complete rule set is very labor-intensive and time-consuming work, and also requires the linguistic expertise. Furthermore, if we want to take more factors and knowledge into account, it is difficult to maintain the complicated heuristic rules. In contrast to the labor intensive rule-based methods, the second class employs statistical models [7] and machine learning techniques [8]. The methods in this class rely heavily upon the availability of large corpora annotated with anaphoric relations. However, collecting such a training corpus also requires a lot of work. In order to alleviate the problem that a lot of manual work is required, [9] proposed a hybrid approach, which combined heuristic ranking rules and machine learning system, SVM. Their results showed that the combination made Japanese zero pronoun resolution more reliable. A similar hybrid architecture has been employed in [10], which combines a rule-based pre-filtering component with a memory-based resolution module for anaphora resolution in German.

Although the approaches discussed above have made great contributions to ZA resolution, most of them still have shortcomings. First, test data in most experiments were selected from one single genre, such as newspaper articles or technical manuals. The problem of whether the proposed approaches are also effective on other genres is unexplored. Semantic knowledge like common sense, is always ignored in resolution because such kind of knowledge is hard to be captured from contexts. Finally, for Chinese texts, no attempt has been made to employ a machine learning framework.

3 Our Approach for ZA Resolution

3.1 ME-Based Classifier

Maximum Entropy (ME) is a general technique for estimating probability distributions from data. The general idea of ME is that the best probability model for the data is the one that maximizes entropy over the set of probability distributions that are consistent with events [11]. In the ME framework, events are represented as multiples of weighted features. In training, the weights of features are derived on the basis of the distribution of the features in the training data, using an iterative algorithm such as generalized iterative scaling (GIS) or improved iterative scaling (IIS) [12]. In employing the model, events of a given context are evaluated by summing the weights of their respective features and normalizing over the context to obtain a probability distribution, as in Eq. 1.

$$p(y|x) = \frac{1}{Z(x)} exp\left[\sum_i \lambda_i f_i(x,y)\right] \tag{1}$$

where $p(y|x)$ represents the probability of event y given context x, and λ_i represents the weight for feature f_i. $Z(x)$ is simply a normalizing factor to ensure a proper probability

We develop a classifier based on ME to determine whether a candidate is the correct antecedent or not. Given a candidate $x \in \{$all possible candidates for a certain ZA$\}$, we need to predict $y \in \{$yes, no$\}$, which classifies x as the correct antecedent or not. The classification process starts from the noun phrase (NP) immediately preceding the ZA and proceeds backwards in the reverse order of the discourse until an antecedent is found.

The primary work of modeling an ME-based classifier is the selection of a set of features (f_i) to be considered in the analysis of the training data and the estimation of the appropriate weight (λ_i) for each feature. In the latter work, we use IIS to find the globally optimal weights given the training data. The features devised for ME modeling are described in Sect. 4.

3.2 Experimental Data

Instead of considering the ZAs of only one genre, that is, newspaper articles [3,6], we have used a diverse text corpus, which is a collection of 85 text files from different genres, about 30 million words in all, consisting of newspaper articles (current affairs, politics, economy, sports and entertainment), Chinese fiction written by different authors and textbooks from elementary school to graduate school level. We randomly selected two subsets of the corpus as our training and test data. The data excludes three types of ZA: intra-sentential, exophora and cataphora. The training data consists of 85 fragments, one fragment from each text file, with a total of 19,293 Chinese words. It includes 627 ZAs in all. There are 3,027 words and 121 ZAs in the test data.

Raw text was input and automatically preprocessed by a pipeline of NLP modules, consisting of sentence segmentation, POS tagging, NE recognition, shallow parsing, nested NP extraction and semantic class determination. All possible annotations in a training text are determined by the outputs of these modules. Our POS tagging and NE recognition were done on the basis of the Chinese lexical analysis system ICTCLAS[2]. ICTCLAS is an approach based on multilayer HMMs and it includes word segmentation, POS tagging and NE recognition. Our shallow parser parses a sequence of words into smaller constituents such as NP and verb phrases (VP) with phrase-level parsing. The module of nested NP extraction divides nested NPs into two groups. They are nested NPs that are modifier nouns. For example, the nested NP for “shi yan jie guo (experimental result)” is “shi yan (experiment)”, and nested NPs that modify the heads using the auxiliary word “de (of)”. For example, for “er tong de jiao yu (children’s education)”, the nested NP is “er tong (children)”. Semantic classes of phrases are

[2] http://sewm.pku.edu.cn/QA/reference/ICTCLAS/Free ICTCLAS

determined based on the Semantic Knowledge-base of Contemporary Chinese (SKCC) [13]. We defined 9 semantic classes for NPs and 7 classes for VPs.

Each ZA was extracted and considered in a 4-sentence context. The correct antecedents were manually annotated. In the training data, each pair of a ZA and its antecedent is considered to be a positive training instance for our classifier, while all the other NPs occurring in the scope between the ZA and it antecedent form negative training instances with the ZA. These NPs consist of general NPs, nested NPs and NEs. This procedure produced a set of 3,261 training instances, in which 627 (19.2%) were positive and 2,634 (80.8%) were negative. In the test data, all the NPs extracted in the context for a certain ZA are regarded as candidates and are used in the later classification experiments.

4 ZA Resolution with the Regular Features and Web-Based Features

4.1 Regular Features

In this section, we describe the ZA resolution using a set of regular features. The values of the features can be obtained directly from the corpus, from the pipeline of NLP modules, e.g., our POS tagger and shallow parser and using semantic dictionaries with the help of heuristic rules. The 13 regular features, RF, includes 4 grammatical, 5 lexical, 1 positional and 3 semantic features (Table 1):

Grammatical features. PREDI-ZA and PREDI-A are the surface forms of the heads of the predicates in the sentences containing the ZA and the antecedent candidate. The values come from the POS tagger and shallow parser. The feature NESTED-A checks whether the antecedent candidate is a nested NP or not. The feature GRAM-FUNC shows the grammatical role of an antecedent candidate and the values are obtained using our shallow parser with some heuristics.

Lexical features. The feature P-P describes the lexical relation between the heads of the predicates in the sentences containing the ZA and the candidate. The four possible values can be obtained based on a synonym/antonym dictionary. The feature DEMON-A checks whether the candidate starts with a demonstrative pronoun or not. PRON-A and NE-A judge whether the candidate is a pronoun or a NE. The feature CONJ evaluates the type of conjunction between the ZA and the candidate. The values can be determined directly by the conjunctions occurring in the sentences.

Positional feature. The feature DIST measures the distance between the antecedent candidate and the ZA in numbers of simple sentences.

Semantic features. The features SEM-A, SEM-P and SEM-P-A describe the semantic classes of the antecedent candidate and the heads of the predicates in the sentences containing the ZA and the candidate, respectively. All the values of these three features can be determined using a Chinese semantic dictionary.

Table 1. Summary of regular features (RFs)

Type	No.	Feature	Description	Values
Grammatical	1	PREDI-ZA	Surface form of the head of the predicate in ZA-S.	VP, AP, PP, MP, NP, unknown
	2	PREDI-A	Surface form of the head of the predicate in A-S.	VP, AP, PP, MP, NP,unknown
	[redacted]	NESTED-A	The candidate is a nested NP.	yes, no
	4	GRAM-FUNC	The grammatical role of the candidate.	subject, object, embedded subject, embedded object, attributive, adverbial modifier, others
Lexical	5	P-P	Lexical relation of the heads of the predicates in ZS-S and A-S.	same, synonym, antonym, others
	6	DEMON-A	The candidate starts with a demonstrative pronoun.	yes, no
	7	PRON-A	The candidate is a pronoun.	yes, no
	8	NE-A	The candidate is a NE.	yes, no
	9	CONJ	Type of conjunction between the candidate and the ZA.	coordinate, continued, causality, transitional, others
Positional	10	DIST	Distance between the candidate and ZA in sentences.	1, 2, 3, 4
Semantic	11	SEM-A	Semantic class of the candidate person.	animal, plant, object, artifact, space, time, data, process ...
	12	SEM-P	Semantic class of the head of the predicate in ZA-S.	state, emotion, change, weather, perception, communication ...
	13	SEM-P-A	Semantic class of the head of the predicate in A-S.	state, emotion, change, weather, perception, communication ...

ZA-S: the sentence containing the ZA; A-S: the sentence containing the antecedent candidate
VP: verb phrase; AP: adjective phrase; PP: prepositional phrase; MP: numeral-classifier phrase

4.2 Experiment Results of the Classifier Trained on the Regular Features

In order to show the improvement of the performance of our approach, we developed the following simple rule-based baseline algorithm.

Baseline rule:

For each ZA in a sentence, choose the NP in the previous sentence having the shortest distance from the ZA as its antecedent.

Table 2 shows the results for the ME-based classifier using RF in comparison with the baseline on the test data (121 ZAs). The performance is evaluated as to *accuracy*, that is, the number of correctly resolved ZAs divided by the total number of ZAs.

Our approach performs much better than the baseline. While these results are encouraging, there were several classification errors. Except the errors caused by the modules such as POS and NE recognition, the largest type of errors is due to the lack of semantic knowledge (47.1%). The examples having this type of errors can roughly be classified into two groups:

Table 2. Results with RFs

Feature	Baseline	RF
Correct number	56	87
Accuracy	46.3%	71.9%

1. Examples having word senses that are missing from the semantic dictionaries, e.g., “second quarter” should be assigned the semantic class “data” instead of “unknown” in example (4) below.
2. Examples that require general world knowledge beyond simple word meanings, e.g., examples (5) and (6) below. In (5), ϕ_6 and ϕ_7 can be interpreted correctly using the knowledge that bananas, being a fruit, are likely to be unripe and the monkeys, being an animal, are likely to be hungry. In (6), ϕ_8 refers to “Li Si” because “Zhang san” no longer has the apple. The general world knowledge on “give” is that the actor no longer has the object after giving it to others.

(4) mei nian de [di er ji du]$_5$ shou ru dou hui jian shao, ϕ_5 yi zhi shi PC jie zui cha de xiao shou ji jie.
Revenue decreases in the second quarter of each year. (The second quarter) is always the slowest quarter of the year for the PC industry.

(5) ta tao chu zhun bei hao de [xiang jiao]$_6$, sui ran ϕ_6 mei shu, ta gei le [hou zi]$_7$, yin wei ϕ_7 shi zai shi e le.
He took out the prepared bananas. Although (they) were not ripe, he gave them to monkeys because (the monkeys) were very hungry.

(6) Zhang San gei [Li Si]$_8$ yi ge ping guo, ϕ_8 hen kuai jiu chi le.
Zhang San gave Li Si an apple. (Li Si) ate the apple.

From the error analysis above, it is obvious that our approach mainly suffers from semantic problems, such as semantic ambiguity and the lack of semantic knowledge. To our knowledge, these problems cannot be resolved using any current semantic dictionary.

4.3 Web-Based Features

The Web has a vast amount of data and the volume is much larger than that of any normal corpus. Therefore, the Web is increasingly being used as an information resource in a wide range of NLP tasks. Lapata and Keller [14] describe how to use Web-based models for resolving eight NLP tasks and show the promising performance of those unsupervised systems.

In our work, we employ the Web as a semantic knowledge resource on the basis of two facts discovered by careful examination of the experiment examples. One fact in ZA resolution is that the semantic constraint that is satisfied by a ZA, must be satisfied by its antecedent. That means that for a certain ZA, its antecedent and predicate must have semantic consistency. For example, considering ϕ_6 and ϕ_7 in (5), the antecedent of ϕ_6 should be “bananas” because the candidate “bananas” has higher semantic consistence than “monkeys. The other fact is that although the relation between a ZA and its antecedent is implicitly expressed, they are linked by a strong semantic relation between the heads of the predicates of the sentences containing the ZA and its antecedent, denoted as V_{antec} and V_{ZA}. For example, considering ϕ_8 occurring in the subject position in (6), the head pair [give :: eat] has the *transfer* relation, which indicates two actions are conducted by different subjects, that is, the subject is supposed to change from one sentence to the next. Therefore the antecedent of ϕ_{13} is not the subject of the previous sentence, “Zhang san”, but “Li Si”. Examples of head pairs of the *transfer* relation include [show :: see], [fell :: fall], and [give :: have]. Besides the *transfer* relation, another relation, the *happens-before* relation defined between V_{antec} and V_{ZA} is considered in our work. This type of relation indicates that the two disjointed actions are conducted by the same subject. Examples of this type of relation include [marry :: divorce], [enroll :: graduate], and [get up :: dress].

From the points discussed above, it is obvious that the semantic consistency and the semantic relations between predicates could benefit ZA resolution. We compute the semantic consistency and discover the semantic relations by querying the Web with certain patterns indicative of each relation. We extract the Web knowledge by the following procedure.

1. **Construct lexico-syntactic patterns**. To compute the semantic consistency, two kinds of predicate-argument relations, subject-predicate and predicate-object, are considered in our work. The patterns are constructed in the form of “NP VP” (for subject-predicate) and “VP NP” (for predicate-object). The patterns for discovering the semantic relations between V_{candi} and V_{ZA} are manually designed and refined by examining examples in a volume of known semantic relations. For example, to detect the *happens-before* relation, the patterns used could be “xian V_{candi} zai V_{ZA} (V_{candi} and then V_{ZA})” and “V_{candi} jie zhe V_{ZA} (V_{candi} and later V_{ZA})”. To detect the *transfer* relation, the patterns could be “bei V_{candi} ran hou V_{ZA} (be V_{candi}ed and then V_{ZA})” and “bei V_{candi} jie zhe V_{ZA} (be V_{candi}ed and subsequently V_{ZA})”. In our work, we use a total of 17 patterns for extracting semantic knowledge from the Web.
2. **Instantiate patterns**. We instantiate the patterns for all antecedent candidates. For example, for ϕ_6 in (5), the pattern “NP VP” is instantiated with “the bananas are ripe” and “the monkeys are ripe”. If a candidate is a pronoun, it would be replaced by its closest nominal antecedent before instantiation. These instantiated patterns are submitted to a Web search engine (Google in our work).
3. **Acquire semantic knowledge from the Web**. To compute the semantic consistency from the Web, our rationale is that if the consistency between the ZA and a candidate is high, it is likely that the pattern instantiated by the candidate occurs frequently on the Web. We use *ConsiScore* to estimate semantic consistence using Eq. (2), where *count(P)* is the Web count of the instantiated pattern returned by Google, while *count(candi)* is the Web

count of the query formed with only the head of the current candidate. N is the number of Google pages ($N \approx 10.1 \times 10^9$). To determine the semantic relations, we adopt a method inspired by mutual information (MI) to measure the strength of association between V_{candi} and V_{ZA}, denoted $R_P(V_{candi}$, $V_{ZA})$, which is estimated using Eq. (3), where *count(*V_{candi}*)* and *count(*V_{ZA}*)* are the Web counts of the queries formed by V_{candi} and V_{ZA}. We determine that the semantic relation R_P indicated by pattern P is present between V_{candi} and V_{ZA} if $R_P(V_{candi}$, $V_{ZA})$, exceed a certain threshold θ (θ=0.3).

$$ConsiScore(candi, ZA) = \frac{count(P)}{count(candi)} \tag{2}$$

$$Rp(V_{candi}, V_{ZA}) = \frac{count(P) \times N}{count(V_{candi}) \times count(V_{ZA})} \tag{3}$$

Unlike research in which web counts are used directly [14], our method of employing the semantic knowledge from the Web is to combine the Web knowledge as features together with the 13 RFs described in Sect. 4.1. The two Web-based features with their possible values are summarized in Table 3. The feature SEM-CONSI obtains the semantic consistency computed from the Web. The values represent the rank of the consistency. For example, SEM-CONSI=1 means the current candidate has the highest semantic consistency among all the antecedent candidates. The feature SEM-RELA represents the semantic relations between the heads of the predicates of ZA and a candidate, as described in Sect. 5.1. In our current work, three values are possible: “transfer”, “happens-before” and “unknown”.

Table 3. Two Web-based features

Type	No.	Feature	Describe	Values
Web-based	14	SEM-CONSI	The rank of the semantic consistency of antecedent candidates.	1, 2, 3, 4 ...
	15	SEM-RELA	The semantic relations between the heads of the predicates in sentences containing ZA and an antecedent candidate.	transfer, happens-before, unknown

4.4 Experiment Results on the Regular Features and the Web-Based Features

We retrained and tested the advanced classifier with the RF set, and the two additional Web-based features. The results are summarized in Table 4. The rightmost column shows the performance of the advanced classifier. We obtain 35.5 and 9.9 point improvements in accuracy in comparison with the baseline and the original classifier trained on the RF set. In particular, all the examples given in this paper can be resolved by the advanced classifier.

Table 4. Results with RF and RF+Web-based features

Feature	Baseline	RF	RF+Web
Correct number	56	87	99
Accuracy	46.3%	71.9%	81.8%

Table 5. Personal pronouns table

	Singular	Plural
1st person	wo (I)	wo men (we)
2nd person	ni (you)	ni men (you)
3rd person	ta, ta, ta (he, she, it)	ta men (they)

5 Incorporating ZA Resolution into MT

5.1 The Frame of Incorporating ZA Resolution into MT

We incorporate the ZA resolution into MT using a general type of automated pre-processing module defined. The advantage of general modules is they are detachable and independent of any particular MT system. Fig. **??** shows the processing flow of applying ZA resolution in MT.

In Fig. 1, for the input of Chinese text, ZA resolution can be viewed as pre-processing of the input, in which ZAs are detected and their antecedents are identified. We replace ZAs by the pronouns denoting their antecedents based on personal pronouns table (Table 5), and then input the corrected texts into MT systems.

For example, Suppose the sentence to be translated is sentence (1) in Section 1. After ϕ_1 and ϕ_2 are recovered by “ta (she)” and “ta men (they)” respectively, the corrected sentence becomes to be (1a). (MT1), (MT2) and (MT3) are the translation outputs for sentence (1a) produced by three existing Chinese-English MT systems.

(1a)[jie jie]$_1$ lai kan wo, (ta) gei le wo ji ge ping guo(ta men) hen hao chi.
[Reference translation]: My elder sister visited me. (She) gave me several apples. (They) were delicious.

Table 6. The experiment results of MT evaluation

MT	BLEU (4-gram)		NIST (4-gram)	
	without ZA Resolution	with ZA Resolution	without ZA Resolution	with ZA Resolution
Babel Fish	0.462	0.619	5.198	6.807
Google	0.310	0.413	4.201	5.635
Kingsoft	0.509	0.622	5.704	7.113

[MT1]: The elder sister comes to see me, she gives me several apples, they are delicious.
[MT2]: Sister come to see me, she gave me a few apples, they are good to eat.
[MT3]: The elder sister comes to see me, she gives me a few apples, they are delicious.

5.2 The Performance of MT with ZA Resolution

We collect 200 pairs of aligned Chinese-English bilingual sentences from Chinese Linguistic Data consortium as the test data. There is at least one ZA in each of the Chinese sentences. The English sentences are viewed as reference translations in later evaluation experiments. Three existing Chinese-English MT systems are investigated in our evaluation. They are Babel Fish MT[3] , Google MT[4] that are free translation services online, and Kingsoft MT[5], which is one of the most famous translation software used in China.

The automated evaluation of MT is carried through by using reference translations for matching the translated texts. There are the N-gram co-occurrence evaluations that analyze the agreement of terms and their sequences (N-gram) between the evaluated and reference translations. In our experiments, we use BLEU and NIST as the evaluation metrics. BLEU averages the precision for unigram, bigram and up to 4-grams and incorporates a size penalty [15]. The NIST metric is derived from the BLEU evaluation criterion but differs in one fundamental aspect. That is instead of n-gram precision the information gain from each n-gram is taken into account.

The BLEU and NIST scores of the 3 MT systems without and with ZA resolution are summarized in Table 6. For Babel Fish MT, BLEU and NIST scores increase from 0.462 to 0.619 and from 5.198 to 6.807. For Google MT, the scores increase from 0.310 to 0.413 and from 4.201 to 5.635. And for Kingsoft, they increase from 0.509 to 0.622 and from 5.704 to 7.113. A significant improvement on evaluation scores after the ZA resolution incorporation into the 3 MT systems is demonstrated.

6 Conclusion

In this paper, we presented a machine learning approach based on ME for ZA resolution in Chinese. Besides commonly used features, we developed two innovative Web-based features for additional semantic knowledge from the Web. Our approach shows that the semantic knowledge obtained from the Web can be effectively introduced into the machine learning framework and significantly improve the performance of the ZA resolution. In the application in MT, ZA resolution is viewed as a pre-processing module to correct the Chinese texts that have ZAs. The BLEU/NIST scores demonstrate a significant improvement after the ZA resolution incorporation in different kinds of MT systems.

[3] Babel Fish Translation: http://babelfish.altavista.com/
[4] Google Translation: http://translate.google.com/
[5] Kingsoft: http://www.kingsoft.net

References

1. Feng, Z.W.: New Review of Machine Translation. Chinese Publishing Company (1994)
2. Li, C., Thompson, S.: Mandarin Chinese - A Functional Reference Grammar. University of California Press (1981)
3. Resnik, P., Smith, N.A.: The web as a parallel corpus. Computational Linguistics **127** (2003) 349–380
4. Yeh, C.L., Chen, Y.C.: Zero anaphora resolution in chinese with shallow parsing. Journal of Chinese Language and Computing (to appear)
5. Zhang, W., Zhou, C.L.: Study on meta-anaphoric resolution in chinese discourse understanding. Journal of Software **13** (2002) 732–738
6. Grosz, B.J., Joshi, A.K., Weinstein, S.: Centering: a framework for modeling the local coherence of discourse. Computational Linguistics **21** (1995) 203–225
7. Ge, N.Y., Hale, J., Eugene, C.: A statistical approach to anaphora resolution. In: Proc. 6th Workshop on Very Large Corpora, Montreal, Canada (1998) 161–170
8. Soon, W.M., Ng, H.T., Lim, C.Y.: Machine learning approach to coreference resolution of noun phrases. Computational Linguistics **127** (2001) 521–544
9. Isozaki, H., Hirao, T.: Japanese zero pronoun resolution based on ranking rules and machine learning. In: Proc. the 2003 Conf. on Empirical Methods in NLP (EMNLP), Sapporo, Japan (2003) 184–191
10. Hinrichs, E.W., Filippova, K., Wunsch, H.: A data-driven approach to pronominal anaphora resolution for german. In: Proc. Recent Advances in Natural Language Processing (RANLP), Borovets, Bulgaria (2005) 239–245
11. Ratnaparkhi, A.: Maximum entropy models for natural language ambiguity resolution. PhD thesis, University of Pennsylvania, Philadelphia (1998)
12. Pietra, S.D., Pietra, V.D., Lafferty, J.: Inducing features of random fields. IEEE Transactions on Pattern Analysis and Machine Intelligence **19** (1977) 380–393
13. Wang, H., Yu, S., Zhan, W.: The specification of the semantic knowledge-based on contemporary chinese. Journal of Chinese Language and Computing **113** (2003) 159–176
14. Lapata, M., Keller, F.: Web-based models for natural language processing. ACM Transactions on Speech and Language Processing (TSLP) **2** (2005) 1–31
15. Papineni, K., Roukos, S., Zhu, T.: Bleu: a method for automatic evaluation of machine translation. In: Proc. 40th Annual Meeting of the Association for Computational Linguistics (ACL-2002), Philadelphia, PA, US (2002) 311–318

Rule-Based Partial MT Using Enhanced Finite-State Grammars in NooJ

Tamás Váradi

Linguistics Institute, Hungarian Academy of Sciences
Benczúr u 33
H-1068 Budapest, Hungary
varadi@nytud.hu

Abstract. The paper argues for the viability and utility of partial machine translation (MT) in multilingual information systems. The notion of partial MT is modelled on partial parsing and involves a bottom-up pattern matching approach where the finite-state transducers assign translation equivalents locally. The article focuses on the linguistic underpinnings of the approach and gives illustrations of its implementation within the NooJ finite-state linguistic development system.

Keywords: multilingual information systems, finite-state language processing, NooJ system, local grammars, machine translation.

1 Introduction

The paper aims to show that undertaking a subset of the tasks involved in a fully functional MT system, while still challenging, is a more realistic goal. It is contended that high quality partial MT is not only viable in principle but can be implemented with relative ease using the integrated finite-state linguistic analysis tool, NooJ [7] [4].

The envisaged system can be deployed with great practical benefit in a variety of cross-linguistic applications ranging from information extraction, multilingual document management, computer assisted language learning to multilingual interface to databases.

The paper is structured as follows. Section 2 provides arguments for the linguistic justification of the approach adopted. Section 3 introduces the functionalities of the NooJ system that allow implementation of partial machine translation with it. Section 4 shows some examples for local grammars with which a Hungarian-English partial MT system may be implemented in NooJ. Finally, section 5 gives a summary and outlines future work.

2 Background Concepts

The linguistic approach followed here is based on the use of local grammars as defined by Maurice Gross [3]. The concept of partial MT is modelled on

Z. Kedad et al.(Eds.): NLDB 2007, LNCS 4592, pp. 376–381, 2007.

the widespread use of partial parsing [1][2]. The syntactic domain we chose for partial parsing is the NP maximally extended at the sentence level. One of the central tenets of the present paper is that the automatic translation of maximally extended NP's in a sentence is a viable enterprise and it has great practical benefit in a number of multilingual applications.

The viability of the enterprise is ensured by syntactic and semantic factors. A favourable syntactic feature in many languages, including Hungarian and English, is that word order within the phrase is by and large fixed. This is true even for Hungarian, which has free constituent order at the clause level. Semantically, maximally extended NP's function as self-contained referring units and as such are preserved in translation i.e. they have corresponding translation equivalents. This does not mean that every NP has such independent semantic role in a sentence. They may be part of a complex idiom such as 'show a bit of a leg' in which case the whole expression should be treated as a lexical unit.

As regards the technology of MT, it may be described as a rule-based bottom-up pattern-matching approach which is similar in its principle, if not design, to the Metamorpho system[5][6]. The flexibility of the system design is well matched by the architecture of the NooJ linguistic development tool in which we implemented our sample rules.

The patterns range in generality as a function of the lexical constraints they contain. They are applied in a cascaded manner such that first the patterns of the narrowest scope (containing the maximal lexical specifications) are deployed first, to be followed by patterns containing increasingly more general categories and finally, as default cases, patterns of the widest scope (typically, consisting of non-terminal elements only) are used. Such cascaded deployment of local grammars is designed to ensure that the most suitable translation equivalent is found earliest. This procedure makes use of the widely used principle in inheritance relation that the more specific instantiation overrules the general cases.

The robustness of the procedure is ensured by the fact that at the end of the cascade there will always be a default pattern to apply. If it was not possible to apply any pattern previously, we apply the one that surely fires, i.e. a word-by-word lookup in the dictionary. To prevent falling back on this crude default case the system should store as many patterns of as narrow scope as possible.

3 Support for MT in Nooj

In this section we will explore how the approach to MT advocated above can be implemented in the NooJ linguistic development system. There are a number of functionalities in the NooJ system that recommend it for such an enterprise. It is an integrated system comprising of a robust lexical and a grammar component, both implemented as finite-state transducers.

The NooJ lexicon contains a typed feature system that cover morphosyntactic and semantic properties of the headwords. In a NooJ dictionary entry the lexeme is followed by its word class category and a set of features, each feature prefixed with a + sign. The value of the feature *FLX* contains reference to the paradigm which specifies the inflectional forms of the entry. Bilingual dictionaries can easily be produced by linking the monolingual dictionaries through the use of features. Entries in the source dictionary contain the target equivalent as the value of a feature whose type is the name of the language. Several target languages can be listed with the appropriate name of the language, making it possible to create multilingual dictionaries of as many languages as there are monolingual dictionaries in the system.

Note that words with several target language equivalents in the same language are listed in separate entries. Lexical lookup will initially assign both equivalents to the text. Eliminating the lexical ambiguity will have to be done at a later stage.

One of the most attractive features of the NooJ system is the ease with which finite-state grammars can be created and edited in a graphical interface. The finite-state grammars are subtantially enhanced through the use of variables, the facility to make reference in graphs to lexical features, the assignment of features to nonterminal units, feature inheritance, lexical constraints as well as the use of complex features in lexical constraints.

The facility to make indirect reference to lexical features provides the ground for accessing translation equivalents of particular lexical items. For example, if the variable `$L` is used to store the particular word captured somewhere in a local grammar, the English equivalent can be accessed through the complex variable `$L$en`. The inflected forms are accessed by sticking their code at the end preceded by the underline character, `$L$en_pl` for example, produces the target language equivalent in the plural.

However, translation equivalence obtains typically not between words but phrases, therefore it is important that phrases should be assigned features and especially the features of their heads. Feature assignment and inheritance can also be implemented with the use of complex variables in NooJ as is illustrated in the examples in the next section.

4 Sample Rules

The translation rules are applied in a series starting from the most specific, i.e. multi-word expressions where all elements are lexically specified and no choice is allowed anywhere (e.g. *German measles, English breakfast*) followed by those patterns which contain varying amount of open slots (e.g. *German/French/English-speaking students/countries/population*) down to patterns which are defined only in terms of word classes *Det A N PP*. Finally, there is a single default rule, which make no reference to structure at all. In practical terms, the strategy can be paraphrased as 'apply all the rules in decreasing order of specificity and if everything fails, resort to word-by-word translation'.

For ease of exposition, we will start with this absolute bottom-line case of word-by-word translation as it requires the least apparatus.

The finite state transducer in Figure 1. produces as output the English equivalent of an arbitrary lexical unit annotated in NooJ as (`<DIC>`) and stored in the variable

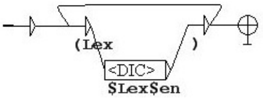


Fig. 1. Word by word translation into English

$LEX. The English translation equivalent is accessed through the complex variable $LEX$en, assuming a dictionary coding described in the previous section.

Of course, this is far too crude to serve anything but a rock-bottom default to fall back on. Typically, it is far more relevant to determine the translation equivalent with reference to the syntactic environment. The grammar fragment in Figure 2 illustrates lexical reordering as well as gender agreement in NooJ. The expressions in angled brackets are lexical constraints which make sure that the path of the graph can only be followed if a dictionary lookup confirms that the value of the word stored in variable $v is a noun of the required number and gender i.e. (m for masculine, f for feminine, s for singular and p for plural.

The central part of Figure 2 serves to illustrate how the right translation equivalent can be selected on the basis of the context. In this simple example, as the left context of the English word *state* is fully defined at the lexical level, the two word combinations might as well have been listed in the dictionary. However, one can easily conceive of cases where the context is specified by more general syntactic <N> or semantic <+bodypart> features.

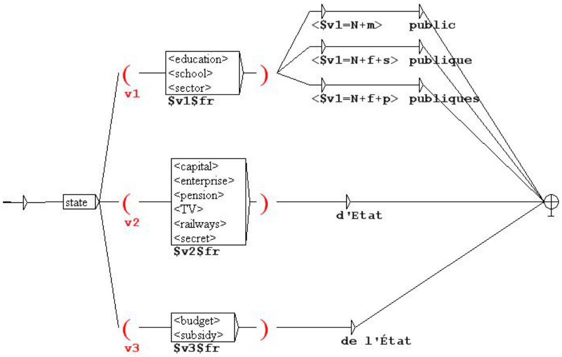


Fig. 2. Lexical and grammatical selection restrictions

The examples so far were far too general and crude, serving mostly to illustrate the facilities NooJ offers for MT applications. Where local grammars really shine are the patterns which are partly lexically defined, partly contain more or less open-class slots, which means that listing them in the dictionary is either not feasible or practical. Well-known areas are named entities (if they need translation at all, that is), dates, time and place expressions. However, it is perhaps, underestimated to what extent ordinary expressions, some of them looking quite innocuous, are governed by subtle lexical constraints which become predominant in a bilingual setting.

Table 1 merely hints at a few typical correspondences between Hungarian and English nominal constructions which appear to be readily amenable to NooJ through the use of the facilities shown in the earlier examples.

Table 1. Local structural correspondences between Hungarian and English NP's

`N with A N`	*girl with black hair*	`A N-ÃŽ N`	*fekete hajÃž lÃąny*
`A-speaking N`	*Spanish-speaking students*	`A nyelvÅś N`	*spanyol nyelvÅś diÃąkok*
`N of N`	*freedom of assembly*	`A N`	*gyÃijlekezÃľsi szabadsÃąg*
`N (Adv) Adv`	*house immediately opposite*	`A N`	*kÃűzvetlen szemkÃűzti hÃąz*
`N P N`	*people at the reception*	`A N`	*a fogadÃąson lÃľvÅŚ emberek*

Consider, as an example, the constructions in the first row of the table. Figure 3. shows the local grammar, which implements its translation from Hungarian to English. Using the facilities reviewed earlier, it is relatively easy to generate the Hungarian equivalents of the English lexical units, including the form produced with the -*ÃŽ* derivational suffix. To simplify matters, only one semantic constraint was applied that the *n0* unit should have the feature `+hum`, there was none imposed on the *n1* unit. This is correct as long as *n1* unit denotes a body part or some inalienable possession or property. In all other cases, the -*Vs* derivational suffix has to be used cf. (*man with a black umbrella*:*fekete esernyÅŚs fÃľrfi*).

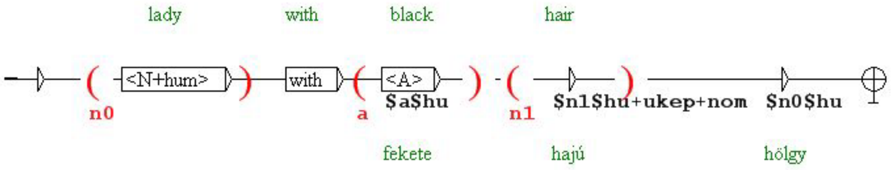


Fig. 3. Hungarian English modifier constructions

5 Summary and Further Work

The present paper intended to demonstrate that the present functionalities in the NooJ linguistic development system provide considerable support to developing complex multilingual information systems, including partial machine translation. Maximally extended NP's at the sentence level serve as appropriate domain for machine translation. Their internal structure is largely governed by local dependencies (save participle constructions or prepositional modifiers, respectively), which are amenable to finite-state technology. In their non-idiomatic, non-metaphoric uses, they represent self-contained semantic units in sentences that stand in straight correspondence with each other despite their disparate internal structure. In addition to handling the well-known areas of named entities, temporal and locative expressions, local grammars are particularly suitable to capture the intricate lexical constraints obtaining across language pairs.

Partial machine translation despite its more limited scope is of great practical use in a wide array of cross-linguistic applications ranging from information extraction, database querying, information systems to foreign language tuition.

On a slightly more ambitious note, adopting the goal of partial MT is not meant to imply that the tool and the techniques reviewed in this paper are inherently incapable of implementing a more comprehensive system. It is offered as a first stage in a research programme, a stage which is feasible and of enough practical benefit to explore and exploit fully.

References

1. Abney, S.: Partial parsing via finite-state cascades. Journal of Natural Language Engineering 2(4), 337–344 (1996)
2. Adney, S.P.: Parsing by chunks. In: Tenny, C. (ed.) The MIT Parsing Volume, 1988-89, MIT Press, Cambridge (1989), http://www.vinartus.net/spa/89d.pdf
3. Gross, M.: The construction of local grammars. In: Roche, E., Schabes, Y. (eds.) Fininte State Language Processing, pp. 329–352. MIT Press, Cambridge, MA (1997)
4. Koeva, S., Maurel, D., Silberztein, M. (eds.) Nooj pour la Linguistique et le Traitement Automatique des Langues. Presses Universitaires de Franche-Comté (2006)
5. Prószéky, G.: Machine translation and the rule-to-rule hypothesis. In: Károly, K., Fóris, Á. (eds.) New Trends in Translation Studies. In: Honour of Kinga Klaudy. Akadémiai Kiadó, Budapest (2005)
6. Prószéky, G., Tihanyi, L.: Metamorpho: A pattern-based machine translation project. In: 24th Translating and the Computer Conference, 19-24, pp. 19–24, London (2002)
7. Silberztein, M.: NooJ Manual, http://www.nooj4nlp.net/NooJ

Biomedical Named Entity Recognition: A Poor Knowledge HMM-Based Approach

Natalia Ponomareva, Ferran Pla, Antonio Molina, and Paolo Rosso

Departamento de Sistemas Informáticos y Computación,
Universidad Politécnica de Valencia, Spain,
{nponomareva,fpla,amolina,prosso}@dsic.upv.es

Abstract. With a recent quick development of a molecular biology domain it becomes indispensable to promote different resources as databases and ontologies that represent the formal knowledge of the domain. As these resources have to be permanently updated, due to a constant appearance of new data, the Information Extraction (IE) methods become very useful. Named Entity Recognition (NER), that is considered to be the easiest task of IE, still remains very challenging in molecular biology domain because of the special phenomena of biomedical entities. In this paper we present our Hidden Markov Model (HMM)-based biomedical NER system that takes into account only parts-of-speech as an additional feature, which are used both to tackle the problem of non-uniform distribution among biomedical entity classes and to provide the system with an additional information about entity boundaries. Our system, in spite of its poor knowledge, has proved to obtain better results than some of the state-of-the-art systems that employ a greater number of features.

1 Introduction

Recently the molecular biology domain has been getting a massive growth due to many discoveries that have been made during the last years and due to a great interest to know more about the origin, structure and functions of living systems. It causes to appear every year a great deal of articles where scientific groups describe their experiments and report about their achievements.

Nowadays the largest biomedical database resource is MEDLINE that contains more than 14 millions of articles of the world's biomedical journal literature. To deal with such an enormous quantity of biomedical texts different biomedical resources as databases, ontologies, search engines adapted to this domain have been created.

In fact, NER is the first step to order and structure all the existing domain information. In molecular biology it is used to identify within the text which words or phrases refer to biomedical entities, and then to classify them into relevant biology concept classes.

Although NER in biomedical domain has received attention by many researchers, the task remains very challenging and the results achieved in this

Z. Kedad et al.(Eds.): NLDB 2007, LNCS 4592, pp. 382–387, 2007.

area are much poorer than in the newswire one. Its difficulty is caused principally by the complex structure of molecular names and the lack of naming convention [1].

In this paper, we present our HMM-based biomedical Named Entity (NE) recognizer which uses only POS tags as an additional feature. We will show that POS information is very useful in biomedical NER task and that only applying this rather poor knowledge we may archieve good results and, moreover, surpass the performance of the systems exploiting a large set of features.

The paper is organized as follows. Section 2 is dedicated to illustrate some important characteristics of the Genia corpus that have been used during the construction of our model and the experiments. In Section 3, our biomedical NE recognizer is described and its comparison with the best state-of-the-art systems is made. Finally, Section 4 draws our conclusions and discusses the future work.

2 The Genia Corpus

Any supervised machine-based model depends on a corpus that has been used to train it. At the moment the largest and, therefore, the most popular biomedical annotated corpus is Genia corpus v. 3.02 which contains 2,000 abstracts from the MEDLINE collection annotated with 36 biomedical entity classes. In our experiments, we have used its JNLPBA version [2].

The JNLPBA corpus is annotated with 5 classes of biomedical entities: protein, RNA, DNA, cell type and cell line. Biomedical entities are tagged using the IOB2 notation. In Table 1 a tag distribution within the training and test corpora is shown. It can be seen that the majority of words (about 80%) does not belong to any biomedical category. Furthermore, the biomedical entities themselves also have an irregular distribution: the most frequent class (protein) contains about 10% of words approximately, whereas the most rare one (RNA) - less than 0.5%. The tag irregularity may cause a confusion among different types of entities with a tendency for any word to be referred to the most numerous class.

Table 1. Entity tag distribution in the training and test corpora

Corpus	Protein, %	DNA, %	RNA, %	cell type, %	cell line, %	no-entity, %
Training	11.2	5.1	0.5	3.1	2.3	77.8
Test	9.7	2.8	0.3	4.9	1.5	80.8

3 Preliminary Results: The HMM Approach

The HMM approach has been proved to be successfully employed in many NLP tasks, such as speech recognition, machine translation, POS tagging, NER, etc. In this section, our HMM-based NER system will be introduced together with the description of its main characteristics. Then, we will present experimental results and their comparison with some other state-of-the-art NER systems based on the same approach.

3.1 HMM-Based Biomedical NE Recognizer Description

Let $\mathbf{w} = (w_1 w_2 ... w_n)$ be a sequence of observed words of length n. Let $\mathbf{t} = (t_1 t_2 ... t_n)$ be a sequence of biomedical entity tags assigned to words from the word sequence $\mathbf{w}$. We denote as $\mathbf{T}$ a collection of various sequences of biomedical entity tags of length n. The solution of the NE recognition task using a second order HMM approach can be presented as follows:

$$\hat{\mathbf{t}} = \arg\max_{\mathbf{t}\in\mathbf{T}} [\log P(t_1) + \log P(t_2|t_1) + \sum_{i=3}^{n} \log P(t_i|t_{i-1}t_{i-2}) + \sum_{i=1}^{n} \log P(w_i|t_i)]$$

It is common to incorporate into NER systems different features which are considered to be useful for the correct classification. Our system exploits only POS feature supplied by the Genia Tagger[1]. It is significant that this tagger was trained on the Genia corpus in order to provide better results in the biomedical texts annotation. As it has been shown by [3], the use of the POS tagger adapted to the biomedical task may greatly improve the performance of the NER system than the use of the tagger trained on any general corpus as, for instance, Penn TreeBank.

In our system, the POS information serves both to provide an additional knowledge about entity boundaries and to diminish an entity class irregularity. As we have seen in Section 2, the majority of words in the corpus does not belong to any entity class. Such data irregularity can provoke errors, which are known as false negatives, and, therefore, may diminish the recall of the model. Besides, there also exists a non-uniform distribution among biomedical entity classes: e.g. class "protein" is more than 20 times larger than class "RNA" (see Table 1).

To solve the above problem we have decided to split the most numerous categories by means of POS tags of words. The idea of splitting or specializing tags was previously successfully applied to other NLP tasks, such as POS tagging, chunking or clause detection [4]. For the biomedical NER task, a similar idea was proposed by [5] who employed it for the SVM approach.

We have constructed three models using different sets of POS tags:

(1) only the non-entity class has been splitted;
(2) the non-entity class and two most numerous entity categories (protein and DNA) have been splitted;
(3) all the entity classes have been splitted.

It may be observed that each following model includes the set of entity tags of the previous one. Thus, the last model has the greatest number of states.

Besides, we have carried out various experimens with a different number of boundary tags, and we have concluded that only adding two tags (E - end of an entity and S - a single word entity) to a standard IOB2 set can notably improve the performance of the system.

Consequently, each entity tag of our models contains the following components:

[1] http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/tagger/

(i) entity class (protein, DNA, RNA, etc.);
(ii) entity boundary (B - beginning of an entity, I - inside of an entity, E - end of an entity, S - a single word entity);
(iii) POS information.

The key point, that should be paid attention to, is the POS set used in the splitting procedure. Kazama et al. [5] applied all the POS tags of the Penn TreeBank tag set. We think that the whole set of POS is rather redundant and contributes neither to the system accuracy, no to its stability.

In order to split a non-entity class, the distribution of its in-class POS tags has been analyzed (Table 2). We have realized several experiments to choose the best set of POS tags. As a result, the POS with a relative frequency of more than 1% have been selected to participate in the entity tag balancing.

Table 2. POS distribution inside of a no-entity category

POS	NN	IN	DT	JJ	<.>	NNS	<,>	CC	VBN	RB	VBD	VBZ	TO	VBP	CD	)	(	VB
%	19.6	16.3	9.6	8.6	4.8	4.7	4.6	4.2	3.8	3.1	2.7	2.4	2.0	1.8	1.7	1.6	1.5	1.4

The classes of biomedical entities have been divided according to the POS distribution within the class "Protein". In order to participate in the splitting procedure, the most frequent POS tags have been chosen (Table 3). As it may be noticed from Table 3, some parts-of-speech can appear only in certain parts of a biomedical entity (e.g. coma, brackets or conjunction never stay at the beginning of an entity).

Table 3. List of POS tags participated in the biomedical entity category splitting

POS	POS position in the entity
NN, JJ, NNS	Everywhere
(, CC, <,>	Inside
CD,)	Inside or at the end

3.2 Experiments

The first experiments we have carried out were devoted to compare our three HMM-based models in order to analyze what entity class splitting provides the best performance. In Table 4, our baseline (i.e., the model without class balancing procedure) is compared with our three models. The results seem to be promising taking into account the poor additional information we have employed. Although all our models have improved the baseline, there is a significant difference between the first model and the other two models, which have shown rather similar results.

Our system has been compared to those that are based on the same approach and used the same training and test corpora (Table 5). The system developed by Zhao et al. [6] deserves special attention because it exploits nothing else but

Table 4. Analysis of the influence of different sets of POS to the system performance

Model	Tags number	Recall, %	Precision, %	F-score
Baseline	21	63.7	60.2	61.9
Model (1)	40	68.4	61.4	64.7
Model (2)	95	69.1	62.5	65.6
Model (3)	135	69.4	62.4	65.7

a huge unlabel corpus extracted from the MEDLINE collection. The other system developed by Zhou et al. [3] achieved the best perfomance in the JNLPBA task. It exploites a large set of features and some deep knowledge resources and techniques, e.g. post-processing operations which serve to correct entity boundaries. Zhou et al. have analyzed a contribution of features, rules and external resources into the system performance and thanks to this information we can compare results of our best model with their system before applying the deep knowledge techniques.

Table 5. Comparison of biomedical NER systems based on the HMM approach

System	F-score
Zhou complete	72.6
Zhou w/o deep knowledge	64.1
Zhao	64.8
Our best model	65.7

Analyzing the results shown in Table 5, it can be appreciated that our system, which only uses in-domain POS information has obtained better results than the Zhou (w/o deep knowledge) and Zhao systems which employed many features or external resources.

We would like to remark the role of post-processing operations for the improvement of NER systems performance [7,3]. Actually, as it can be seen in Table 5, Zhou has increased the F-score on 8.5 after using deep knowledge techniques. Patrick et al. [7], who employed ME approach, also have obtained a great improvement of the F-score from 61.1 to 68.2 after applying a set of post-processing rules. All the above shows the importance of post-processing procedures for the biomedical NER task.

4 Conclusions and Future Work

In this paper, we have presented our biomedical NE recognizer. In order to tackle the problem of non-uniform distribution among biomedical entity classes, the possibility of splitting the most numerous categories by means of POS tags has been investigated. We have explored different sets of POS to realize a splitting procedure. As a result, a splitting of only the non-entity class has improved the performance of our system on about 3 points of F-score. The best result was obtained by the model, when all the entity classes were splitted (about 4 points of improvement). Despite the poor knowledge which has been used, we were able to obtain better performance than some of the state-of-the-art systems that exploited much more additional information.

As future work we plan to develop a rule-based post-processing module for our NER system. Futhermore, we will investigate different sets of features in order to find the optimal one. In fact, as it was already shown by some researchers, a rich set of features does not always help to achieve good results and could even worsen the system performance [8,9].

Acknowledgements

This work has been partially supported by MCyT TIN2006-15265-C06-04 research project.

References

1. Zhang, J., Shen, D., Zhou, G., Jian, S., Tan, C.L.: Enhancing hmm-based biomedical named entity recognition by studying special phenomena. Journal of Biomedical Informatics 37(6) (2004)
2. Kim, J.D., Ohta, T., Tsuruoka, Y., Tateisi, Y.: Introduction to the bio-entity recognition task at jnlpba. In: Proceedings of the Int. Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA 2004), pp. 70–75 (2004)
3. Zhou, G., Su, J.: Exploring deep knowledge resources in biomedical name recognition. In: Proceedings of the Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA 2004), pp. 96–99 (2004)
4. Molina, A., Pla, F.: Shallow parsing using specialized hmms. JMLR Special Issue on Machine Learning approaches to Shallow Pasing (2002)
5. Kazama, J., Makino, T., Ohta, Y., Tsujii, J.: Tuning support vector machines for biomedical named entity recognition. In: Proceedings of the Workshop on NLP in the Biomedical Domain (at ACL 2002), pp. 1–8 (2002)
6. Zhao, S.: Name entity recognition in biomedical text using a hmm model. In: Proceedings of the Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA 2004) (2004)
7. Patrick, J., Wang, Y.: Biomedical named entity recognition system. In: Proceedings of the Tenth Australasian Document Computing Symposium (ADCS 2005) (2005)
8. Settles, B.: Biomedical named entity recognition using conditional random fields and novel feature sets. In: Proceedings of the Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA 2004), pp. 104–107 (2004)
9. Collier, N., Takeuchi, K.: Comparison of character-level and part of speech features for name recognition in bio-medical texts. Journal of Biomedical Informatics 37(6), 423–425 (2004)

Unsupervised Language Independent Genetic Algorithm Approach to Trivial Dialogue Phrase Generation and Evaluation

Calkin S. Montero and Kenji Araki

Graduate School of Information Science and Technology, Hokkaido University,
Kita 14-jo Nishi 9-chome, Kita-ku, Sapporo, 060-0814 Japan
{calkin,araki}@media.eng.hokudai.ac.jp

Abstract. This paper describes an ongoing work on the generation and the automatic evaluation of trivial dialogue phrases. The described system uses a genetic algorithm (GA)-like transfer approach and n-grams frequency information obtained from the Web in order to automatically generate and evaluate phrases. The experiments present promising results, showing to achieve 78.51% of the *user understandability* of the generated and evaluated *good phrases* for English, and 79.06% for Spanish, and indicating the portability of the approach.

Keywords: GA-like transfer approach, trivial phrases, Web frequency information, language independent.

1 Introduction

The problematic domain of human-computer conversation (HCC) has been of particular interest to NLP-researchers, prompting them to develop conversational systems that range from applications to goal specific computer spoken dialogue, e.g., Jupiter, providing a telephone-based conversational interface for international weather [1], airline travel information systems [2], restaurant guides [3], telephone interfaces to emails or calendars [4], and so forth, to attempts to simulate human trivial dialogue - chat - e.g. ELIZA [5].

In this regard natural language generation as a sub-field of computational linguistics and as a core of HCC has withdrawn a good deal of research. From the *canned text* approach to the *template filling* approach the applications and limitations of language generation have been widely studied. One of the most famous examples of conversational systems, ELIZA [5], uses the template filling approach to generate the system's response to a user input, however it tends to become repetitive and monotonous, falling to sustain a coherent chat for long.

In recent research Inui et al. [6] have used a corpus based approach to language generation for dialogue system. Due to its flexibility and applicability to open domain, the corpus based approach might be considered more robust than the template filling approach when applied to dialogue systems. Other HCC systems, e.g. ALICE by Wallace [7], Jabberwacky by Carpenter [8] apply as well a corpus

Z. Kedad et al.(Eds.): NLDB 2007, LNCS 4592, pp. 388–394, 2007.

based approach to natural language generation in order to retrieve system's trivial dialogue responses to user inputs.

However, the creation of the hand crafted knowledge base, that is to say, a dialogue corpus, is a highly time consuming and hard to accomplish task[1]. In this paper we describe a work in progress for the unsupervised automatic generation and evaluation of a trivial dialogue phrases database. A trivial dialogue phrase is defined as an expression used by a chatbot program as the answer of a user input. A genetic algorithm (GA)-like transfer method is used to generating the trivial dialogue phrases for the creation of a natural language generation (NLG) knowledge base. The automatic evaluation of a generated phrase is performed by producing n-grams and retrieving their frequencies from the World Wide Web (WWW). The results obtained after applying the algorithm to Spanish and English are shown.

2 System Overview and Related Work

We apply a GA-like transfer approach to automatically generate new trivial dialogue phrases from given *seed phrases.* GA has been applied to information retrieval (IR) [9] -generating syntactic grammar rules and tag, dialogue system [10] - finding an optimal grammar which places natural language sentences input into a group of outputs clusters, and so forth, see [11] for a survey on GA/evolutionary computing together with other NLP techniques applications.

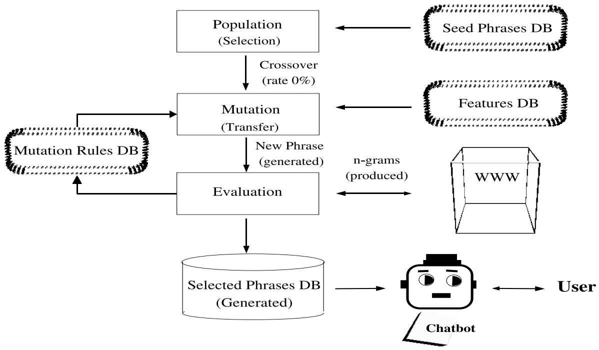


Fig. 1. System Overview

Blasband's application of GA to the automatic generation of grammar for a spoken dialogue system [10] was limited since the concepts expressed in every utterance along with their value needed to be determined manually. A transfer

[1] The creation of the ALICE chatbot database (ALICE brain) has cost more that 30 researchers, over 10 years work to accomplish. The Jabberwacky database is being developed since 1988 (on the Web since 1997) http://www.alicebot.org/superbot.html http://alicebot.org/articles/wallace/dont.html http://feeling.jabberwacky.com:8081/j2about

approach to language generation has been used by Arendse [12], where a sentence is being *re-generated* through word substitution. Problems of erroneous grammar or ambiguity are solved by referring to a lexicon and a grammar, re-generating substitutes expressions of the original sentence, and the user deciding which one of the generated expressions is correct. Our method differs in the application of a GA-like transfer process in order to automatically insert new features on the selected seed phrase, and in the application of the WWW in order to automatically evaluate the newly generated phrase, attempting to reduce the load of handwork. We assume the automatically generated database of trivial phrases is desirable as a knowledge base for open domain dialogue systems. The system general overview is shown in Fig. 1 and details of each process are given in Sec.3.

3 GA-Like Transfer Approach

3.1 Initial Population Selection

In the process of selection of the initial population a small number of phrases is chosen randomly from the Seed Phrases DB[2]. This is a small database created beforehand that contains phrases used during real human-human trivial dialogues. These are the seed phrases used during the regeneration.

For the experiment the English Seed Phrases DB contained phrases extracted from real human-human trivial dialogues obtained from the corpus of the University of South California [13] and from the hand crafted ALICE [7] database. The Spanish Seed Phrases DB contained dialogue phrases manually obtained from Spanish chat rooms. The initial population is then formed by a number of seed phrases randomly selected between one and the total number of seeds in the database. No evaluation is performed to this initial population.

3.2 Crossover

In order to ensure a language independent method our algorithm does not use syntactic information (part of speech tagging). Hence, as to avoid the distortion of the seed phrase due to completely blind crossover, in our system the crossover rate was selected to be 0%. It is found in the literature examples of GA applications to NLP were the mutation parameter has been solely the basis, e.g. [9]. In our approach the generation of the new phrase is as well given solely by the mutation process explained below.

3.3 Mutation

During the mutation process, each one of the seed phrases selected in the initial population is mutated at a rate of $1/N$, where N is the total number of words in the phrase (for Spanish we consider a word to be an *article+noun* combination when it appears). The mutation is performed through a transfer process, using

[2] In this paper DB stands for database.

the Features DB. For the experiment this database contained 100 different features for Spanish and 110 features for English (the word "features" refers here to nouns, adjectives, adverbs and verbs).

The word to be replaced within the seed phrase is randomly selected as well as it is randomly selected the feature to be used as a replacement from the Features DB. As stated previously, in order to obtain a language independent algorithm, the system does not have knowledge of part of speech. After the newly generated phrase is evaluated by the system, a mutation rule is created in the form: $A- > B : Value$, where A represents the feature substituted in the seed phrase, B represents the substitution feature (taken from the Feature DB) and *Value* is a weight automatically given to the rule during the evaluation process to determine its usability. Negative value rules are discharged.

3.4 Evaluative Criteria

In our approach we attempt to evaluate whether a generated phrase is *correct* through the frequency of appearance of its n-grams in the Web, i.e., the *fitness* as a function of the frequency of appearance. Since matching an entire phrase on the Web might result in very low retrieval, or even non retrieval at all, the sectioning of the given phrase into its respective n-grams is useful. The n-grams frequency of appearance on the Web (using Google search engine) is searched and ranked. The n-grams are evaluated according to the following algorithm:

```
if $\alpha < NgramFreq < \theta$, then $Ngram$ "weakly accepted"
elsif $NgramFreq > \theta$, then $Ngram$ "accepted"
else $Ngram$ "rejected", foreach $Ngram = bigram, trigram, quadrigram$
```

where, α and θ are thresholds that vary according to the n-gram type, and $NgramFreq$ is the frequency, or number of hits, returned by the search engine for a given n-gram. The number of n-grams created for a given phrase is automatically determined by the system and it varies according to the length of the phrase. The system evaluates a phrase as "good" if all of its n-grams were labeled "accepted", it is to say, all of the n-grams are above the given threshold. If for a given phrase there is at most *one rejected n-gram* or *one weakly accepted n-gram*, the phrase is evaluated as "usable", otherwise the phrase is "rejected".

4 Experiments and Results

The system was setup to perform 1,000 generations for each one, English and Spanish databases. In the case of English, the system generated 2,513 phrases, from which 405 were evaluated as "good", 924 were "usable" and the rest 1,184 were "rejected". In the case of Spanish, there were 1,767 different phrases generated, from which 43 were evaluated as "good", 340 were evaluated as "usable" and the rest 1,384 were "rejected" by the system.

As part of the experiment, the generated phrases were evaluated by a native English speaker and a native Spanish speaker in order to determine their "understandability". By understandability here we refer to the semantic information contained by the phrase, that is to say, how well it expresses information to the

Table 1. Human Evaluation: Understandability of the Spanish Phrases

S. Evaluation	User Evaluation: Semantics			User Evaluation: Grammar		
	5	3	1	5	3	1
Good [43]	18.60% [8]	60.46% [26]	20.93%[9]	30.23% [13]	32.56% [14]	37.21% [16]
Usable [340]	12.06% [41]	49.41% [168]	38.53% [131]	13.82% [47]	39.12% [133]	47.06% [160]
Rejected [1384]	0.66% [9]	4.12% [65]	94.63% [1310]	0.79% [11]	4.06% [56]	95.15% [1317]

S. = System *[ ] number of phrases

user. We argue that one of the characteristics of human chat is the ability to express semantic information within a given context. This implies that a given phrase does not necessarily have to be grammatically correct in order to be understood and used. A large amount of examples can be found in the chatting rooms on the Web, in English and Spanish. The human evaluation of the generated phrases was performed under the criteria of the following two categories:

a) Grammatical correctness: ranging from 1 to 5, where 1 is incorrect and 5 is completely correct.
b) Semantic correctness: evaluates the ability of the phrase to convey information, asking the question *is the meaning of the phrase understandable?* It ranges from 1 to 5, where 1 is no-understandable, and 5 is completely understandable.

The results of the evaluation are shown in Table 1 and Table 2. Table 1 shows the results for the Spanish phrases. According to the user evaluation of the semantics for the "usable phrases" evaluated by the system, 61.47% (considering the pharses evaluated with 3 to 5 points) of those phrases were semantically understood by the user. The understandability rose considerably, to 79.06%, for the "good phrases". In the case of the Spanish phrases grammar evaluation, the percent of correct grammar for the "good phrases" was around 62.79% while for the "usable phrases" ranked around 52.94%. It is seen here that in many cases (around 18% of the cases) the expresion was understood despite of its grammar.

Table 2 shows the results for the English phrases. For the "good phrases" the understandability ranked around 78.51%, and for the "usable phrases" it ranked 77.05%. On the other hand, the grammar for the "good phrases" is around 96.05% and for the "usable phrases" it is around 98.81%. It is worth noticing that in the case of English the ranking of the grammar was higher than the ranking of the semantics. This is considered to be caused by the appereance of grammatically correct patterns that may not convey semantic information, e.g., [VBP/PRP/VBD/ PP/NN], which might be *have you had your lunch?* as well as *have you had your hand?*, being the latter evaluated as grammatically correct (5 points) but semantically incorrect (no-understandable, 1 point) by the user. Therefore in the case of English, in contrast to Spanish, despite of completely correct grammar in many cases lack of meaning arose. On the other hand, the algorithm showed to successfully identify *incorrect phrases*, ranking around 94% in average for the evaluation of grammar and semantics for both English and Spanish.

According to the experiments results, grammatical correctness, however important, does not seem to have a strongly decisive impact on the understandability of

Table 2. Human Evaluation: Understandability of the English Phrases

S. Evaluation	User Evaluation: Semantics			User Evaluation: Grammar		
	5	3	1	5	3	1
Good [405]	34.07% [138]	44.44% [180]	21.48% [875]	39.75% [161]	56.30% [228]	3.95% [16]
Usable [924]	19.91% [184]	57.14% [528]	22.94% [212]	13.96% [129]	84.85% [784]	1.19% [11]
Rejected [1184]	1.17% [21]	3.80% [45]	94.43% [1118]	3.46% [41]	3.72% [44]	92.82% [1099]

S. = System *[ ]= number of phrases

a trivial dialogue phrase. Therefore, these results reinforce our argument regarding the prevalence of semantics over grammar for trivial dialogue conversation.

5 Conclusions and Future Work

In this paper the automatic generation of trivial dialogue phrases was shown through the application of a GA-like transfer approach applied to English and Spanish. The system automatic evaluation of a generated phrase using the WWW as a knowledge database was then judged by a user and a tendency towards the prevalence of understandability over completely correct grammar was observed in the resultant evaluation. As on going work, room is left to explore the application of crossover to the mutation rules, to survey other users opinions in order to identify tendencies, as well as to explore the applicability of the system to Japanese and the validity of the application of the obtained trivial phrases databases to dialogue systems.

References

1. Zue, V., Seneff, S., Glass, J., Polifroni, J., Pao, C., Hazen, T.J., Hetherington, L.: Jupiter: A telephone-based conversational interface for weather information. IEEE Transactions on Speech and Audio Processing 8, 85–96 (2000)
2. Bratt, H., Dowding, J., Hunicke-Smith, K.: The SRI telephone-based ATIS system. In: Proceedings of the ARPA Spoken Language System Technology Workshop, pp. 22–25 (1995)
3. Lau, R., Flammia, G., Pao, C., Zue, V.: Webgalaxy: Beyond point and click - a conversational interface to a browser. In: Proceedings of the 6th International WWW Conference, pp. 119–128 (1997)
4. University of Texas at Austin: Smartvoice. http://www.utexas.edu/its/smartvoice/
5. Weizenbaum, J.: Eliza a computer program for the study of natural language communication between man and machine. Commun. ACM 9(1), 36–45 (1966)
6. Inui, N., Koiso, T., Nakamura, J., Kotani, Y.: Fully corpus-based natural language dialogue system. In: Natural Language Generation in Spoken and Written Dialogue, AAAI Spring Symposium (2003)
7. Wallace, R.: A.L.I.C.E. Artificial Intelligence Foundation (2005), http://www.alicebot.org
8. Carpenter, R.: Jabberwacky. Learning Artificial Intelligence, http://www.jabberwacky.com, http://www.icogno.com/

9. Losee, R.M.: Learning syntactic rules and tags with genetic algorithms for information retrieval and filtering: An empirical basis for grammatical rules. Information Processing & Management 32(2), 185–197 (1996)
10. Blasband, M.: GAG: Genetic Algorithms for Grammars (the sex life of grammars). Technical report, Compuleer (1998)
11. Kool, A.: Literature survey (1999), http://citeseer.ist.psu.edu/kool99literature.html
12. Arendse, B.: Easyenglish: Preprocessing for MT. In: Proceedings of the Second International Workshop on Controlled Language Applications (CLAW98), pp. 30–41 (1998)
13. USC.: Dialogue Diversity Corpus (2005), http://www-rcf.usc.edu/~billmann/diversity/DDivers-site.htm

Large-Scale Knowledge Acquisition from Botanical Texts

François Role[1], Milagros Fernandez Gavilanes[2],
and Éric Villemonte de la Clergerie[3]

[1] L3i, Université de La Rochelle, France
Francois.Role@univ-lr.fr
[2] University of Vigo, Spain
mfgavilanes@uvigo.es
[3] INRIA Rocquencourt, France
Eric.De_La_Clergerie@inria.fr

Abstract. Free text botanical descriptions contained in printed floras can provide a wealth of valuable scientific information. In spite of this richness, these texts have seldom been analyzed on a large scale using NLP techniques. To fill this gap, we describe how we managed to extract a set of terminological resources by parsing a large corpus of botanical texts. The tools and techniques used are presented as well as the rationale for favoring a deep parsing approach coupled with error mining methods over a simple pattern matching approach.

1 Introduction

In this paper we are concerned with new methods for making more readily accessible the large amount of information contained in existing printed floras. Floras are books that contain descriptions of taxa (plant species, genus, etc.) occurring in a particular geographic area. A taxon description usually includes short sections relating to nomenclature, ecology, geographical distribution, and a free-text account of the plant morphology. This information is crucial to scientists in the field of botany. Traditional printed floras published before the computer age are rich documents containing a wealth of information which is still of great value but is difficult to search and exploit. In spite of this fact, not many investigations address the problem of the digitization of these legacy texts. Among the few projects that resemble our endeavor is the digitization of Flora Zambeziaca conducted at the herbarium of the Royal Botanic Garden Kew. Flora Zambeziaca is a large African flora covering about 8500 species and available as a set of printed volumes published from 1960 onwards [1]. Other similar projects include Flora of Australia online and Flora of North America. All these systems use the information contained in the floras only to implement online databases that can be searched by traditional access points (scientific name, synonyms, geographical location, etc.) However, the morphological description sections are left unanalyzed and can only be searched using basic full-text techniques.

Z. Kedad et al.(Eds.): NLDB 2007, LNCS 4592, pp. 395–400, 2007.

In contrast to these approaches, we propose to mine these textual portions in order to extract terminological resources using statistical and NLP techniques. These terminological resources will be used to help users make more precise queries against digitized botanical texts. They could also be used to serve as a starting point for the creation of domain ontologies, a future goal of our research.

This research was conducted as part of the "Biotim" project [2] on processing flora. Work concentrated on the *Flore du Cameroun* [**FdC**], an African flora comprising 40 volumes, each volume running to about 300 hundred pages. Although published between 1963 and 2001, the **FdC** exhibits a relatively regular structure. Each taxon description comprises a set of paragraphs. Most of these paragraphs (sections relating to author and collector, bibliography, ecology, distribution, etc.) are short, stereotyped fields relatively easy to recognize and analyze using pattern matching. One exception, however, is the description section which provides a detailed free-text account of the main morphological features of a taxon. This section is a (possibly) long text, which consists of several sentences with complex syntactic structures. Analyzing such textual content is a task that requires the use of linguistic resources and tools. However, it is a necessary step to fully exploit the informational richness of the Flora.

In section 2 we describe how we were able to derive the logical structure from the digitized text. Section 3 discusses the various approaches we developed to better exploit the complex content found in the free-text sections of taxonomic descriptions. Finally, we present a comparison with related work and conclude.

2 Capturing the Logical Structure

Starting from the digitized printed pages, the logical structure of the 37 volumes of **FdC** was retrieved using Perl regular expressions. The different sections of the plant descriptions (author, name, bibliography, type, distribution, ecology, material, morphological description) were recognized and marked-up in XML.

The documents have been stored in an online repository. They can be browsed via an interface where XML elements can be expanded and collapsed by clicking on icons. We also experimented with shredding and storing the XML documents into the columns of an object-relational database which could then be searched using XML query languages such as XQuery. Another benefit of capturing the logical structure is that we are then in a position to easily derive a semantic web-compatible representation of hierarchical relationships between taxa. Once converted into an XML document, the marked-up text so obtained can be fed into a simple XSLT program which generates a hierarchy of OWL classes mirroring the taxonomic hierarchy. This can be seen as the starting point for the construction of a domain ontology.

3 Analyzing the Textual Content

Having captured the logical structure we are then able to identify and target for analysis the free text sections that describe the plants in terms of their physical features. As said in the introduction, these free-text descriptions are often poorly exploited. They usually contain a separate sentence for each major plant organ. Marking up this implicit structure provides a context to identify adjectives specific to each organ. For example, adjectives like "*ligulée-lancéolée*" (ligulate-lanceolate) or "*bilobée*" (bilobate) are suitable for describing a leaf while "*multiflore*" (multiflora) is appropriate for describing an inflorescence.

Having available a list of the adjectives that are used to describe specific parts of a plant may help users formulate more precise queries against the free-text description section. In order to create this terminology we performed a morpho-syntactic analysis of the sentences in the description section. We used the tools developed by the INRIA ATOLL team to generate morpho-syntactic annotations in an XML format compliant with the MAF proposal. The tagging tools proved very reliable in unambiguously detecting punctuation marks, thus allowing us to segment each description section into sentences relating to a certain organ. Using simple XPath expressions, we then searched each sentence for all adjectives or past participles that agree in gender, in case and number with the noun at the beginning of the sentence. We used this technique on three volumes dealing with the orchid family, which accounts for about 10% of the taxa described in **FdC**.

Overall, more than 400 adjectives appropriate for describing the physical features of a tropical orchid were identified and validated. Again we encoded the obtained resource in OWL as a potential starting point to develop a domain ontology derived from the flora. This shallow parsing strategy enables us to identify dependencies between nouns and adjectives that are adjacent or very near to each other within the text. However, it fails in modeling long-range dependencies. By way of an example, in the case of a sentence such as "*feuille charnue à nervure étroite*" (fleshy leaf with narrow vein) the shallow parsing strategy described above does not allow to know if "*étroite*" (narrow) relates to "*feuille*" (leaf) or to "*nervure*" (vein). In fact, this approach even fails to detect that "*nervure*" is a component of "*feuille*" and more generally is not appropriate for dealing with suborgans names that are deeply nested within the sentences. Thus the surface form of the text directly governs our ability to pinpoint occurrences of organ names, a situation which is not desirable. We also experimented with third-party tools (ACABIT, FASTR) but still failed to detect long-range dependencies. Finally, besides not being adequate for detecting long-range dependencies, shallow parsing techniques for information extraction generally rely on hand-crafted extracting patterns. Designing these patterns is a costly task which often requires the help of a domain expert and has to be redone for each new domain and often tailored for slightly different corpora in a same domain (style variations, different authors, ...).

We have considered that these issues could be solved by adapting a generic French deep parser while trying to minimize the amount of required human intervention. The tuning of the parser was facilitated by both the use of MetaGrammars and of error mining techniques. Indeed, our French grammar, named FRMG, is compiled from a source Meta-Grammar, which is modular and hierarchically organized. It is trivial to deactivate phenomena that are not present in the corpora (for instance clefted constructions, questions, imperative, . . .).

On the other hand, error mining techniques have been used to quickly spot the main sources of errors and then fix the grammar accordingly. The idea behind this is to track the words that occur more often that expected in sentences whose analysis failed. Those words may be obtained by using a fix-point algorithm and generally indicate some kind of problem, often related to lexical entries but sometimes to segmentation or grammatical issues. The algorithm is as follows. We note p_i the i-th sentence in the corpus and $O_{i,j}$ the j-th word in the i-th sentence. $F(O_{i,j}) = f$ denotes the form of occurrence $O_{i,j}$. Let $S_{i,j}$ be the probability that word occurrence $O_{i,j}$ was the reason why the analysis of a given sentence p_i failed. We first estimate $S_{i,j}$ using the formula $S_{i,j} = \mathrm{error}(p_i)/|p_i|$ where $\mathrm{error}(p_i)$ returns 1 if the analysis failed and 0 else. It means that, for a sentence whose analysis failed, we first assume that all its word occurrences have the same probability to be the cause of the failure. We then use these (local) estimations of $S_{i,j}$ to compute $S_f = \frac{1}{|O_f|} \cdot \sum_{O_{i,j} \in O_f} S_{i,j}$ where $O_f = \{O_{i,j} | F(O_{i,j}) = f\}$. It is an estimation of the average failure rate of the form $f = F(O_{i,j})$. We then use this (global) estimation of S_f to refine the initial estimation of $S_{i,j}$ and so on until convergence. Considering the sentences which receive at least one full parse, FRMG usually achieves a coverage around 40 to 50% percent on general corpora. Without adaptation, FRMG only attained an initial 36% coverage on **FdC**. Nevertheless, by parsing the whole corpus (around 80 000 sentences depending on sentence filtering) 14 times and exploiting the feedback provided at each round using the error mining techniques, we were eventually able to improve the overall performance of the parsing from 36% to 67%. Each round took around a night processing on a local cluster of 6-7 PCs.

Once the grammar and vocabulary have been adequately tailored, our parser returns a shared forest of dependencies for each successfully parsed sentence. A dependency is a triple relating a source governor term to a target governed one through some syntactic construction provided as a label. According to Harris distributional hypothesis that semantically related terms tend to occur in similar syntactic contexts, examining the dependency edges that enter or leave nodes in the graph should allow us to extract terms that are semantically related. The problem is that even in a successfully parsed sentence, ambiguous dependencies may remain, due to the fact that several derivations may be possible and several syntactic categories may still be in competition for assignment to a word. At first glance, it seems that some of these ambiguities can be eliminated by using linguistic markers such as range constructions "*X à Y*" [X to Y], where both X and Y are adjectives (such as in the phrase "yellow to orange"), which implies that X and Y belong to the same semantic class, or very explicit linguistic markers such as "*coloré en X*" (X-colored), "*en forme de X*" (in form of X). However the number of these markers is limited, not to mention that they may be ambiguous (for example the range construction in French in competition with prepositional attachments). Second, from the beginning of the project we have sought to develop solutions that are not too corpus specific.

We therefore came upon the idea of developing a statistical iterative algorithm inspired from our error mining techniques to converge toward a better classification. The main idea is to locate the dependencies that are the most frequent at the corpus level. As for error mining, this "global knowledge" is then reused locally (at the sentence level) to reinforce the weight of some local dependencies at the expense of competing ones. The updated local weights can then be used to re-estimate the global weights and so on. After convergence, we globally get the most probable dependencies for high-frequency terms and, therefore, the most probable syntactic contexts for a term. Preliminary results have been obtained trying to classify terms into "organs", "properties" (in general), and "others", initiating the process with only 6 seed terms describing organs : "*feuille*" (leaf), "*pétale*" (petal), "*fruit*" (fruit), "*ovaire*" (ovary), "*rameau*" (small branch), and "*arbre*" (tree) and around ten properties. Actually, a weight $w_c(t)$ is attached to each term t for each class c, and we order following $w_c(t) * ln(\#t)$ to favor high frequency terms. Indeed, we are mostly interested in getting some knowledge about the most frequent terms, and, moreover, the results are statistically less significant for low frequency terms. Table 1 lists the 10 best terms for each category. The results seem satisfactory, but for two entries that reflect segmentation issues. It is worth observing that some of the best terms for the "other" category are good candidates to be property introducers ("*forme*"/shape, "*couleur*"/color, "*taille*"/size, "*hauteur*"/height, ...).

Table 1. The best-ranked terms found by our method

organs	**properties**	**other**
nervure (vein)	oblong (oblong)	diamètre (diameter)
fleur (flower)	ovale (oval)	longueur (length)
face (face)	ovoïde (ovoid)	hauteur (height)
feuille (leaf)	elliptique (elliptical)	largeur (width)
limbe (limb)	glabre (hairless)	taille (size)
rameau (small branch)	lancéolé (lanceolate)	forme (form)
sommet (apex)	ellipsoïde (ellipsoid)	forêt (forest)
sépale (sepal)	globuleux (globular)	* d
foliole (foliola)	floral (floral)	couleur (color)
base (base)	aigu (acute)	* mètre (meter)

Overall, the top 100 terms in each category seem to be correct, an intuition that we hope to confirm in the following way. In the first step, the automatically extracted list of terms will be compared to existing terminological resources, such as the thesaurus of the French Society of Orchidophily recently put at our disposal. Terms not found in the thesaurus will be marked as such and sent to experts from two national herbaria (Cameroun and Senegal) for advice and validation, and standard statistical methods such as kappa will be used to quantify inter-rater agreement.

4 Comparison with Related Work

Most systems designed to derive clusters of related words mostly use shallow parsing [3,4,5,6]. Those relying on deep parsers usually adopt a sequential approach. First a sentence is parsed using a trained parser and then the head of each constituent of the sentence is identified using patterns [7]. As a variant, [8] uses a parser that directly generates dependency trees. In both cases it is assumed that the parsers have been accurately trained and can generate high accuracy parses, thus providing a sound basis for the dependency extraction.

In contrast, we accept that the parsing step returns errors, and we rely on statistical methods to progressively tune a generic untrained parser. This work also provided the opportunity to learn from non-verbal structures (noun phrases) whereas other research mainly focuses on dependencies between a verb and its arguments, to detect verbs that denote the same ontological relations. Last but not least, adapting the error mining techniques in order to exploit dependencies minimizes the need to design domain-dependent rules.

5 Conclusion

We have explored new ways for exploiting the rich scientific information found in botanical texts. To the best of our knowledge it is the first time that advanced linguistic tools have been applied to analyze the text descriptions found in printed floras. A combination of NLP and statistical tools allowed us to produce terminological resources. These resources can now be utilized for several purposes ranging from helping users make more precise queries against botanical databases to producing domain ontologies in the field of botany.

References

1. Kirkup, D., Malcolm, P., Christian, G., Paton, A.: Towards a digital african flora. Taxon 54(2), 457–466 (2005)
2. Rousse, G., de La Clergerie, É.V.: Analyse automatique de documents botaniques: le projet Biotim. In: Proc. of TIA'05, Rouen, France, pp. 95–104 (April 2005)
3. Daille, B.: Terminology mining. In: Pazienza, M.T. (ed.) Information Extraction in the Web Era. Lectures Notes in Artifial Intelligence, pp. 29–44. Springer, Heidelberg (2003)
4. Faure, D., Nédellec, C.: ASIUM: learning subcategorization frames and restrictions of selection. In: Proc. of the 10th Conference on Machine Learning (ECML 98) Workshop on Text Mining (1998)
5. Grefenstette, G.: Explorations in Automatic Thesaurus Construction. Kluwer Academic Publishers, Dordrecht (1994)
6. Cimiano, P., Staab, S., Hotho, A.: Clustering ontologies from text. In: Proceedings of LREC'04, pp. 1721–1724 (2004)
7. de Marneffe, M.-C., MacCartney, B., Manning, C.D.: Generating typed dependency parses from phrase structure parses. In: Proc. of LREC'06 (2006)
8. Lin, D., Pantel, P.: DIRT - discovery of inference rules from text. In: Proceedings of KDD-01, San Francisco, CA, pp. 323–328 (2001)

Lexical-Based Alignment for Reconstruction of Structure in Parallel Texts*

Alexander Gelbukh[1], Grigori Sidorov[1], and Liliana Chanona-Hernandez[2]

[1] Center for Research in Computer Science, National Polytechnic Institute,
Av. Juan Dios Batiz, s/n, Zacatenco, 07738, Mexico City, Mexico
www.Gelbukh.com, sidorov@cic.ipn.mx
[2] Faculty of Electric and Mechanical Engineering,
National Polytechnic Institute,
Mexico City, Mexico

Abstract. In this paper, we present an optimization algorithm for finding the best text alignment based on the lexical similarity and the results of its evaluation as compared with baseline methods (Gale and Church, relative position). For evaluation, we use fiction texts that represent non-trivial cases of alignment. Also, we present a new method for evaluation of the algorithms of parallel texts alignment, which consists in restoration of the structure of the text in one of the languages using the units of the lower level and the available structure of the text in the other language. For example, in case of paragraph level alignment, the sentences are used to constitute the restored paragraphs. The advantage of this method is that it does not depend on corpus data.

1 Introduction

For a text in two different languages, the parallel text alignment task consists in deciding which element of one text is translation of which one of the other text. Various researchers have tried different approaches to text alignment, usually at sentence level [5], and a number of alignment tools are available. Some methods rely on lexical similarity between two texts [3]. In our previous paper [2], we have suggested an alignment method based on measuring similarity using bilingual dictionaries and presented an approximate heuristic greedy alignment algorithm. We evaluated it on fiction texts that represent difficult cases for alignment. In this paper, our goals are to introduce an optimization algorithm that finds the best solution, instead of the approximate heuristic-based algorithm, using the same measure of lexical similarity as well, and to propose an alternative method of evaluation of alignment algorithms based on reconstruction of the global text structure in one of the languages.

* Work done under partial support of Mexican Government (CONACyT, SNI) and National Polytechnic Institute, Mexico (SIP, COFAA).

Z. Kedad et al.(Eds.): NLDB 2007, LNCS 4592, pp. 401–406, 2007.

2 Similarity Measures

For assigning weight to a possible correspondence, we need to calculate the similarity between two sets of paragraphs. We define this function as similarity between two texts that are obtained by concatenation of the corresponding paragraphs.

The first baseline method is relative position of the paragraphs. Common sense suggests that the corresponding pieces of texts are located at approximately the relative same distance from the beginning of the whole text. We define the baseline distance between two pieces of text, T_A in the language *A* and T_B in the language *B*, as follows:

$$\text{Distance}(\mathrm{T}_A, \mathrm{T}_B) = |\text{start}(T_A) - \text{start}(T_B)| + |\text{end}(T_A) - \text{end}(T_B)|, \tag{1}$$

where start(T_X) is the relative position of the first word of the text T_X measured in percentage of the total number of words in the text in the corresponding language, and similarly for end(T_X). We could also use the position of the paragraph instead of word as percentage of the total number of paragraphs, but the measure based on word counts has been reported as better than the one based on paragraph counts, which agrees with our own observations.

We also used the well-known algorithm by Gale and Church [1] as another baseline for comparison.

As far as lexical similarity is concerned, we define the similarity between two texts in different languages as the number of words in both texts that are not mutual translations of each other [5]. Note that it is more correct to call this penalization; we use the term “similarity” just for the sake of uniformity with other approaches. The greater is this value, the less similar are the paragraphs.

For calculating this, we take into account the number of words that are such translations taken from a dictionary. Then we calculate the number of word tokens without translation in both paragraphs, under the hypothesis that these two paragraphs correspond to each other, namely:

$$\text{Distance}(T_A, T_B) = |\,\mathrm{T}_A\,| + |\,T_B\,| - 2 \times \text{translations}. \tag{2}$$

The cost of an alignment hypothesis is the total number of words in both texts that are left without translation under this hypothesis. Note that under different hypotheses this number is different: here we consider two word tokens to be translations of each other if both of the following conditions hold: (a) they are dictionary translations (as word types) and (b) the paragraphs where they occur are supposed to be aligned. Note that we perform morphological lemmatization and filter out the stop words.

3 Algorithm

To find the exact optimal alignment, we apply a dynamic programming algorithm. It uses a $(N_A + 1) \times (N_B + 1)$ chart, where N_X is the number of paragraphs in the text in the language *X*.

The algorithm works as follows. First, the chart is filled in:

```
$a_{00}$ := 0, $a_{i0}$ := $-\infty$, $a_{0j}$ := $-\infty$ for all $i, j > 0$.
for $i$ from 1 to $N_A$ do
    for $j$ from 1 to $N_B$ do
        $a_{ij}$ := min ($a_{xy}$ + Distance ($T_A [x + 1 .. i]$, $T_B [y + 1 .. j]$))
```

Here, a_{ij} is the value in the (i,j)-th cell of the chart, $T_X [a .. b]$ is the set of the paragraphs from a-th to b-th inclusive of the text in the language X, and the minimum is calculated over all cells (x,y) in the desired area to the left and above the (i,j)-th cell.

As in any dynamic programming algorithm, the value a_{ij} is the total weight of the optimal alignment of the initial i paragraphs of the text in the language A with the initial j paragraphs of the text in the language B. Specifically, upon termination of the algorithm, the bottom-right cell contains the total weight of the optimal alignment of the whole texts. The alignment itself is printed out by restoring the sequence of the assignments that led to this cell:

```
$(i,j)$ := $(N_A, N_B)$.
while $(i,j) \neq (0, 0)$ do
    $(x,y)$ := argmin ($a_{xy}$ + Similarity ($T_A [x + 1 .. i]$, $T_B [y + 1 .. j]$))
    print "paragraphs in $A$ from $x + 1$ to $i$ are aligned with
            paragraphs in $B$ from $y + 1$ to $j$."
    $(i,j)$ := $(x,y)$
```

Here, again, the minimum is sought over the available area to the left and above the current cell (i,j). Upon termination, this algorithm will print (in the reverse order) all pairs of the sets of paragraphs in the optimal alignment.

4 Experimental Results: Traditional Evaluation

We experimented with a fiction novel *Advances in genetics* by Abdón Ubídia and its original Spanish text *De la genética y sus logros*, downloaded from Internet. The English text consisted of 114 paragraphs and Spanish 107, including the title.[1] The texts were manually aligned at paragraph level to obtain the gold standard.

As often happens with literary texts, the selected text proved to be a difficult case. In one case, two paragraphs were aligned with two: the translator broke down a long Spanish paragraph 3 into two English paragraphs 4 and 5, but joined the translation of a short Spanish paragraph 4 with the English paragraph 5. In another case, the translator completely omitted the Spanish paragraph 21, and so on.

Both texts were preprocessed by lemmatizing and POS-tagging, which allowed for correct dictionary lookup. Stop-words were removed to reduce noise in comparison; leaving the stop-words in place renders our method of comparison of paragraphs completely unusable. Then our algorithm was applied, with both baseline and suggested distance measures.

We evaluate the results in terms of precision and recall of retrieving the hyperarcs (union of several units, or arcs in hypergraph that corresponds to alignment):

[1] We did not experiment with a larger corpus because we are not aware of a gold-standard manually aligned Spanish-English parallel corpus.

precision stands for the share of the pairs in the corresponding alignments; recall stands for the share of the pairs in the gold standard that are also found in the row corresponding to the method. Alternatively, we broke down each hyperarc into pair-wise correspondences, for example, 48–50=47 was broken down into 48 ~ 47, 49 ~ 47, 50 ~ 47, and calculated the precision and recall of our algorithm on retrieving such pairs; see the last two columns of Table 1.

Table 1. Comparison of the similarity measures

Measure	Hyperarcs		Single arcs	
	Precision, %	Recall, %	Precision, %	Recall, %
Proposed	89	85	88	90
Baseline	65	28	43	54
Gale-Church	89	86.5	87.5	91.5

One can see that the proposed distance measure based on the bilingual dictionaries greatly outperforms the pure statistically-based baseline and is practically at the same level as the algorithm of Gale and Church. Still, algorithm of Gale and Church uses certain parameters especially pre-calculated, thus, it cannot be considered an unsupervised algorithm as it is in our case. Also, it relies on the hypothesis of normal distribution, in contrast with our algorithm that does not rely on any distribution.

5 Evaluation Based on Reconstruction of Text Structure

Traditional evaluation schemes usually invoke direct comparison with gold standard, or reference text alignment, see formal definitions of this kind of alignment in [4]. Both precision and recall can be computed, as well as the derived F-measure. It is mentioned in that paper that we can measure these values using different granularity, i.e., for alignment on the sentence level, correctly aligned words or characters can be measured. The authors do not mention the task of paragraph level alignment.

We suggest considering evaluation of an alignment algorithm as the task of global text structure reconstruction. Namely, if we are evaluating the correctness of correspondences at the paragraph level, let us eliminate all paragraph boundaries in one of the texts and allow the algorithm to put back the paragraph marks based on the paragraph structure of the other text and the data of the alignment algorithm itself. Then we evaluate the correctness of the restored paragraph marks using the structure of paragraphs in the other language. We cannot rely on the known paragraph structure for the same language, because the paragraphs can be aligned correctly in different manner (2-1, 3-1, etc.). In practice, this is done by considering all sentences in one of the text as paragraphs, and then paragraph-level alignment is performed.

The restoration of text structure is somehow similar to the evaluation technique based on counting the correspondences on the other level of granularity (say, using sentences for paragraphs, etc.), because it also uses the units of the lower level, but it is essentially the different task. The main difference is that while the algorithm is trying to recreate the text structure using the units of the lower level of granularity, it comes across many possibilities that it never would consider working only with the existing units. It is especially well-seen for alignment at the paragraph level. Usually, the alignment of paragraphs is not considered as an interesting task since in the majority of existing parallel text the paragraphs, even the large ones, have clear correspondences. Meanwhile, if we consider the task of text reconstruction, the paragraph alignment task becomes an interesting problem. Thus, we can evaluate and compare different approaches to paragraph level alignment. This technique can be useful also for automatic search of parallel texts in Internet.

Another consideration is related to corpus structure. As the majority of parallel texts have very similar structure at paragraph level, the problem of alignment at this level is difficult to evaluate, because in any corpus there are few interesting cases of paragraph alignment. Applying the suggested method of evaluation, we resolve the problem of the lack of non-trivial cases of the paragraph level alignment, because now any paragraph of any text is split into sentences and it is a challenge for aligning algorithms.

We conducted experiments using dynamic programming approach described above. Our goal was to compare the performance of the statistical and lexical approaches to similarity calculation using the proposed evaluation method based on reconstruction of the global text structure.

As an example of statistical approach, we used an implementation of Gale and Church algorithm [1], though we had to modify it according to the task. The problem is that this algorithm only takes into account alignment of maximum 2-2 correspondences (i.e., 3-2 is impossible, etc.) and it is penalizing the correspondences that are different from 1-1. We had to remove these penalizations because there can be many more possible correct correspondences, like, for example, 10-1, etc., and these should not be penalized. Obviously, it affects the original algorithm performance. It is the question of further investigations to determine how to modify penalizations in this algorithm or what improvements should be added to achieve the best performance.

For the lexical approach, we used the implementation of our lexical-based alignment algorithm for English-Spanish text pairs (see previous sections). For the moment, we also do not add any penalization for size of fragments, for absolute positions of fragments, or for relative position of lexical units in fragments. We expect that implementation of these parameters will improve the performance of our algorithm.

We made our experiments using the extract of 15 paragraphs from the text mentioned above. Note that it is a difficult case of non-literal translation. We made complete analysis using dynamic programming. The information about Spanish paragraphs was suppressed.

The results of the comparison using both methods are as follows for precision: 84% in lexical approach vs. 26% in statistical approach. We count the correct correspondences using the paragraph structure of the English text. When the algorithm united two paragraphs that were separated both in the Spanish text and in the English text, we counted it as an error for the half of the restored sentences. Still, it is interesting to analyze if it is the same type of error as failing to find the correct correspondence. Note that the information about the paragraph separation in Spanish text was not used.

The problem with the statistical method is that once it makes incorrect alignment, it is difficult for it to return to the correct correspondences.

6 Conclusions

We described a dynamic programming algorithm with lexical similarity for alignment of parallel texts. This is unsupervised algorithm. We conducted the experiments of the traditional evaluation obtaining very similar results with the supervised algorithm of Gale and Church. We used fiction texts that are difficult cases for alignment.

We also presented a new method for evaluation of the algorithms of parallel texts alignment. This method consists in restoration of the structure of the text in one of the languages using the units of the lower level and the structure of the text in the other language. For example, in case of the paragraph level alignment, the sentences are used to constitute the restored paragraphs in one of the languages. The advantage of this method is that it does not depend on corpus data that is random. Another consideration is that in case of paragraphs the corpus data often is trivial. Applying the proposed method, we obtain the basis for comparison of different alignment algorithms that is not trivial at the paragraph level. We conducted experiments on a fragment of English-Spanish text using the restoration method. The text was a fiction text with non-literal translation. Lexical and statistical approaches were tried for calculation of similarity using dynamic programming approach. We obtained much better results for the lexical method, though we expect that the statistical method can be improved for the proposed task.

References

[1] Gale, W.A., Church, K.W.: A program for Aligning Sentences in Bilingual Corpora. In: Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics, Berkeley, California (1991)

[2] Gelbukh, A., Sidorov, G., Vera-Félix, J.Á.: Paragraph-Level Alignment of an English-Spanish Parallel Corpus of Fiction Texts using Bilingual Dictionaries. In: Sojka, P., Kopeček, I., Pala, K. (eds.) Text, Speech and Dialogue. LNCS (LNAI), vol. 4188, pp. 61–67. Springer, Heidelberg (2006)

[3] Chunyu, K., Webster, J.J., Sin, K.K., Pan, H., Li, H.: Clause alignment for Hong Kong legal texts: A lexical-based approach. International Journal of Corpus Linguistics 9(1), 29–51 (2004)

[4] Langlais, Ph., M. Simard, J. Veronis, Methods and practical issues in evaluation alignment techniques. In: Proceeding of Coling-ACL-98 (1998)

[5] Moore, R.C.: Fast and Accurate Sentence Alignment of Bilingual Corpora. AMTA-2002, pp. 135–144 (2002)

Electronic Dictionaries and Transducers for Automatic Processing of the Albanian Language

Odile Piton[1], Klara Lagji[2], and Remzi Përnaska[3]

[1] University Paris1 Panthéon-Sorbonne
[2] University of Tirana Albania
[3] University of Poitiers France

Abstract. We intend on developing electronic dictionaries and Finite State Transducers for the automatic processing of the Albanian Language. We describe some peculiarities of this language and we explain how FST and generally speaking NooJ's graphs enable to treat them. We point out agglutinated words, mixed words or 'XY' words that are not (or cannot be) listed into dictionaries and we use FST for their dynamic treatment. We take into consideration the problem of unknown words in a lately reformed language and the evolving of features in the dictionaries.

Keywords: Morphological Analysis, Electronic Dictionary, Finite State Transducer, Albanian Language, Automatic Processing of Open Lists of Words.

1 Introduction

The evolution of powerful of computers and the increased size of central and external memory storage make it possible to process natural language with dictionaries. However, "Large-scale lexical acquisition is a major problem since the beginning of MT" [1]. Boitet suggests mutualizing creation and usage of lexical resources. The automatic processing of Albanian is not yet developed [7] [9] [10]. So this work focuses on syntactic and basic semantic features. The completion of semantic information requires many steps and lots of human labour.

A lot of papers have been written on dictionaries for NLP. In this one, we won't describe the usual work of generating the list of known words with their category and features, building tools to describe and generate flexed words. Instead, we address some advanced problems of this language and we explain how Finite State Transducers -FST- described with NooJ's graphs enable to solve them.[1] Nooj furnishes tools for automatically constructing words by transducers: there are inflectional grammars, morphological grammars and syntactic grammars.[2]

Dictionaries are two levelled structures [14]. "The words of the natural language are complex entities and they sometimes have a much more elaborated internal structure… they often join other lexical elements to form wider units as compound

[1] See http://www.nooj4nlp.net/ for information on NooJ.

[2] See http://marin-mersenne.univ-paris1.fr/site_equipe_mm/O_Piton/Piton.html for more information.

Z. Kedad et al.(Eds.): NLDB 2007, LNCS 4592, pp. 407–413, 2007.

words or expressions [15].” Language is a creative process. The dictionaries needed by automatic treatment must register basic vocabulary and be associated with tools, with creative paradigms, that compute the new words of the constructed part [2]. The choice of the set of syntactic and semantic features is important.

2 Some Properties of Albanian

The late reform, in 1972, has lead to standard Albanian Literary Language. A linguistic move took place in order to “unify” the two Albanian dialects: Gheg and Toskë. If the official language seems to be the language of the media [6] and the language of the schools, it is not the language of every Albanian speaker. This language is still subject to variation. Two usual assumptions are not true for Albanian: *all the stems are not in dictionaries and syntactic information is often lacking*. This gives an unusual importance to the treatment of “unknown words”. We must note that lot of verbal forms are not plain forms but “analytic forms”. We cannot build the automated processing of Albanian using a bottom-up method. We develop specific tools for analytic forms.

Our first data was a paper dictionary. As it has been noted: “Structure of entries in dictionaries varies considerably, inside one dictionary as well as between different dictionaries: it seems that any type of information can be found in any position in a dictionary. Nevertheless, in spite of these variations, human readers are able to interpret the entries easily and this, without needing to consult introductive explanations. It is clear that there are several principles and underlying regularities.”[14] These regularities are made of slashes, commas, parenthesis and hyphens, which surround grammatical, linguistic and technical information. We have used the Albanian-French dictionary of the book “Parlons Albanais” [4]. We have observed its format. Part of speech: noun, verb, etc. is not always written. We have developed heuristic to infer it.

2.1 About Digraphs

Albanian alphabet has 36 letters. It uses the Latin alphabet. 9 letters are written with “digraphs” or double characters: *dh gj ll nj rr sh th xh zh*, but they must be considered as one letter.

Some regular paradigms to make stem allomorphs have to be redefined. Some nouns construct an allomorph in loosing *ë* and receiving *a*. E.g. *motër*→ *motra.* The rules “*go left one letter*” then delete one letter has to be redefined for *vjehërr*→ *vjehrra* as “*go left two chars*” according to the fact that ‘*rr*’ is one letter with two characters. The whole set of words that obey the regular paradigm, has to be dispatched between the ‘1 char.’ vs. ‘2 char.’ paradigms.

Some words like son *bir* and devil *djall* drop their last letter and receive *j* in the plural. But the last consonant of *djall* is a double letter, so we need two paradigms. ‘*Delete 1 letter* vs. *delete 2 letters* then insert j’ *bir*→ *bij* and *djall*→ *djaj.*

2.2 About Verbal System

A verb can have one or two forms: *Z* active form and/or *Z-hem* not active forms: e.g. *laj* (to wash) and *lahem* (to wash oneself). The two forms have the same past participle and four tenses of non-active forms are build on the same tense as the active form preceded by the particle "u": e.g. *lava* (I washed-aorist) *u lava* (*I* washed myself). Some tenses use particles: *të* and *do* and some tenses include others. E.g. *laj, të laj, do të laj*, (active form) and *lahem, të lahem, do të lahem* (non-active form) are 6 different tenses. If we recognize *laj* as present tense, we don't see that it is part of *do të laj,* a form of the future tense. So we need to describe analytic forms. We have drawn corresponding graphs [11], to build one form for each person of each conjugation. Active and non active verbs are described by separate graphs (200 graphs and sub-graphs). Non active forms can be separated into subsets with syntactic and semantic features.

For some verbs, *aorist* is marked by a process known as "*ablaut*"; there is a change in vowel, e.g. '*e*→*o'*. We have to distinguish two subsets of verbs and to define two graphs: 'insert a, *go left 1 letter* vs. *go left2 letters*, delete 1 letter, insert o' *heq*→ *hoqa*, *hedh*→*hodha.*

2.3 About Nouns, Pronouns and Adjectives

In Albanian, nouns and pronouns are usually inflected. A great number of masculine words in the singular are feminine in the plural. So the gender is not a static property of a noun. *It is not written in the dictionary.* Declension position is at the end of the word, but for three pronouns, it is inside, e.g. *cilido, cilitdo, and cilindo.* It is the same phenomenon for '*auquel* / *auxquels'* in French: two concatenated words and the first one is flexed.

Foreign Named Entities are transcript according to Albanian phonetic: *Shekspir* Shakespeare*, Xhems Xhojs* James Joyce*, Sharl dë Gol* Charles de Gaulle. They are flexed as other nouns. So are acronyms, but they are preceded by a dash *OKB-ja*, *OKB-në,* two flexions of *OKB* (UNO).

Most adjectives, some nouns and some pronouns are preceded by a particle called article. These articles have declensions; their four forms can be *i, e, të, së*. These declensions vary according to the place of the articulated adjective or articulated noun in nominal syntagm.

Homonymy interferes with articulated words: e.g. *parë* seen; '*të parë'*, aspect, '*i parë'*, chief; and ancestor. The whole sequence must be recognized all over.

3 Dynamic Method for Albanian with NooJ

NooJ processes both simple and compound words, if their lemma is in NooJ's dictionaries. Some agglutinated words, or 'XY' built words need extra tools.

3.1 FST for Agglutinated or Mixed Forms

Agglutination and Mix for the Imperative Tense. The imperative can concatenate with the clitic complement *më*: *laj* (wash*), lani* (let you wash), *më laj* (wash me), *më lani* (let you wash me) can be used in the form *lamë* and *lamëni*. These agglutinations obey to specific rules (phonetic or category of verb) and must be described. NooJ's graphs allow us to recognize such agglutinated and mixed forms: *lamëni = lani + më.*

Graphs for Atonic (or Short) Forms. Morphological grammar allows us to process contracted words. For example, in French the word '*au'* is expanded as '*à le'*. There are similar contractions in Albanian. When there are several clitics before a verb, they are often contracted: *'të e'* into *ta* (*të* particle for subjunctive, or clitic dative and *e* clitic accusative), *'më e'* into *ma*, and *'i i'* or *'i e'* into *ia*. Clitic forms are '*më të e i na ju u'*. The contraction can be one single sequence or use an apostrophe: '*të i*'→ *t'i.* But some contractions are ambiguous and the FST uses extra information to disambiguate.

See a graph to expand *ma, ta* and *ia* in figure 1.

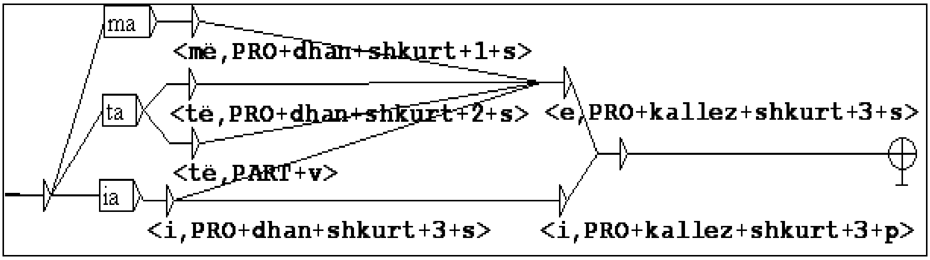


Fig. 1. Morphological graph for "decontraction" of some 'short forms' of personal pronouns: *ma, ia, ta*. Read from left to right. Last line processes 'ia' (in the box, the outputs are under the lines, they perform the translation). The graph has two ways: one way translates '*ia'* into <i> <i>, the second translates '*ia'* into <i> <e>.

3.2 FST for Open Lists

Albanian has some productive paradigms, from derivation rules to concatenation rules that are very active and produce concatenated words. It is impossible to list all XY words in dictionaries, because *the number is infinite*. This is called *open lists of words.* An interesting property is that most of the words are concatenated without contraction or modification. That makes them easy *to recognize dynamically*. See in figure 2 a graph to recognize dynamically unknown words beginning with *para, pa* or *nën.*

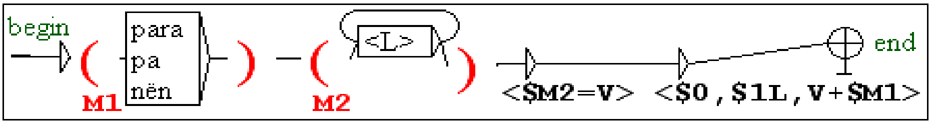


Fig. 2. Graph for XY words where X is *para*, *pa* or *nën*; Y is a verbal form. Explanation for *nënkuptoj:* the graph compares the first letters with *para*, *pa* and *nën. A variable M1* receives the result *nën*; then the loop on <L> extracts the second part *kuptoj* into M2. M2 is compared to the verbs by <$M2=V>. K*uptoj* is recognized as a verb, so the last part of the graph generates dynamically the entry: *nënkuptoj,kuptoj,V+nën.*

‘XY’ Words Built by Concatenation. Y is a verb, noun, or adjective. X can be:

- an affix: prefix like *ç, sh*, zh, *pa* (negation), e.g. *krehur* (combed), *pakrehur* (not combed).
- a preposition like *nën* (under), e.g. *kuptoj* (to understand), *nënkuptoj* (to suggest).
- a noun like *flokë* (hair) e.g. *bardhë* (white), *flokëbardhë* (white haired).
- an adjective like *keq* (bad), *besim* (trust), *keqbesim* (distrust); or an adjective finished by a “o” e.g. *leksiko-gramatikore* (lexical-grammar), X is invariable, Y only takes the marks of feminine or plural: *sesioni tecniko-shkencor* (technico-scientific session) [7].

This process is recursive and very productive. With the morphologic graphs, NooJ gives a tool that allows us to recognize such constructed words on demand [12]. A morphological graph can recognize several parts in a word and can compare them either to a specified list, or to a set of words. When the comparison is successful, the word is recognized and receives the lexical features and the syntactic features.

‘XY’ Words with Numbers. Concatenated cardinal numbers can be the first part of words. The second part of the ‘XY’ word is compared to a list or to the dictionary. E.g. number 22 *njëzet e dy* concatenated and followed by *vjeçar* (aged of) gives an adjective: *njëzetedyvjeçar* (twenty-two years old). This can also be done with words like *katesh* (floor), *mujor* (month), *orësh* (hour), etc. Words like *njëzetedyvjeçar* are not listed into dictionaries.

3.3 FST for Derivation

We noticed earlier that lot of words are not in dictionaries. Unknown words are submitted to different processes. They are compared to the derived forms (according to the stem). Their affix gives information on their POS. E.g. most of words with suffix *ik* or *ike* are adjectives, while a few are nouns. The “Inverse Dictionary of Albanian” [8] makes it possible to classify the affixes of entries. However it does not present the plural forms.

Derivational morphology is a well-known strategy to improve coverage of lexicon by affix operations by iterative process. Automatic analysis and filtration is exposed by [13]. A study of derivation is exposed by [3]. These words inherit argument structure of the base word, or part of it. The Albanian language is strongly constraint and derivation is very regular. The derivation is used for the operation of tagging unknown words. NooJ’s morphological graphs construct derived words from a root and give them a category and some features.

About words with an article. We have noticed that the words that occur many times in the text should be grouped before making an entry for it. We notice a problem for articulated words that are not grouped with their article because ‘unknowns’ are single words.

Table 1. Example of features for the nouns

N_Nb = s + p;	The number can be singular or plural
N_Genre = m + f + as;	The gender can be masc., fem. or neutral
N_Shquar = shquar + pashquar;	A noun can be definite or undefinite
N_Rasa = emer + rrjedh + gjin + kallez + dhan;	This is the list of names of declensions for nouns

3.4 Features in the Dictionaries

This is necessary to be able to evolve and/or improve the dictionaries. An important part of the work is to organize the syntactic and semantic features and tags for each category. Whenever it is necessary to evolve syntactical and semantic tags, the dictionary will be modified. It is important to be able to adapt it easily. For example, a new study can make it necessary to define two tags instead of one. So the set of features and tags must be carefully registered and regularly updated. NooJ registers the set of features in a file called "*properties definition*". This file can be used to display lexical information as table.

4 Conclusion

It is necessary to add entries from others Albanian dictionaries to these first electronic dictionaries. We are aware that there is a lot more work to be done.

In Albania, the Computational Linguistics is not very developed.

Since 1998, a course in Computational Linguistics is organized with the students of the Department of Albanian Linguistics at the Tirana University. It aims at the sensitization of the future linguists and language teachers about new approaches of the natural language processing. At the end of the course, the students have to do a study application of this methodology to Albanian language. But, actually, no organized collaboration exists between linguists and computer scientists.

References

1. Boitet, Chr.: Méthode d'Acquisition lexicale en TAO: des dictionnaires spécialisés propriétaires aux bases lexicales généralistes et ouvertes. TALN, 8E Conf. Sur le TALN. Tours, pp. 249–265 (2001)
2. Carré, R., Degrémont, J.F., Gross, M., Pierrel, J.-M., Sabah, G.: Langage Humain et Machine, Paris, Presses du CNRS (1991)
3. Gdaniec, C., Manandise, E., McCord, M. Derivational Morphology to the rescue: How It Can Help Resolve Unfound Words in MT. In: Proceedings, Santiago de Compostela, Spain, pp. 129–131 (2001)
4. Gut, Chr., Brunet-Gut, A., Përnaska, R.: Parlons Albanais Paris L'Harmattan Paris (1999)
5. Habert, B., Nazarenko, A., Zweigenbaum, P., Bouaud, J.: Extending an existing specialized semantic lexicon. In: Rubio, A., Gallardo, N., Castro, R., Tejada, A. (eds.) First International Conference on Language Resources and Evaluation, Granada, pp. 663–668 (1998)
6. Hasani, Z.: Le Déclin de la Langue Albanaise? In: Shekulli (12 Novembre 2002)
7. Lagji, K.: Etude sur le Statut du Mot en Albanais dans le Cadre des Traitements Automatiques des Langues. In: Annotation Automatique de Relations Sémantiques et Recherche d'Informations: vers de Nouveaux Accès aux Savoirs. Université Paris-Sorbonne, 27-28 Octobre, Paris (2006)
8. Murzaku, A.: Albanian Inverse Dictionary 32,005 words http://www.lissus.com/resources_download.htm
9. Piton, O., Përnaska, R.: Etude de l'Albanais en Vue de Construire des Outils pour son Traitement Automatique, Journées NooJ Besançon (2005)

10. Piton, O., Përnaska, R.: Constitution de Dictionnaires Electroniques pour l'Albanais, et Grammaire du Groupe Nominal avec Nooj, Belgrade (2006)
11. Silberztein, M.: Dictionnaires électroniques et analyse automatique de textes. Le système INTEX. Masson Ed. Paris Milan Barcelone Bonn (1993)
12. Silberztein, M.: NooJ's dictionaries. In: Proceedings of LTC (2005) Poznan University (2005)
13. Tsoukermann, E., Jacquemin, Chr.: Analyse Automatique de la Morphologie Dérivationnelle, et Filtrage de Mots Possibles, Mots possibles et mots existants, Silexicales. (28-29 avril 1997), pp. 251–259 (1995)
14. Véronis, J., Ide, N.: Encodage des dictionnaires électroniques: problèmes et propositions de la TEI. In: Piotrowsky, D. (ed.) Lexicographie et informatique - Autour de l'informatisation du Trésor de la Langue Française. Actes du Colloque International de Nancy (29, 30, 31 mai 1995), pp. 239–261 (1995)
15. Wehrli, E.: L'analyse syntaxique des langues naturelles. In: Problèmes et Méthodes, Masson Ed. Paris Milan Barcelone (1997)

Two Methods of Evaluation of Semantic Similarity of Nouns Based on Their Modifier Sets[*]

Igor A. Bolshakov and Alexander Gelbukh

Center for Computing Research (CIC),
National Polytechnic Institute (IPN), Mexico City, Mexico
{igor,gelbukh}@cic.ipn.mx

Abstract. Two methods of evaluation of semantic similarity/dissimilarity of English nouns are proposed based on their modifier sets taken from Oxford Collocation Dictionary for Student of English. The first method measures similarity by the portion of modifiers commonly applicable to both nouns under evaluation. The second method measures dissimilarity by the change of the mean value of cohesion between a noun and modifiers, its own or those of the contrasted noun. Cohesion between words is measured by Stable Connection Index (*SCI*) based of raw Web statistics for occurrences and co-occurrences of words. It is shown that the two proposed measures are approximately in inverse monotonic dependency, while the Web evaluations confer a higher resolution.

1 Introduction

There are numerous works on evaluation of semantic similarity/dissimilarity between words, see [10] and references therein for a review. The majority of evaluations are based on semantic hierarchies of WordNet or EuroWordNet [2, 3]. Semantic dissimilarity between words is measured by the number of steps that separate corresponding nodes of the hierarchy. The hierarchy nodes are synsets including the words under evaluation, while the arcs are subset-superset links connecting these synsets. The greater is the distance, the lower is similarity. This measure proved to be useful in many applications and tasks of computational linguistics, such as word sense disambiguation [8, 416], information retrieval, etc.

In fact, there exists an alternative way to evaluate semantic similarity, namely through comparison of the sets of words frequently co-occurring in texts in close vicinity to words under evaluation. The more similar are the recorded beforehand sets of standard neighbors of any two words of the same POS, the more semantically similar are the words. As applied to nouns, the accompanying words are primordially modifiers, whose role in European languages is usually played by adjectives and—in English—also by attributively used nouns staying in preposition.

In this paper, semantic similarity/dissimilarity of English nouns is evaluated by two different methods, both based on those standard modifier sets for few tens of commonly used English nouns that are registered for them in OCDSE—the most reliable

[*] Work done under partial support of Mexican Government (CONACyT, SNI, SIP-IPN).

Z. Kedad et al.(Eds.): NLDB 2007, LNCS 4592, pp. 414–419, 2007.

source of English collocations [9]. The nouns were selected with preference to those with greater numbers of modifiers recorded.

In the first method, the similarity $Sim(N_1, N_2)$ of the noun N_1 to the noun N_2 is measured by the ratio of the number of modifiers commonly applicable to the both nouns and the number of modifiers of N_2.

In the second method, the dissimilarity $DSim(N_1, N_2)$ of N_1 from N_2 is measured by the residual of two mean values of specially introduced *Stable Connection Index*. *SCI* is close in its definition to Mutual Information of two words [7]. It operates by raw statistics of Web pages that contain these words and their close co-occurrences and does not require repetitive evaluation of the total amount of pages under search engine's control [4]. One mean value covers *SCI*s of all 'noun → its own modifier' pairs, another mean value covers *SCI*s of all 'N_1 → modifier of N_2' pairs. English modifiers usually stay just before their nouns forming bigrams with them, and this facilitates rather reliable Web statistic evaluations.

To put it otherwise, *Sim* is determined through coinciding modifiers of nouns, while *DSim* is determined through alien modifiers. As the main result, the *Sim* and *DSim* measures proved to be approximately connected by inverse monotonic dependency. However, *DSim* seems preferable because of its higher resolution: the numerous noun pairs with zero *Sim* values differ significantly with respect to *DSim*.

2 Experimental Modifier Sets

We took English nouns with all their recorded modifiers—both adjectives and nouns in attributive use—from OCDSE. The nouns were picked up from there in rather arbitrary manner, approximately one noun per nine OCDSE pages. At the same time, our preferences were with the most productive nouns, i.e. having vaster modifier sets.

Table 1. Selected nouns and sizes of their modifier sets

S/N	Noun	MSet Size	S/N	Noun	MSet Size	S/N	Noun	MSet Size
1	*answer*	44	12	*difference*	53	23	*experience*	53
2	*chance*	43	13	*disease*	39	24	*explanation*	59
3	*change*	71	14	*distribution*	58	25	*expression*	115
4	*charge*	48	15	*duty*	48	26	*eyes*	119
5	*comment*	39	16	*economy*	42	27	*face*	96
6	*concept*	45	17	*effect*	105	28	*facility*	89
7	*conditions*	49	18	*enquiries*	45	29	*fashion*	61
8	*conversation*	52	19	*evidence*	66	30	*feature*	51
9	*copy*	61	20	*example*	52	31	*flat*	48
10	*decision*	40	21	*exercises*	80	32	*flavor*	50
11	*demands*	98	22	*expansion*	44			

For 32 nouns taken, total amount of modifiers (partially repeating) is 1964, and the mean modifiers group size equals to 61.4, varying from 39 (for *comment* and *disease*) to 119 (for *eyes*). The second and the third ranks determined by the set sizes are with *expression* (115) and *effect* (105). The nouns selected and sizes of their modifier sets are demonstrated in Table 1.

We had to limit the number of *Nouns* to 32 units, since the total amount of accesses to the Web in experiments of the second method (cf. Section 5) grows rapidly, approximately as (*Nouns* + 40) × (*Nouns* + 1), so that, taking into account severe limitations of Internet searchers, we could afford several days for acquiring all necessary statistics, but scarcely a month or more.

The nouns *conditions, demands, enquiries, exercises,* and *eyes* were taken in plural, since they proved to be more frequently used with their recorded modifier sets in plural than in singular.

3 Semantic Similarity Based on Intersection of Modifier Sets

The similarity $Sim(N_i, N_j)$ in the first method is mathematically defined through the intersection ratio of modifier sets $M(N_i)$ and $M(N_j)$ of the two nouns by the formula

$$Sim\left(N_i, N_j\right) \equiv \frac{\left|M\left(N_i\right) \cap M\left(N_j\right)\right|}{\left|M\left(N_i\right)\right|}, \tag{1}$$

where $|M(N_i)|$ means cardinal number of the set $M(N_i)$ and $\cap$ set intersection, cf. [6].

With such definition, the similarity measure is generally asymmetric: $Sim(N_i, N_j) \neq Sim(N_j, N_i)$, though both values are proportional to the number of commonly applicable modifiers. We can explain the asymmetry by means of the following extreme case. If $M(N_i) \subset M(N_j)$, each member of $M(N_i)$ has its own counterpart in $M(N_j)$, thus $Sim(N_i, N_j)$ reaches the maximum equal to 1 (just as when $M(N_i) = M(N_j)$), but some members of $M(N_j)$ have no counterparts in $M(N_i)$, so that $Sim(N_j, N_i) < 1$.

4 Measurement of Words Cohesion by Means of Internet

It is well-known that any two words W_1 and W_2 may be considered forming a stable combination if their co-occurrence number $N(W_1, W_2)$ in a text corpus divided by S (the total number of words in the corpus) is greater than the product of relative frequencies $N(W_1)/S$ and $N(W_2)/S$ of the words considered apart. Using logarithms, we have obtain the log-likelihood ratio or Mutual Information [7]:

$$MI(W_1, W_2) \equiv \log \frac{S \cdot N(W_1, W_2)}{N(W_1) \cdot N(W_2)}.$$

MI has important feature of scalability: if the values of all its building blocks S, $N(W_1)$, $N(W_2)$, and $N(W_1, W_2)$ are multiplied by the same factor, *MI* preserves its value.

Any Web search engine automatically delivers statistics on a queried word or a word combination measured in numbers of relevant Web pages, and no direct information on word occurrences or co-occurrences is available. We can re-conceptualize *MI* with all $N()$ as numbers of relevant pages and S as the page total managed by the engine. However, now $N()/S$ are not the empirical probabilities of relevant events: the words that occur at the same a page are indistinguishable in the raw statistics, being counted only once, while the same page is counted repeatedly for each word included. We only keep a vague hope that the ratios *N*()/*S* are monotonically connected with the corresponding empirical probabilities for the events under consideration.

In such a situation a different word cohesion measure was construed from the same building blocks [1]. It conserves the feature of scalability, gives very close to *MI* results for statistical description of rather large sets of word combinations, but at the same time is simpler to be got from Internet, since does not require repeated evaluation of the whole number of pages under searcher's control. The new cohesion measure was named Stable Connection Index:

$$SCI(W_1,W_2) \equiv 16 + \log_2 \frac{N(W_1,W_2)}{\sqrt{N(W_1)\cdot N(W_2)}}. \qquad (2)$$

The additive constant 16 and the logarithmic base 2 were chosen rather arbitrary, but such scaling factors do not hamper the purposes of this paper and permit to consider words W_1 and W_2 cohesive, if $SCI(W_1,W_2)$ is positive.

Since our experiments with Internet searchers need at least several days to complete, some additional words on Web searchers are worthwhile now.

The statistics of searcher have two sources of changing in time. The first source is monotonic growing because of steady enlargement of searcher's DB. In our experience, for well saturated searcher's BDs and words forming stable combinations, the raw statistics $N(W_1)$, $N(W_2)$, $N(W_1,W_2)$ grow approximately with the same speed, so that *SCI* keeps the same value (with the precision to the second decimal digit), even if the statistics are got in different time along the day of experiments.

The second, fluctuating source of instability of Internet statistics is selection by the searcher of a specific processor and a specific path through searcher's DB—for each specific query. With respect to this, the searchers are quite different. For example, Google, after giving several very close statistics for a repeating query, can play a trick, suddenly giving twice fewer amount (with the same set of initial snippets), thus shifting *SCI* significantly. Since we did not suffer of such troubles so far on behalf of AltaVista, we preferred it for our purposes.

5 Semantic Dissimilarity Based on Mean Cohesion Values

Let us first consider the mean cohesion values

$$\frac{1}{|M(N_i)|}\sum_{A_k \subset M(N_i)} SCI(N_i, A_k)$$

between the noun N_i and all modifiers A_k in its own modifier set $M(N_i)$. One can see in the Table 2 that all mean *SCI* values are positive and mainly rather big (4 to 8), except for *enquiries.* On the latter occasion, we may suppose that occurrence statistics of British National Corpus—the base for selection of collocations in OCDSE—differ radically from Internet statistics that is not British oriented in its bulk. Hence the collocations *intellectual/joint/open/critical/sociological... enquiries,* being rather rare in whole Internet, were inserted to OCDSE by purely British reasons. This is not unique case of British vs. USA language discrepancies. Except of orthographic differences like *flavour* vs. *flavor*, but we did not feel free to sift out such OCDSE collocations as *coastal flat* 'property by the sea,' which proved to be rare in Internet as a whole.

Table 2. The mean SCI values of nouns with their own modifiers

S/N	Noun	Mean SCI	S/N	Noun	Mean SCI	S/N	Noun	Mean SCI
1	*answer*	6.3	12	*difference*	6.2	23	*experience*	7.7
2	*chance*	4.9	13	*disease*	8.3	24	*explanation*	6.1
3	*change*	6.5	14	*distribution*	6.7	25	*expression*	4.9
4	*charge*	5.6	15	*duty*	5.6	26	*eyes*	6.0
5	*comment*	4.4	16	*economy*	6.7	27	*face*	5.7
6	*concept*	5.9	17	*effect*	6.7	28	*facility*	4.5
7	*conditions*	6.5	18	*enquiries*	1.4	29	*fashion*	5.1
8	*conversation*	6.0	19	*evidence*	8.0	30	*feature*	5.9
9	*copy*	5.4	20	*example*	6.1	31	*flat*	4.3
10	*decision*	7.2	21	*exercises*	4.0	32	*flavor*	6.1
11	*demands*	4.1	22	*expansion*	6.4			

Passing to SCI evaluation of 'noun → modifier of a different noun' pairs that mainly are not normal collocations, we can frequently meet the cases with zero co-occurrence number in Internet. Then formula (2) gives *SCI* value equal to –∞. To avoid the singularity, we take the value –16 for such cases, i.e. maximally possible positive value, but with the opposite sign.

Table 3. Most and lest similar noun pairs

Lest dissimilar noun pairs				Most dissimilar noun pairs			
Noun$_1$	**Noun$_2$**	**Sim**	**DSim**	**Noun$_1$**	**Noun$_2$**	**Sim**	**DSim**
enquiries	*explanation*	0.156	0.3	*disease*	*enquiries*	0.000	18.5
enquiries	*distribution*	0.022	0.5	*eyes*	*enquiries*	0.017	15.8
enquiries	*comment*	0.111	0.6	*effect*	*enquiries*	0.029	14.8
enquiries	*conversation*	0.089	0.6	*face*	*enquiries*	0.010	14.7
enquiries	*change*	0.044	0.9	*experience*	*enquiries*	0.000	14.4
difference	*change*	0.321	1.1	*disease*	*economy*	0.000	14.2
enquiries	*fashion*	0.022	1.1	*disease*	*chance*	0.000	14.0
enquiries	*charge*	0.067	1.2	*flavor*	*enquiries*	0.020	14.0

We determine the dissimilarity measure by the formula

$$DSim\left(N_i, N_j\right) = \frac{1}{\left|M\left(N_i\right)\right|} \sum_{A_k \in M(N_i)} SCI\left(N_i, A_k\right) - \frac{1}{\left|M\left(N_j\right)\right|} \sum_{A_k \in M(N_i)} SCI\left(N_j, A_k\right) \qquad (3)$$

The diminuend at the right part of (3) is the mean *SCI* value of N_i with its own modifiers, while the subtrahend is the mean *SCI* value of N_j estimated with respect to all modifiers of N_i. It is clear that *DSim*(N_i, N_j) is minimal possible (0) for $i = j$.

For different nouns, lest and most dissimilar noun pairs are shown in Table 3. The pair {*enquiries, explanation*} proved to be the most similar by *DSim* criterion, while the pair {*disease, enquiries*}, the most dissimilar.

6 Conclusions

We have proposed two methods of how numerically evaluate semantic similarity of any two English nouns. The evaluations are based on comparison of standard modifiers of the nouns. The first method evaluates similarity by the portion of common modifiers of the nouns, while the second one evaluates dissimilarity by the change of the mean cohesion between a given noun and modifiers, when the set of its own modifiers commuted into the set of alien ones.

The comparison of *Sim* and *DSim* values for as few as 16 pairs in Table 3 shows that the pairs with maximal *Sim* usually have minimal *DSim* and vice versa, i.e. an inverse monotonic dependency exists between the two measures. One can note that *DSim* has higher resolution for semantically most different nouns. Indeed, the numerous pairs with zero *Sim* values have quite diverse *DSim* values, from 14.0 for {*disease, flat*} to 4.2 for {*flat, answer*}. Hence the use of *DSim* measure seems preferable.

Cohesion measurements are based on raw Web statistics of occurrences and co-occurrences of supposedly cohesive words. For both methods, the standard modifier sets are taken from Oxford Collocations Dictionary for Students of English. It is shown that dissimilarity measured through the Web has higher resolution and thus may have greater reliability.

Both method do not depend on language and can be easily tested on the resources of other languages. For English, it is worthwhile to repeat evaluations for a greater number of nouns and for different source of modifiers sets, e.g. for a large corpus of American origin.

References

1. Bolshakov, I.A., Bolshakova, E.I.: Measurements of Lexico-Syntactic Cohesion by means of Internet. In: Gelbukh, A., de Albornoz, Á., Terashima-Marín, H. (eds.) MICAI 2005: Advances in Artificial Intelligence. LNCS (LNAI), vol. 3789, pp. 790–799. Springer, Heidelberg (2005)
2. Fellbaum, C.: WordNet: An Electronic Lexical Database. MIT Press, Cambridge (1998)
3. Hirst, G., Budanitsky, A.: Correcting Real-Word Spelling Errors by Restoring Lexical Cohesion. Natural Language Engineering 11(1), 87–111 (2005)
4. Keller, F., Lapata, M.: Using the Web to Obtain Frequencies for Unseen Bigram. Computational linguistics 29(3), 459–484 (2003)
5. Ledo-Mezquita, Y., Sidorov, G.: Combinación de los métodos de Lesk original y simplificado para desambiguación de sentidos de palabras. In: International Workshop on Natural Language Understanding and Intelligent Access to Textual Information, in conjunction with MICAI-2005, Mexico, pp. 41–47 (2005)
6. Lin, D.: Automatic retrieval and clustering of similar words. COLING-ACL 98 (1998)
7. Manning, C.D., Schütze, H.: Foundations of Statistical Natural Language Processing. MIT Press, Cambridge (1999)
8. McCarthy, D., Rob, K., Julie, W., John, C.: Finding PredominantWord Senses in Untagged Text. ACL-2004 (2004)
9. Oxford Collocations Dictionary for Students of English. Oxford University Press (2003)
10. Patwardhan, S., Banerjee, S., Pedersen, T.: Using Measures of Semantic Relatedness for Word Sense Disambiguation. In: Gelbukh, A. (ed.) Computational Linguistics and Intelligent Text Processing. LNCS, vol. 2588, Springer, Heidelberg (2003)

A Service Oriented Architecture for Adaptable Terminology Acquisition

Farid Cerbah[1] and Béatrice Daille[2]

[1] DPR/ESA, Dassault Aviation – 78, quai Marcel Dassault
92552 Saint-Cloud – France
farid.cerbah@dassault-aviation.fr

[2] Université de Nantes - LINA, 2 rue de la Houssinière - BP 92208,
44322 Nantes cedex 03, France
beatrice.daille@univ-nantes.fr

Abstract. Terminology acquisition has proven to be useful in a large panel of applications, such as Information Retrieval. Robust tools have been developed to support these corpus-based acquisition processes. However, practitioners in this field cannot yet benefit from reference architectures that may greatly help to build large-scale applications. The work described in this paper shows how an open architecture can be designed using web services technology. This architecture is implemented in the HyperTerm acquisition platform. We show how it has been used for coupling HyperTerm and ACABIT term extractor.

1 Introduction

It is widely recognized in the NLP community that terminology complication is about to reach the stage of a mature technology [1,2]. Over the last decade, robust methods and tools have been developed to provide a significant computational support for well-understood processes, as terminology extraction and structuring, variant detections and term alignment from parallel corpora. Surprisingly, this maturity has not really encouraged researchers to take a software engineering perspective over these terminology acquisition processes. Practitioners in this field cannot yet benefit from reference architectures that may greatly help to build up large-scale applications.

In this study, terminology acquisition is conceived as an extended application intended to end-users that are not familiar with corpus manipulation techniques. A large-scale application is often realized through a complex combination of heterogeneous tasks where the core steps of terminology analysis are embedded within a number of complementary tasks for corpus preprocessing, data management and validation. It is only within such a globalized conception of this acquisition process that task integration and architectural issues are perceived as critical. Additionally, even though many of the involved processing steps are highly generic, the terminological parts often need to be adapted from an application to another. We show how the *web services* technology has been used to elaborate a structuring framework that ease the development of terminology

Z. Kedad et al.(Eds.): NLDB 2007, LNCS 4592, pp. 420–426, 2007.

processing applications through *reuse* of generic components and *adaptation* of domain-dependent components.

2 The Core of Terminology Extraction

Identifying terms of a specific domain has been an active field of research since the early nineties [1,2] which has lead a few years after to a set of methods and tools determined by the need of various applications: information retrieval, information extraction, science and technology watch, and ontology acquisition. The various methods involve on the input texts various processing tasks, from tokenization to syntactic analysis, and on the output term candidates, manual validation process through dedicated interfaces. Whatever the sophistication of the input or output processing steps, all methods include the following two steps that make up the core of the extraction process: (1) Identification and collect of term-like units in the texts (mostly multi words phrases) and (2) Filtering of the extracted term-like units to keep the most representative of the specialized field and the most useful for the target application.

Despite a high-level similarity, the methods may differ on several aspects:

– The complexity of the terminological unit, usually of the noun phrase, that is accepted: single noun, short noun phrase, maximal noun phrase. Considering the text sequence *low-temperature grain neutron diffraction*, some tools will keep only the single term *diffraction* while others will extract the multi-word term *neutron diffraction*, or even the complete sequence.

– The variability of the linguistic form of the multi-word term, and the degree of variability that is accepted: orthographical, morphological, syntactical or semantic variants. Some tools would consider the French occurrences *échange d'ions* (Lit. 'exchange of ions') and *échange ionique* (Lit. 'ionic exchange') as the same term while others would keep them separate [3].

– The organization of the terms using semantic relations or clustering methods [4].

– The filtering mechanisms involved to accurately eliminate irrelevant candidate terms using statistical measures or predefined lexical sources, such as existing terminological resources and stop-term lists.

Term extraction is close to maturity, and a number of tools are now available to efficiently support this data intensive process. Acabit described in [5], is one of the most representative tools in this domain.

3 The HyperTerm Platform

The HyperTerm platform which is redesigned in this work provides an extended support for the basic terminology acquisition and management processes. The core of HyperTerm ensures the storage and management of the large data collections manipulated in this extended process, and supervises the execution and sequencing of the processing steps. A number of full-fledged components built on top of this kernel support the main steps on the term acquisition process:

– **The Document Manager:** This component is dedicated to preprocessing and indexing of heterogeneous corpora. It can handle various document formats and parses the documents to make the content ready for linguistic parsing.

– **Part of Speech Tagger:** The first linguistic processing step is part of speech tagging. We developed a rule based tagger called MultAna on top of MMORPH morphological parser [6]. The MMORPH parser assigns to each word its possible morphological descriptions by looking up in lexicons and applying morphological decomposition rules.

– **The Term Extractor:** The extraction component implements a term identification technique based on morpho-syntactic patterns which define a local grammar of compound terms. For sake of efficiency, the patterns are encoded in finite state automata which are applied on input sequences of tagged words. The extractor also identifies hierarchical relations between terms using lexical overlap rules, and the resulting relations are exploited to assign a global structure to the extracted terminology.

– **The Data Explorer:** The data explorer component of HyperTerm provides many functions to support filtering, management and validation operations that are required during the final manual stage of terminology construction. Term validation is facilitated by the browsing mechanisms that allow the user to quickly put back the extracted terms in their document contexts.

HyperTerm can be seen as one of the possible answers to a recurring need: To provide software support to the generic process of terminology acquisition. However, the confrontation of this system with specialized corpora from different technical domains quickly reveals the inadequacy of such a ”one-for-all” solution to the problem. More concretely, HyperTerm has been originally designed to deal with corpora from the aeronautic domain, and the extraction steps have been optimized for that type of corpora. We noticed that experiments conducted outside of that privileged domain were less conclusive. Even if the extraction methods are often defined independently of any domain, experiences of terminology acquisition supported by extraction tools show that further adaptation is needed to deal with the linguistic specificities and term formation rules of each domain. For example, in the medicine domain, many neoclassical compound nouns and adjectives are composed of greco-latin roots (*gastr-*, *-phage*, *-hydr-*). A root shares its part-of-speech and its meaning with the contemporary lexical entry it replaces, thus gastr- means *stomach*). A morphosemantic parser such as Dérif [7] is able to conflate synonymic variations of terms as *hepatic* and *liver*. There is a real need for a more adaptable software support that would allow an easy customization of specific extraction steps while keeping a strong ability to reuse generic services.

4 A Reference Architecture for Terminology Acquisition

We designed a service oriented architecture which is intended to meet the specific requirements of corpus-based information extraction processes. It includes a set of enabling means that facilitates the development of new processing configurations: (a) A *Service Typology* where each type of adaptable service is characterized by a precise interface definition and a base implementation, (b) A *Stratified Data Model* to ease the definition of new extraction methods.

4.1 Service Typology

We adopted a service typology conceived as an extension of *Gate* [8] component model. We distinguish three high-level types of services:

– **Stable Services:** These are basically the services that can be directly reused from an application to another. This category includes the document pre-processing, indexation and data storage services.

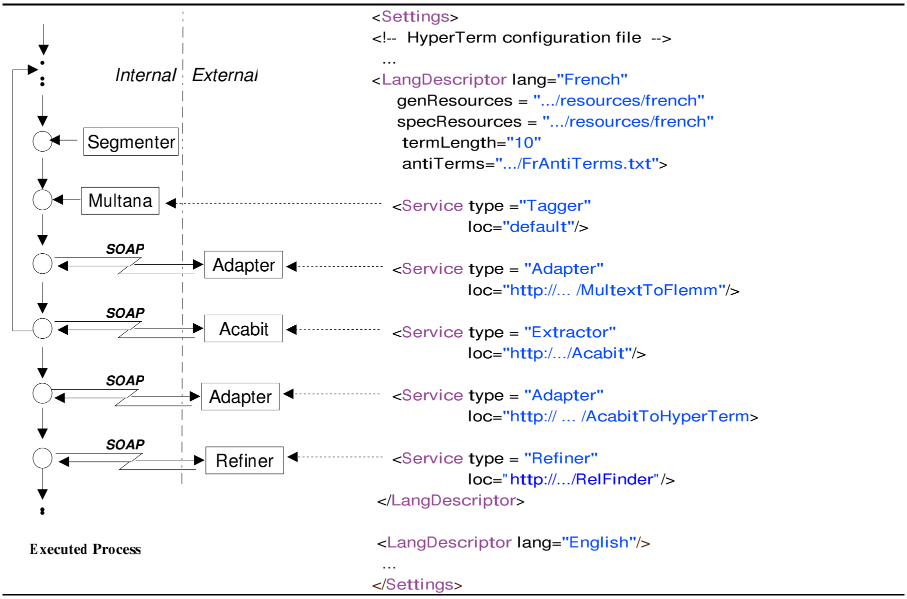
– **Instantiable Services:** This category is introduced to account for processing steps that often need to be adapted to meet the specific requirements of the application at hand. Each type of instantiable services is defined with a WSDL interface specification that can be instantiated either by an internal service implementation included in the infrastructure or by an external implementation deployed as a web service. Tagger and Extractor are two major types of instantiable services with precise interface specifications.

Most of the predefined instantiable services are associated with well-defined and recurring "'business" functions of the target application classes. In addition to these clearly identified services, we distinguish two other types of instantiable services with broader interfaces that are intended to cover diversified needs:

- **Data Adapters:** Integrating external components in processing chains often requires an adaptation of inputs and outputs. Adapters are inserted between adjacent services to ensure syntactic and semantic interoperability. For example, the tags provided by a morphological analyzer should be translated to be properly interpreted by a morpho-syntactic disambiguator that uses a different tag set.
- **Refiners:** These services are used to include additional steps to finalize the processing chains. This capability brings up flexibility to the infrastructure by allowing further refinement of the initial predefined process such as adding lexico-semantic relations or term definitions.

– **Visualization Services:** To be widely accepted by users which are not necessarily skilled in language engineering, specific GUI interfaces should be designed. The application front-end should allow an easy exploration of extracted data and documents enriched through linguistic processing. Dedicated validation views are needed to easily identify relevant terms within the large pool of extracted terms.

Figure 1 illustrates an application built following this model (see section 5).



```
<Settings>
<!--  HyperTerm configuration file  -->
 ...
<LangDescriptor lang="French"
    genResources = ".../resources/french"
    specResources = ".../resources/french"
     termLength="10"
     antiTerms=".../FrAntiTerms.txt">
      <Service type ="Tagger"
               loc="default"/>
      <Service type = "Adapter"
               loc="http://... /MultextToFlemm"/>
      <Service type = "Extractor"
               loc="http:/.../Acabit"/>
      <Service type = "Adapter"
               loc="http:// ... /AcabitToHyperTerm>
      <Service type = "Refiner"
               loc="http://.../RelFinder"/>
 </LangDescriptor>

 <LangDescriptor lang="English"/>
  ...
</Settings>
```

Fig. 1. A processing configuration set up around Acabit term extractor, invoked as an external service. Two adaptors are used to ensure interoperability between Acabit and HyperTerm internal services.

4.2 A Stratified Data Model

Integration of new services is facilitated by a common data model which allows modular design of the services. When developing a new service, a user of the infrastructure should only have to manipulate data directly related to the processing step covered by the service. A stratified data model, composed of the three layers *Indexing*, *Tagging*, and *Terminology*, is designed to fulfil this need. For example, the logic of an extractor will be defined using entities from *Terminology* and *Tagging* levels, where are formalised concepts related to terms and morphologically tagged words. However, for indexing purposes, each extracted term should be enriched with information pertaining to the *indexing* level. This is done automatically thanks to the mappings between the levels handled by the infrastructure.

5 A Case Study: Coupling HyperTerm with Acabit

The re-architectured platform has been validated with several applications from which services have been extracted to build up a first library of reusable assets. The interfacing of HyperTerm and Acabit is a significant use case that clearly illustrates the integration facilities of the platform. This interfacing helped to fill a recurring gap observed in several HyperTerm-based applications: the lack of efficient term variants detection mechanisms.

The processing configuration is illustrated in figure 1. In this configuration, Acabit is in the center of the service sequence and stable services for document processing, indexing and POS tagging are reused. Adapters are added to ensure interoperability with Acabit. More concretely, the implementation of this configuration required the following minimal changes:

– **Changes in the core of Acabit:** This is an important dimension in our evaluation of this integration framework which is primarily conceived to allow easy reuse of external services that are compliant with interface specifications of the predefined service types (see section 4.1). Acabit, as a term extractor, falls in this category of services that can be easily integrated into the processes supported by HyperTerm. The extension of Acabit was restricted to export data for the indexing services.

– **Acabit service implementation:** So, most effort has been put on adapting the interfaces. High-level functions of Acabit have been exploited to provide implementation to the WSDL interface of the extraction instantiable service.

– **Adapters:** Acabit is inserted between two data adapters (figure 1). The first one, named MultextToFlemm, maps two morphological models with different but compatible tag sets. The second adapter, AcabitToHyperTerm, is invoked once at the end of the extraction process. It exploits the result of Acabit to build a network of terms that can be interpreted by HyperTerm.

6 Conclusion

This work allowed us to validate the relevance of the service oriented approach for terminology acquisition. For that category of applications that involve intensive processing of large corpora, scaling up often requires some sacrifices with regard to the initial theoretical models. The re-architectured version of HyperTerm is fully compliant with our theoretical model and it has been validated on several applications. The performances are fair, even though significant improvement could be obtained through optimization of serialization and deserialization operations involved during invocation of external services. To improve data interoperability, we are also planning to take advantage of recent standards for linguistic data, such ISO 16642 dedicated to terminological data.

References

1. Bourigault, D., Jacquemin, C., L'Homme, M.-C.: Recent Advances in Computational Terminology. John Benjamins, Amsterdam (2001)
2. Kageura, K., Daille, B., Nakagawa, H., Chien, L.F.: Recent trends in computational terminology. Terminology 10(2), 1–25 (2004)
3. Daille, B.: Variations and application-oriented terminology engineering. Terminology 411(1), 181–197 (2005)
4. Nazarenko, A., Hamon, T. (eds): Structuration de Terminologie. vol. 43 of TAL. Hermès, Paris (2002)

5. Cabré, M.T., Bagot, R.E., Platresi, J.V.: Automatic term detection: A review of current systems. In: Bourigault, D., Jacquemin, C., L'Homme, M.-C. (eds.) Recent Advances in Computational Terminology, vol. 2, pp. 53–88. J. Benjamins, Amsterdam (2001)
6. Petitpierre, D., Russell, G.: Mmorph – The Multext Morphology Program. Technical report, Multext Deliverable 2.3.1 (1995)
7. Namer, F., Zweigenbaum, P.: Acquiring meaning for french medical terminology: contribution of morphosemantics. In: Fieschi, M., Coiera, E., Li, Y.J. (eds.) Actes, 10th Congress on Medical Informatics, San Francisco (2004)
8. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: Gate: A framework and graphical development environment for robust NLP tools and applications. In: Proceedings of ACL'02, Philadelphia (2002)

Domain Relevance on Term Weighting

Marko Brunzel[1,2] and Myra Spiliopoulou[2]

[1] DFKI GmbH - German Research Center for AI
[2] Otto-von-Guericke Universität Magdeburg, Germany
marko.brunzel@dfki.de, myra@iti.cs.uni-magdeburg.de

Abstract. The TFxIDF term weighting scheme is the standard approach on vectorization of textual data. For a data set where textual data stemming from web document structure is to be vectorized [2] the need for a enhanced term weighting scheme arose. In this publication we introduce a term weighting scheme which improves the behavior compared to the traditional TFxIDF scheme by adding a component which is based on the linguistically inspired notion of domain relevance. Domain relevance measures the degree to which a term is regarded as more relevant within a data set compared to a reference data set. By means of this external component a potential weakness of TFxIDF on non standard distributed data sets is overcome. This weighting scheme favours domain relevant terms, which can be regarded as more useful in settings where the clustering is performed to be consumed by an human supervisor e.g for semi-automatic ontology learning.

1 Introduction

The application of term weighting is often performed for processing textual data represented by the vector space model (bag of words). The most prominent weighting scheme is TFxIDF [11]. Terms which are present in many documents, and which can be regarded as not very distinguishing between instances (e.g. documents) get a lower weight than terms which are characteristic for only some instances. TFxIDF relies on the frequency and distribution within the data set itself. This is a appropriate approach for supervised learning like classification opposed to unsupervised learning like clustering. Clustering is often performed with the aim of presenting found patterns to human consumers as it is frequently performed in the field of ontology learning [8,5,13] to name only few out of many. The goal of a clustering then is usually to obtain clusters which are meaningful for the domain expert. Up to now this was done with the same term weighting scheme as it is used for classification where the separability of features is of major interest.

For the XTREEM (Xhtml Tree Mining) method [2,3] on mining sibling semantics from Web documents, we derived a vectorization which is different from the traditional vector space model [12]. In [3], we have shown by means of gold standard evaluation that this vectorization is indeed suitable mining meaningful sibling terms (co-hyponyms, co-meronyms) from semi-structured Web documents.

Z. Kedad et al.(Eds.): NLDB 2007, LNCS 4592, pp. 427–432, 2007.

For this evaluation a vocabulary stemming from the gold standard ontologies was used as feature space. In real world settings, where no manually selected feature space is available, e.g. as described in [2], the results are influenced by not desired - not domain relevant terms. In this paper, we extend the initial approach to derive an automatically calculated, gradual notion of "stopwordness", by computing domain relevance on terms. The resulting term weighting scheme is able to automatically boost domain specific terms and assign low weights to generic terms.

In this publication we will introduce a domain relevance enhanced term weighting approach. This approach is appropriate for creating clusters which should reflect domain relevant patterns. In our experiments we will perform clustering and we compare the obtained results regarding the resulting domain relevance/specificity of clusters.

2 Related Work

As related work mainly existing term weighting approaches are interesting. The *family of TFxIDF [11] term weighting schemes* eliminates "stop words" which are present in nearly every document, which is the case for stop words in regular text. This works proper on documents as instances of interest. But on other approaches on accessing textual data, e.g. in the approach for finding semantic sibling terms [2,3] from semi-structured Web documents, such stop words may occur in a fraction of instances - not wanted but also not simply eliminated by regular TFxIDF weighting.

Also relevant are the approaches for *obtaining domain relevant terms by comparing corpora* e.g. [10,6,9] and [7].

3 Domain Relevance Enhanced Term Weighting Schemes

We present the TFxIDFxDR method for weighting terms on the vector space model. Traditional term weighting only relies on the inner distribution of terms (within a data set to be processed). For supervised learning like classification, where the separability of terms/features is of major interest, this may be sufficient. But on unsupervised learning approaches like clustering, where results are often presented to a human user, one wants to present something which is highly relevant for the domain of interest. General world knowledge is likely to be of minor interest and should be automatically faded out.

3.1 The Need for Domain Relevance on Term Weighting

From our experiments on mining sibling semantics from Web documents on an open vocabulary by means of the XTREEM Group-By-Path approach [2] it became desirable to reduce the influence of non domain relevant terms on the results. Though correct with respect to being siblings, clusters such as

July, *August*, *September* and *Thursday*, *Wednesday*, *Saturday* are not very informative for the human domain ontology engineer. One could manually create a Web document and/or domain specific stop word list. This is not desirable for several reasons. (a) It is laborious and more important (b): It is not as straightforward to decide if something is an irrelevant stop word or not.

If the feature space is automatically derived by domain relevance comparison e.g. [13], a Boolean decision is taken. If a term is included in the feature space because it has passed a certain threshold or it is within the top-n most domain relevant terms, the gradually notion of domain relevance was lost. We argue to keep (or push) this information on domain relevance into the subsequent processing. The processing therefore stays unsupervised to bigger extend, though it can benefit on domain relevance information calculated before.

We argue to keep (or push) this information on domain relevance into the subsequent processing. The processing therefore stays unsupervised to bigger extend, though it can benefit on domain relevance information calculated before.

Term Weighting I - Inner Characteristic: Term Frequency - Inverse Document Frequency - TFxIDF[11]. Terms which occur frequently or rarely get a low weight compared to terms which hold a balance. This weighting is based on the assumption that terms with extreme occurrence behavior are not suited to contribute to the clustering result. For the vectors created from sibling sets, the assumption which was created for vectors of text documents is violated. Since the vectors of textual siblings (siblings grouped by a certain path structure) are created differently to traditional document vectors, TFxIDF does not work as expected. On a certain fraction of paths, also trivial terms occur. Since they may lie in the rather preferred region of the distribution after TFxIDF weighting, they are not punished as hard as they would be punished on regular text document vectors. E.g. the term "bottom" is likely to occur together with other terms when Web page authors did not use a strong structuring of the Web documents. Since this may occur on say a third of documents, TFxIDF would reward such a term, against our expectation.

Term Weighting II - Outer Characteristic: Domain Relevance/ Specificity - DR. The usage of contrastive corpora has a long tradition in corpus linguistics [10,9,6]. It is frequently applied as in [14,4]. One can compare the occurrence characteristics in an analysis corpus (domain corpus) with the occurrence characteristics in a reference corpus (general language corpus). This can simply be the frequency of words/terms or those of other patterns such as bi-grams [6]. The terminology "corpus" and "document collection" are used synonymously in this publication, as "domain relevance" and "domain specificity". For our purpose, we want to calculate a value on domain relevance. It should reflect the extend to which a term is characteristic for a domain corpus compared to other terms of this corpus. The occurrence frequency is the primary object of comparison.

In the following RC refers to the reference corpus, AC to the analysis corpus.

DR$_{freq}$(t)= $\frac{\frac{f_{RC}(t)}{F_{RC}}}{\frac{f_{AC}(t)}{F_{AC}}}$ whereas f(t) depicts the number of occurrences of a term and F is the sum of all frequency counts (the size of the corpus in terms). Since the whole number of terms is not known for the Web document collection, we propose/allow to use the sum of counts for the terms involved.

4 Experiments and Evaluation

4.1 Evaluation Methodology

The aim of incorporating domain relevance in term weighting is to obtain clusters which are described by domain relevant terms to a bigger extend than without a DR incorporating term weighting. We multiply the share a certain term occurs within a cluster (relative inner cluster frequency) with a calculated value of domain relevance/specificity. We do so for all terms of a cluster, and for all clusters. The resulting sum value (DR_{Sum}) reflects how much the cluster characteristics of a clustering are influenced on domain relevant/specific terms.

For term weighting there exist two variants, one where the vectors are normalized to unit length and term weighting without normalization to unit length. In our experiments we will consider both variants.

4.2 Description of Experimental Influences

We have applied our experiments on a data set which is different to the traditional bag of words vector space model. This data set is based on the processing facility described in [2]. The core operation of this XTREEM approach is described in more detail in [3]. The establishment of the document collection is the first task of the XTREEM procedure. The seed consisted of the keywords "Semantic Web", "Ontology" and "Ontologies". We used the Google Search API. The result was a set of 4209 distinct URLs (October 2004), from which we retrieved 4015 Web Documents from 2112 domains. From these, we have removed approximately 10 percent documents that were recognized as non-English language documents. According to the pre-processing tasks of XTREEM, the Web documents have been converted to XHTML and the frequencies of text elements over the whole document collections have been counted. We have chosen the 1000 most frequent text elements as features. The Group-by-Path algorithm has processed 22462 document paths. For the purpose of finding siblings, at least two non-zero values per vector are required, resulting in 7713 vectors retained so far. For the number of clusters to be generated by the K-Means clustering algorithm, we set K=100 (a heuristically chosen value which is feasible for 1000 features). Since the result of a K-Means clustering is dependent on the seed centroids, we will perform each clustering 10 times with different randomly chosen seed centroids. The DR_{Sum} values will be averaged over these 10 runs. For calculation of domain relevance score we used the British National Corpus (BNC) [1] as reference for the general language.

4.3 Results

Experiment 1: DR_{Sum} with Unit Length Normalization. In the first experiment we conducted the processing with unit length normalization. Table 1 shows the results. The combination of domain relevance and inverse document frequency gave the highest DR_{Sum} score, followed by the solely domain relevance term weighting. The results for clusterings where domain relevance enhanced term weighting was applied shielded the highest scores.

Table 1. DR_{Sum} for Term Weighting with Unit Length Normalization

Term Weighting approach:	no weighting	IDF	DR	DRxIDF
DR_{Sum} Score:	88,041	91,198	101,802	109,286

For the results obtained while performing unit length normalization there is a again a clear trend: term weighting incorporating DR is always better than term weighting not incorporating DR.

Experiment 2: DR_{Sum} without Unit Length Normalization. In the second experiment we conducted the processing without unit length normalization. The results are shown in table 2. Also for this experiment, the domain relevance enhanced term weighting shielded the highest DR_{Sum} score.

Table 2. DR_{Sum} for Term Weighting without Unit Length Normalization

Term Weighting approach:	no weighting	IDF	DR	DRxIDF
DR_{Sum} Score:	94,783	113,899	155,902	163,433

4.4 Conclusion of Evaluation

With DR_{Sum} we wanted to measure the contribution of domain relevant terms on the results of a clustering.

Our experiments support our hypothesis that IDFxDR term weighting can be regarded as leading to better results regarding creating clusters where the terms of this clusters are more domain relevant than for clusters created upon traditional term weighting.

5 Conclusions and Future Work

We have presented our term weighting approach which combines two widespread approaches (TFxIDF and DR) into one. Former external (pre-processing) knowledge, which is used to select a domain specific vocabulary, is forwarded inside the processing facility. This is especially appropriate when human interaction is only desired at the end of the processing. DR enhanced term weighting brings indeed domain relevant terms to the top labeling features of a cluster. The differences are not that large, but for human consumption even small improvements are desirable.

Acknowledgements. Parts of this work have been supported by European Union IST fund (Grant FP6-027705, project Nepomuk).

References

1. Aston, G., Burnard, L.: The BNC Handbook. Edinburgh University Press, Edinburgh (1998)
2. Brunzel, M., Spiliopoulou, M.: Discovering multi terms and co-hyponymy from xhtml documents with XTREEM. In: Nayak, R., Zaki, M.J. (eds.) Knowledge Discovery from XML Documents. LNCS, vol. 3915, pp. 22–32. Springer, Heidelberg (2006)
3. Brunzel, M., Spiliopoulou, M.: Discovering semantic sibling groups from web documents with XTREEM-SG. In: Staab, S., Svátek, V. (eds.) Managing Knowledge in a World of Networks. LNCS (LNAI), vol. 4248, pp. 141–157. Springer, Heidelberg (2006)
4. Chung, T.M.: A corpus comparison approach for terminology extraction. Terminology 9(2), 221–246 (2003)
5. Cimiano, P., Staab, S.: Learning concept hierarchies from text with a guided agglomerative clustering algorithm. In: Biemann, C., Paas, G. (eds.) Proceedings of the ICML 2005 Workshop on Learning and Extending Lexical Ontologies with Machine Learning Methods, Bonn, Germany (August 2005)
6. Damerau, F.J.: Generating and evaluating domain-oriented multi-word terms from texts. Inf. Process. Manage. 29(4), 433–447 (1993)
7. Drouin, P.: Detection of domain specific terminology using corpora comparison. In: Proceedings of the fourth international Conference on Language Resources and Evaluation (LREC), Lisbon, Portugal (2004)
8. Faure, D., Nedellec, C.: Knowledge acquisition of predicate argument structures from technical texts using machine learning: The system asium. In: Fensel, D., Studer, R. (eds.) Knowledge Acquisition, Modeling and Management. LNCS (LNAI), vol. 1621, pp. 329–334. Springer, Heidelberg (1999)
9. Kilgarriff, A.: Comparing corpora. International Journal of Corpus Linguistics 6(1), 97–133 (2001)
10. Pierre, L.: Sur la variabiliti de la friquence des formes dans un corpus. M.O.T.S 1, 127–165 (1980)
11. Salton, G., Buckley, C.: Term weighting approaches in automatic text retrieval. Technical report, Ithaca, NY, USA (1987)
12. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. Commun. ACM 18(11), 613–620 (1975)
13. Schaal, M., Müller, R.M., Brunzel, M., Spiliopoulou, M.: Relfin - topic discovery for ontology enhancement and annotation. In: Gómez-Pérez, A., Euzenat, J. (eds.) The Semantic Web: Research and Applications. LNCS, vol. 3532, pp. 608–622. Springer, Heidelberg (2005)
14. Velardi, P., Missikoff, M., Basili, R.: Identification of relevant terms to support the construction of domain ontologies. In: Proceedings of the workshop on Human Language Technology and Knowledge Management, Morristown, NJ, USA, Association for Computational Linguistics, pp. 1–8 (2001)

Flexible and Customizable NL Representation of Requirements for ETL processes

Dimitrios Skoutas[1] and Alkis Simitsis[2]

[1] National Technical University of Athens,
Department of Electrical and Computer Engineering,
Athens, Hellas
dskoutas@dbnet.ece.ntua.gr
[2] IBM Almaden Research Center,
San Jose, California, USA
asimits@us.ibm.com

Abstract. The design of an Extract – Transform – Load (ETL) workflow for the population of a Data Warehouse is a complex and challenging procedure. In previous work, we have presented an ontology-based approach to facilitate the conceptual design of an ETL scenario. In this paper, we elaborate on this work, by investigating the application of Natural Language (NL) techniques to the ETL environment and we present a flexible and customizable template-based mechanism for generating natural language representations for the ETL process requirements and operations.

Keywords: ETL, Data Warehouses, Conceptual Model, Natural Language, Ontologies, Semantic Web, Metadata.

1 Introduction

During the initial phases of a data warehouse (DW) design and deployment, one of the main challenges is the identification of the involved sources and the determination of appropriate inter-schema mappings and transformations from the data sources to the DW. To support this procedure, specialized tools, commonly referred to as ETL tools, have already been proposed [LuVT04, TrLu03, VaSS02], while several commercial solutions already exist [IBM05, Info05, Micr05, Orac05]. However, the design part of these tools mainly focuses on the representation and modeling of the ETL processes. The identification of the required mappings and transformations is done manually, due to the lack of precise metadata, regarding the semantics of the data sources and the constraints and requirements of the DW.

In previous work, we argued that an ontology-based approach is suitable for capturing the information needed for the conceptual design of ETL processes [SkSi07a]. We addressed the issues of formally defining the semantics of the datastore schemata, by means of an appropriate application ontology, and the use of a reasoning process to infer correspondences between the sources and the DW. In this paper, we complement our previous effort having as outermost goal the use of the application ontology, as a common language, to produce a textual description of the

Z. Kedad et al.(Eds.): NLDB 2007, LNCS 4592, pp. 433–439, 2007.

requirements of an ETL process. Such descriptions in a comprehensive textual format facilitate both the communication among the involved parties and the overall process of design, implementation, maintenance, and documentation. We introduce a template-based technique to represent the semantics and the metadata of ETL processes as a narrative, based on information stored in the application ontology, which captures business requirements, documentation, and existing schemata. Our technique can be used for the customization and tailoring of reports to meet diverse information needs, as well as the grouping of related information to produce more concise and comprehensive output.

Due to space limitations, we refer the interested reader to the long version of the paper, for further details and examples [SkSi07b].

2 Related Work

Conceptual design for ETL. Several approaches exist for the conceptual part of the design of ETL scenarios [LuVT04, TrLu03, VaSS02]. However, these approaches are concerned with the graphical design and representation of ETL processes. Existing commercial solutions facilitate the design of ETL workflows without providing any method for the automatic identification of the appropriate transformations according to the semantics of the datastores involved [e.g., IBM05, Info05, Micr05, Orac05].

Ontology translation. Due to the emergence of Semantic Web, there have been some recent efforts towards the generation of textual representations from ontologies [WiJo03, Wilc03, BoWi04, Bont05]. However, these constitute general-purpose ontology verbalizers, and therefore are agnostic of the types of classes, properties, and operations used in our approach to semantically describe datastores and infer correspondences among them. Thus, in our case, the resulting output would be rather verbose and redundant, failing to focus on the aspects of interest from the perspective of the ETL design task. It would also be more difficult to customize the output and achieve different levels of granularity according to specific information needs.

Application of NL in databases. Our approach is different in that it is the first, as far as we are aware of, that employs NL techniques to ETL processes, to facilitate and clarify their design. Most of previous work deals with the generation of a well-formed model from the analysis of (un-)structured information: e.g., object oriented analysis model [IlOr05] and EER model [Buc+95, DuMe06, TjBe93, TsCY92]. Another research effort towards the application of NL techniques to facilitate the database design presents an automated database design system based on a naïve semantics ontology [StGU02]. Some of these results can be used for the construction of our global ontology from different sources. Research work regarding the validation of the model produced [MeML93, RoPr92], although orthogonal to our effort, still can be used for the validation of the outcome of our work. Results on automatic documentation [BoWi04, ReML95], ontology creation [Kof05], and data cleaning [KeMe99, KeMe02], are not directly applicable to our environment, mostly due to the existence of complex transformations (e.g., functions and aggregations) in the ETL processes, which cannot be resolved using simply linguistic techniques.

3 Using Ontologies for the Design of ETL Workflows

In this section, we briefly present previous results regarding an ontology-based conceptual design of ETL workflows, focusing only on the aspects directly related to current work. For a more detailed analysis, we refer the interested reader to [SkSi07a].

First, a graph-based representation of the datastore schemata is employed, allowing different types of sources (e.g., relational, XML) to be handled in a uniform way. An appropriate application ontology is then constructed to resolve structural and semantic conflicts among the datastores. To allow for capturing the requirements and properties of the datastores, the ontology comprises the following:

- a set of classes C_C, representing the concepts of interest in the application;
- a set of properties P_P, representing attributes or relationships between these concepts;
- a set of classes C_{TP}, used to denote the ranges of the properties in P_P;
- three sets of classes, denoted by C_{TF}, C_{TR} and C_{TE}, used to represent, respectively, different representation formats or units of measurements for the values of an attribute (e.g., different currencies), value intervals and sets of distinct values;
- a property convertsTo, used to relate classes in C_{TF}, indicating that a conversion from one format to another is possible;
- finally, a set of classes C_G and C_{TG} are used to denote, respectively, aggregation functions (e.g., average, sum, count, max) and attribute values resulting from aggregation operations (e.g., average age, total cost).

The application ontology is represented as a graph, with a different visual notation for each type of classes and properties, to facilitate the tasks of creating, viewing and maintaining the ontology, as well as annotating the datastores. The datastores are annotated by specifying mappings from each datastore graph to the ontology graph. Based on these mappings, a defined class is automatically constructed and inserted in the ontology for each element in the datastore schema. The definition consists of a set of *property restrictions* applied on a primitive class, which are determined by the mappings. There are two types of property restrictions: *value constraints*, which have the form $\forall P.C$ and restrict the range of the property when applied to a particular class, and *cardinality constraints*, which have one of the forms $=nP$, $\geq nP$ or $\leq nP$ and restrict the number of values the property can take. A class C used in a value restriction $\forall P.C$, may belong to one of the sets C_{TF}, C_{TR}, C_{TE}, or C_{TG}, indicating, respectively, that the values of this property have either a particular representation format or belong to a specified interval or set of values or result from an aggregation.

Based on the application ontology and the annotated datastore graphs, automated reasoning techniques infer correspondences and conflicts among the datastores, and identify relevant sources and propose conceptual operations for the population of DW.

4 Generating Reports for ETL Workflows Description

To derive reports for describing the datastores, transformations, and inter-attribute mappings involved in an ETL process, we exploit the structured nature of the ontology

Table 1. A set of provided built-in functions

Function	Output
HEAD(D)	The primitive class appearing in the definition D
PARSE_DEF(D)	The list of restrictions R appearing in the definition D
PARSE_RES(R)	The type of restriction R (see section 3)
TEXT(X)	The textual description of entity X
RANGE(P)	The class being the range of property P
INTERVAL(C)	The lower/upper bounds of the value interval specified by class $C \in \mathbf{C}_{TR}$
ENUM(C)	The list of members of class $C \in \mathbf{C}_{TE}$
AGGR_FUNC(C)	The class related to $C \in \mathbf{C}_{TG}$ via the property "aggregates"
AGGR_ATTR(C)	The list of classes related to $C \in \mathbf{C}_{TG}$ via the property "groups"
PARSE_FLOW(W)	The list of operations constituting a workflow W
PARSE_OP(F)	The type of operation F
PARAMS(F)	The list of parameters of an ETL operation F
SIZE(L)	The size of the list L

and translate its content into a textual form. In particular, we provide a mechanism to verbalize two types of structures:

- The *class definitions* in the ontology, which correspond to semantic descriptions of source or target schema elements. This information formally describes the characteristics of the datastores and is used to guide the construction of the ETL process, as well as to verify that the proposed mappings and operations meet the initial requirements. A textual form of this information, instead of a symbolic expression in a formal language, assists the reading and understanding of the properties of the sources and their comparison to the target requirements.
- The *generic operators* proposed as an outcome of our approach [SkSi07a]. Similarly, a textual description of the conceptual transformations required makes it easier for the involved parties to validate the design and to generate reports and documentation regarding the activities taking place.

Templates. We follow a template-based technique to exploit the well-defined structure of the ontology and the well-formed results of the reasoner, providing a comprehensive, flexible and easily customizable mechanism for generating textual representations. A template is a piece of text written using a typical template language, including variables, directives, built-in functions and macros. The text may also contain HTML tags, so that highly formatted output can be produced. Next, we present the elements of our template language.

Variables. A variable is denoted by its name preceded by the symbol $. When the template is processed by the template engine, each variable is replaced by its corresponding value.

Directives. A set of directives is provided to allow for a high degree of flexibility in specifying templates. Specifically, the directives #set, #if / #elseif / #else, and #foreach are provided to set the value of a parameter, allow conditional output and iterate through a list of objects, respectively. Notice also that the standard arithmetic, logic and comparison operators are supported.

Functions. A set of built-in functions is available to the template designer for retrieving the information of interest to include in the output (for a no finite set see Table 1). The usual arithmetic and string manipulating functions are also supported.

Macros. Macros facilitate the creation of templates, by allowing simpler templates to be reused, customized, extended or even combined with each other to create ones that are more complex. A macro is defined for each type of restriction occurring in the definition of a class, as well as for each different type of ETL operator, to specify the textual description of that element. Another common use for macros is to specify how the elements of a list should be rendered.

An example template for rendering the definition of a class is structured as follows:

```
Template: PRINT_DEF(D)
#set ($head=#HEAD($D)); #set ($res_list=#PARSE_DEF($D));
#HEADER( $head );
#foreach( $res in $res_list)
 #if (#PARSE_RES($res)==“EXACT_CARD”) #EXACT_CARD($res);
 #elseif (#PARSE_RES($res)==“MIN_CARD”) #MIN_CARD($res);
   ...
#end
```

where HEAD, PARSE_DEF and PARSE_RES are provided built-in functions, while HEADER, EXACT_CARD and MIN_CARD are macros for rendering, the head of the definition, and exact and minimum cardinality restrictions. For example, given the definition of a sample datastore containing information about products:

```
DS_Product ≡ Product ⊓ =1hasProductID ⊓ ≥1suppliedBy ⊓
    =1hasPrice ⊓ ∀hasPrice.Dollars ⊓ =1belongsTo ⊓
    ∀belongsTo.{software,hardware,accessories}
```

The above template generates the following output:

> *Each record in this store contains information about a Product. It has exactly 1 product id. It is supplied by at least 1 supplier. It has exactly 1 price. It has price of type dollars. It belongs to exactly 1 category. It belongs to category one of: software, hardware, accessories.*

With more elaborated templates and built-in functions that group together elements of the same type, more concise outputs may be generated:

> *...It has exactly 1 product id, has exactly 1 price, of type dollars, and belongs to exactly 1 category...*

Other typical cases are, for instance, to verbalize the first n operations of the workflow or the operations concerning a specific property. The later case is especially useful to track the transformations occurring on a specific attribute throughout the workflow. Another practical case is to list groups of order-equivalent transformations to help the administrator to design the execution order of an ETL workflow [Simi05].

5 Conclusions and Future Work

In this paper, we have investigated the application of NL techniques to the ETL environment. We have presented a comprehensive, flexible, and easily customizable template-based mechanism for generating textual descriptions of the requirements of an ETL process and documenting automatically both the semantics of the datastores and the generic conceptual operations required to compose an ETL design.

As far as future work is concerned, we already have some preliminary results on using our ontology-based framework to deal with the evolution of ETL designs.

References

[Bont05] Bontcheva, K.: Generating Tailored Textual Summaries from Ontologies. In: Gómez-Pérez, A., Euzenat, J. (eds.) The Semantic Web: Research and Applications. LNCS, vol. 3532, Springer, Heidelberg (2005)

[BoWi04] Bontcheva, K., Wilks, Y.: Automatic Report Generation from Ontologies: The MIAKT Approach. In: Meziane, F., Métais, E. (eds.) Natural Language Processing and Information Systems. LNCS, vol. 3136, Springer, Heidelberg (2004)

[Buc+95] Buchholz, E., Cyriaks, H., Düsterhöft, A., Mehlan, H., Thalheim, B.: Acquiring Complex Information from Natural Language for EER Database Design. In: NLDB (1995)

[DaHo93] Dalianis, H., Hovy, E.H.: Aggregation in Natural Language Generation. In: EWNLG (1993)

[DuMe06] Du, S., Metzler, D.P.: An Automated Multi-component Approach to Extracting Entity Relationships from Database Requirement Specification Documents. In: Kop, C., Fliedl, G., Mayr, H.C., Métais, E. (eds.) Natural Language Processing and Information Systems. LNCS, vol. 3999, Springer, Heidelberg (2006)

[IBM05] IBM. IBM WebSphere DataStage

[IlOr05] Ilieva, M.G., Ormandjieva, O.: Automatic Transition of Natural Language Software Requirements Specification into Formal Presentation. In: Montoyo, A., Mu̕noz, R., Métais, E. (eds.) Natural Language Processing and Information Systems. LNCS, vol. 3513, Springer, Heidelberg (2005)

[Info05] Informatica. PowerCenter

[KeMe99] Kedad, Z., Métais, E.: Dealing with Semantic Heterogeneity During Data Integration. In: Akoka, J., Bouzeghoub, M., Comyn-Wattiau, I., Métais, E. (eds.) Conceptual Modeling ER'99. LNCS, vol. 1728, Springer, Heidelberg (1999)

[KeMe02] Kedad, Z., Métais, E.: Ontology-Based Data Cleaning. In: Andersson, B., Bergholtz, M., Johannesson, P. (eds.) Natural Language Processing and Information Systems. LNCS, vol. 2553, Springer, Heidelberg (2002)

[Kof05] Kof, L.: Natural Language Processing: Mature Enough for Requirements Documents Analysis? In: Montoyo, A., Mu̕noz, R., Métais, E. (eds.) Natural Language Processing and Information Systems. LNCS, vol. 3513, Springer, Heidelberg (2005)

[LuVT04] Luján-Mora, S., Vassiliadis, P., Trujillo, J.: Data Mapping Diagrams for Data Warehouse Design with UML. In: Atzeni, P., Chu, W., Lu, H., Zhou, S., Ling, T.-W. (eds.) Conceptual Modeling – ER 2004. LNCS, vol. 3288, Springer, Heidelberg (2004)

[MeML93] Metais, E., Meunier, J., Levreau, G.: Database Schema Design: A Perspective from Natural Language Techniques to Validation and View Integration. In: ER (1993)

[Micr05] Microsoft. Data Transformation Services

[Orac05] Oracle. Oracle Warehouse Builder Product Page

[ReMe99] Reape, M., Mellish, C.: Just what is aggregation anyway. In: ENLG (1999)

[ReML95] Reiter, E., Mellish, C., Levine, J.: Automatic generation of technical documentation. In: Applied Artificial Intelligence 9(3) (1995)

[RoPr92] Rolland, C., Proix, C.: A Natural Language Approach for Requirements Engineering. In: Loucopoulos, P. (ed.) Advanced Information Systems Engineering. LNCS, vol. 593, Springer, Heidelberg (1992)

[Simi05] Simitsis, A.: Mapping Conceptual to Logical Models for ETL Processes. In: DOLAP (2005)

[SkSi07a] Skoutas, D., Simitsis, A.: Ontology-based Conceptual Design of ETL Processes for both Structured and Semi-structured Data. In: IJSWIS, 2007 (to appear)

[SkSi07b] Skoutas, D., Simitsis, A.: Flexible and Customizable NL Representation of Requirements for ETL Processes. Technical Report. http://www.dblab.ntua.gr/~asimi/publications/SkSi07b.pdf

[StGU02] Storey, V. C., Goldstein, R. C., Ullrich, H.: Naive Semantics to Support Automated Database Design. In: IEEE TKDE, vol. 14(1) (2002)

[TjBe93] Min, T.A., Berger, L.: Transformation of Requirement Specifications Expressed in Natural Language into an EER Model. In: ER (1993)

[TrLu03] Trujillo, J., Lujan-Mora, S.: A UML Based Approach for Modeling ETL Processes in Data Warehouses. In: ER (2003)

[TsCY92] Tsen, F.S.C., Chen, A.L.P., Yang, W.-P.: On mapping natural language constructs into relational algebra through E-R representation. In: DKE (9) (1992)

[VaSS02] Vassiliadis, P., Simitsis, A., Skiadopoulos, S.: Conceptual Modeling for ETL Processes. In: DOLAP (2002)

[WiJo03] Wilcock, G., Jokinen, K.: Generating Responses and Explanations from RDF/XML and DAML+OIL. In: IJCAI (2003)

[Wilc03] Wilcock, G.: Talking OWLs: Towards an Ontology Verbalizer. In: ISWC (2003)

Author Index

Aboutajdine, Driss 107
Andres, Frederic 272
Aoughlis, Farida 341
Araki, Kenji 364, 388
Ataa Allah, Fadoua 107

Berrut, Catherine 240
Bolshakov, Igor A. 414
Brunzel, Marko 427

Cerbah, Farid 420
Chanona-Hernandez, Liliana 401
Chen, Xiaoying 264
Chen, Yuquan 96, 264
Chevallet, Jean Pierre 240
Çıltık, Ali 35
Conesa, Jordi 143
Cordy, James R. 61

Daille, Béatrice 420
Ding, Yihong 131
Duan, Jianyong 264

Embley, David W. 131

Fernandez Gavilanes, Milagros 395
Ferrández, Antonio 352
Ferrández, Óscar 284, 352
Ferrández, Sergio 352
Fliedl, Günther 156
Fortier, Randy J. 12
Frost, Richard A. 12
Fu, Yan 73

Gao, Aiqiang 73
Gelbukh, Alexander 401, 414
Grosky, William I. 107
Güngör, Tunga 35

Hepp, Martin 131
Hindriks, Koen V. 204
Hoppenbrouwers, Stijn 204
Hu, Yi 96, 264

Ibekwe-SanJuan, Fidelia 252
Immaneni, Trivikram 119

Jonker, Catholijn M. 204

Kawtrakul, Asanee 272
Kiyavitskaya, Nadzeya 61
Kof, Leonid 181
Kop, Christian 156

Labadié, Alexandre 295
Lagji, Klara 407
Laukaitis, Algirdas 193
Laukaitis, Ricardas 193
Li, Jinhui 25
Li, Peifeng 25
Li, Qiangqiang 168
Lilleng, Jeanine 229
Liu, Hui 96
Lonsdale, Deryle 131
Lu, Ruzhan 96, 264

Maisonnasse, Loic 240
Mesfar, Slim 305
Mich, Luisa 61
Micol, Daniel 284
Miura, Takao 84
Molina, Antonio 382
Montero, Calkin S. 388
Muñoz, Rafael 284, 352
Mylopoulos, John 61

Oakes, Michael P. 217

Palomar, Manuel 284
Peng, Jing 364
Përnaska, Remzi 407
Piton, Odile 407
Pla, Ferran 382
Ponomareva, Natalia 382
Prince, Violaine 295

Rilling, Juergen 168
Role, François 395
Rosso, Paolo 382

SanJuan, Eric 252
Schumacher, Kinga 317
Shaik, Mastan Vali 119
Sidorov, Grigori 401
Silberztein, Max 1

Simitsis, Alkis 433
Skoutas, Dimitrios 433
Spiliopoulou, Myra 427
Storey, Veda C. 143
Sugumaran, Vijayan 143

Tang, Shiwei 73
Thirunarayan, Krishnaprasad 119
Tomassen, Stein L. 229
Toral, Antonio 352
Torres-Moreno, Juan-Manuel 252
Tykhonov, Dmytro 204

Váradi, Tamás 376
Velázquez-Morales, Patricia 252
Vilares, Jesús 217
Vilares, Manuel 217
Villemonte de la Clergerie, Éric 395
Vöhringer, Jürgen 156

Wakabayashi, Kei 84
Wang, Gang 329
Wang, Haofen 48, 329
Wang, Tengjiao 73
Wang, Yang 48
Witte, René 168

Xu, Li 131

Yang, Dongqing 73
Yingsaeree, Chaiyakorn 272
Yu, Yong 48, 329

Zeni, Nicola 61
Zhang, Dongyi 96
Zhang, Huajie 329
Zhang, Yonggang 168
Zhu, Haiping 48
Zhu, Qiaoming 25