Proceedings of the International Workshop on Text Mining Research, Practice and Opportunities

1st Workshop on
Text Mining Research, Practice and Opportunities

held in conjunction with RANLP 2005
Borovets - Bulgaria
24th of September 2005

It is a well known fact that huge quantities of valuable knowledge are embedded in unstructured texts that can be found in the World Wide Web, in intranets and on personal desktop machines. In recent years, there has been an increasing research interest in technologies for extracting and analysing useful structured knowledge from unstructured texts. At the same time, there has been an increasing commercial interest with a number of tools appearing on the market that address the needs of the users to some extent.

The workshop discusses recent advances regarding the research into text mining approaches that combine research from machine learning, text mining, natural language processing, information extraction, information retrieval and ontology learning. Emphasis is given to approaches that emphasize the complete document lifecycle i.e., from document collection to document archiving and analysis. At the same time, the workshop discusses the deployment of text mining technology within everyday business problems. Emphasis is given on reports that discuss the implementation of text mining projects that have lead to significant and measurable improvements in business operations or some other equally important benefit to society such as an important scientific discovery. Finally, the workshop discusses the lessons learned from the deployment of text mining technology and the opportunities that appear on the horizon as future challenges for the research community.

The workshop is organized around 12 paper presentations and an invited talk by Professor Alessandro Zanasi.

The Programme Committee that helped with the review process consists of the following:

Bill Black, University of Manchester, UK
Hamish Cunningham, Univ. of Sheffield, UK
Robert Dale, Macquarie Univ., Australia
Ronen Feldman, Bar-Ilan Univ., Israel
Gregory Grefenstette, CEA, France
Olivier Jouve, SPSS, USA
Aaron Kaplan, Xerox, France
Ian Lewin, Computer Laboratory, Univ.of Cambridge, UK
David Milward, Liguamatics, UK
Dr. Leonel Ruiz Miyares, Centro de Linguistica Aplicada, Santiago, Cuba
Andreas Persidis, Biovista, Greece
Stelios Piperidis, ISLP, Athens, Greece
Marie-Laure Reinberger, Univ.of Antwerp, Belgium
Costas Spyropoulos, NCSR, Greece
John Tait, Univ.of Sunderland, UK
Claire Thie, QinetiQ, UK
Christos Tsalidis, Neurosoft, Greece
Alessandro Zanasi, TEMIS, Italy

The workshop organiser and editor of the proceedings is Babis Theodoulidis, University of Manchester (babis.theodoulidis@manchester.ac.uk)

1st Workshop on
Text Mining Research, Practice and Opportunities

held in conjunction with RANLP 2005
Borovets - Bulgaria
24th of September 2005

It is a well known fact that huge quantities of valuable knowledge are embedded in unstructured texts that can be found in the World Wide Web, in intranets and on personal desktop machines. In recent years, there has been an increasing research interest in technologies for extracting and analysing useful structured knowledge from unstructured texts. At the same time, there has been an increasing commercial interest with a number of tools appearing on the market that address the needs of the users to some extent.

The workshop discusses recent advances regarding the research into text mining approaches that combine research from machine learning, text mining, natural language processing, information extraction, information retrieval and ontology learning. Emphasis is given to approaches that emphasize the complete document lifecycle i.e., from document collection to document archiving and analysis. At the same time, the workshop discusses the deployment of text mining technology within everyday business problems. Emphasis is given on reports that discuss the implementation of text mining projects that have lead to significant and measurable improvements in business operations or some other equally important benefit to society such as an important scientific discovery. Finally, the workshop discusses the lessons learned from the deployment of text mining technology and the opportunities that appear on the horizon as future challenges for the research community.

The workshop is organized around 12 paper presentations and an invited talk by Professor Alessandro Zanasi.

The Programme Committee that helped with the review process consists of the following:

Bill Black, University of Manchester, UK
Hamish Cunningham, Univ. of Sheffield, UK
Robert Dale, Macquarie Univ., Australia
Ronen Feldman, Bar-Ilan Univ., Israel
Gregory Grefenstette, CEA, France
Olivier Jouve, SPSS, USA
Aaron Kaplan, Xerox, France
Ian Lewin, Univ.of Cambridge, UK
David Milward, Liguamatics, UK
Dr. Leonel Ruiz Miyares,
Centro de Linguistica Aplicada, Cuba
Andreas Persidis, Biovista, Greece
Stelios Piperidis, ISLP, Athens, Greece
Marie-Laure Reinberger, Univ. Antwerp, Belgium
Costas Spyropoulos, NCSR, Greece
John Tait, Univ.of Sunderland, UK
Claire Thie, QinetiQ, UK
Christos Tsalidis, Neurosoft, Greece
Alessandro Zanasi, TEMIS, Italy

Babis Theodoulidis,
University of Manchester (babis.theodoulidis@manchester.ac.uk)

Table of Contents

Invited Talk: Intelligence Analysis through Text Mining
Professor Alessandro Zanassi

Unsupervised text mining for designing a virtual web environment
Marie-Laure Reinberger

OLE — A New Ontology Learning Platform
Vít Nováček and Pavel Smrz

Mining Personal Data Collections to Discover Categories and Category Labels
Shih-Wen Ke, Michael P. Oakes, Chris Bowerman

Information Management: The Parmenides Approach
Alexander Mikroyannidis, Apostolos Mantes and Christos Tsalidis

Semantic Indexing: Cognitive Maps Based Approach
Vladimir F. Khoroshevsky

Text Mining Software Survey
Haralampos Karanikas and Thomas Mavroudakis

From likelihoodness between words to the finding of functional profile for ortholog genes
Natalia Grabar, Magali Sillam, Marie-Christine Jaulent, Céline Lefebvre, Édouard Henrion, Christian Néri

Uncovering Terrorist Networks in Parmenides
Nikolaos Lazaridis

Using Hearst's Rules for the Automatic Acquisition of Hyponyms for Mining a Pharmaceutical Corpus
Michael P. Oakes

A Comparison Between a Rule-Based and a TBL-Based Approach for Temporal Element Extraction
Argyrios Vasilakopoulos and William J. Black

Textual Information Extraction Using Structure Induction
Alexandre S. Saidi

Learning sure-fire rules for Named Entity Recognition
Enrique Alfonseca and Maria Ruiz-Casado

Uncovering Terrorist Networks in Parmenides

Nikolaos Lazaridis

School of Informatics, University of Manchester,
Sackville Street, Manchester M60 1QD, United Kingdom
N.Lazaridis@postgrad.manchester.ac.uk

Abstract

Recent advances in the Semantic Web technologies and the abundance of highly heterogeneous and disparate information on the web opens a new path to the analysis of terrorist networks. Numerous systems have been collaboratively developed between agencies and academia applying different metaphors in order to untangle terrorist activities. Among them, the network metaphor is the most popular, and social network analysis, visualization and data mining methods are employed to assist analysts in the discovery of structural characteristics and patterns of interaction. Although, Natural Language Processing and Information Extraction have matured to the point where the conversion of freeform documents to these diagrams can be largely automated, existing approaches provide a restrictive view of the entities and relations using simplified statistical approaches. In addition, constructed networks are considered static, thus inhibiting the discovery of temporal patterns. This paper focuses on the network construction process by utilizing advanced NLP and IE techniques. The Parmenides framework proposed a set of techniques and a methodology to the consistent and efficient gathering, organization and dissemination of relevant information towards the effective construction and analysis of terrorist networks.

1 Introduction

Intelligence analysis is considered a highly complex and time consuming process in the fight against terrorism. Recent advances in Semantic Web and Information Extraction technologies have enabled the exploitation of the vast amounts of heterogeneous data sources. This offers analysts the ability to share information and extract useful knowledge that may have gone unseen.

Traditional methodologies for intelligent analysis involved ad hoc databases for criminal activity tracking, often created around the investigator's needs ignoring the advantages of collaboration and information sharing. Being able to analyse all the available information requires a methodology and technologies that allow analysts to efficiently extract relevant objects of interest maintain them in a consistent manner in order to enhance the building of scenarios that will unravel the inner workings of the terrorist networks.

Social Network Analysis (SNA), Visualization and Data Mining (DM) are the techniques employed most often to analyze the structure of criminal networks and discover their structural characteristics and patterns of interaction. However, most existing systems provide simplified network construction techniques and focus on the algorithmic aspects of the network analysis. Most importantly, these techniques ignore most of the dynamic aspects of these networks, assuming that participating entities and formed links are static.

In the IST Parmenides project (IST-2001-39023), a framework for supporting the document lifecycle was developed, along with NLP and IE and Knowledge Discovery techniques for the analysis and semantic annotation as well as temporal monitoring of the domain and extracted knowledge. An XML representation facilitates the analysis by storing document metadata, allowing analysts to quickly obtain the relevant data.

The remainder of this paper is organized as follows. Section 2 presents some related work that has been carried out in the area terrorist network analysis, social network analysis using NLP and IE techniques. The proposed network construction metrics defining entity-relationships strengths are introduced in section 3. Section 4 describes the information management facilities offered in the framework to assist the analysts in performing a temporal analysis. Finally, conclusions are presented in section 5.

2 Related Work

Several applications have been developed to support criminal/terrorist network analysis each focusing on a different aspect of the problem and employing a subset of SNA, Visualization and Data Mining (DM) techniques.

A categorisation given by (Klerks 01) classifies existing tools into three generations. The first generation includes tools such as Anacapa Link Chart (Harper & Harris 75) where analysts initially identify relevant entities and

relations from raw data, in order to manually construct the association matrix. Finally, analysts can proceed to the analysis of the network representation built from this matrix.

The second generation tools provide higher automation in the network building process (most existing tools belong to this generation). The Xanalys Link Explorer[1] (Watson) provides means for the analysis of both structured and unstructured data (details regarding the extraction of these entities is not disclosed in their website). Discovered relations among the discovered entities can be displayed using multiple visualization metaphors such as timelines and networks. An award winning second generation tool is Analyst's Notebook[2] which is part of the I2 product family. It provides an environment for link and timeline analysis by allowing the automatic network creation on both structured and unstructured data. Second generation tools are characterised by a modest sophistication level with a restricted analytical functionality because they rely on the manual inspection of the graphical representations (Xu & Chen 04).

Third generation tools employ SNA techniques offering advanced analytical functionality to investigators. They employ data mining techniques to allow analysts easily discover the structural characteristics, interactions and patterns among entities. The application of SNA to criminal networks was studied by (Sparrow 91), which encountered the following problems:

- *Incompleteness* since information needed to draw a clear picture may be intentionally misleading, inaccurate, and incomplete.
- *Fuzzy boundaries* implying that it is not always clear whether a link between entities exists (temporary links may exist).
- *Dynamic Nature* of these networks since their structure changes while their trying to accomplish their goal (Krebs 01).

The CrimeNET Explorer, developed by (Xu & Chen 05), employs link analysis techniques based on modified shortest-path algorithms aiming at the identification of the strongest associations between two or more entities. Entity extraction is performed from unstructured documents using a neural network noun phrasing tool (Chau et al. 02) . Their analysis builds on the Concept Space (CS) approach proposed by (Chen & Lynch 92). Associations between entities and their strengths are computed on the basis of co-occurrence weights between terms in documents. It should be noted that this approach does not take into account semantic information when extracting terms from documents, considering the importance of a term only on the basis of its appearance frequency.

NetMiner[3], as a third generation tool, compiles a unique set of features combining SNA with advanced graph drawing techniques providing analysts with an enhanced toolset to support exploratory data analysis (EDA). Statistical analysis tools are integrated into NetMiner allowing users to analyze network data through brand-new and standard analysis routines.

Most of the above approaches ignore the dynamic nature of the networks. Carley proposed Dynamic Network Analysis (DNA) to deal with large dynamic, multi-node, multi-link networks under varying levels of uncertainty (Carley 03). DNA focuses on the evolution, change and adaptation of these networks as well as on ways these can be destabilized. The incompleteness and fuzzy boundaries problems mentioned above are taken into account in DNA, since relations are probabilistic and changes in one node can be propagated to other parts of the network. DNA, in contract to SNA, focuses on entities in terms of their roles and not just their positions. Their approach develops on the Meta-Matrix allowing the construction of a set of interlinked networks focusing on entities of interest such as people, knowledge, resources, events, tasks and organizations.

Close to our work is the work of (Sheth et al. 05) proposing semantic-oriented approaches to support the analytics of vast amount of heterogeneous data in the context of national security applications. Their work focuses on the large-scale semantic annotation of data and semi-automatic population of ontologies, proposing metrics for the semantic similarity and connectivity, as well as algorithms for the searching and ranking aiming at the identification of meaningful semantic relationships.

3 The Parmenides Framework

This section provides an overview of the document lifecycle processes supporting the extraction and consolidation of relevant entities and meaningful relationships with ultimate goal the creation of a terrorist network that reflects the pragmatic links between its members. We use the term Concept Space (CS) in our discussion to provide an outline of the essential steps in the network creation process. The CS approach generally involves the steps presented in the following subsections.

[1] http://www.xanalys.com/solutions/linkexplorer.html

[2] http://www.i2.co.uk/

[3] http://www.netminer.com/

3.1 Collection

As a first step, analysts need to locate the sources where the relevant entities need to be extracted from. However, this is not a simple task and analysts are overwhelmed by the vast amount of heterogeneous and fragmentary data existing in either online (web pages, emails etc) or in internal/classified reports. The Parmenides framework provides an agent-based collector to assist analysts with the automation of this process, as well a high performance document warehousing architecture for the systematic classification and consolidation of the documents. In addition, since documents may appear in multiple forms there is a need to transform them into a common representation to enhance the further processing. All documents stored in the warehouse adhere to the *Common Annotation Scheme* (CAS) (Rinaldi et al. 03).

3.2 Semantic Annotation

The second step in the CS construction process is the filtering and indexing of the terms extracted from the document collection. Existing approaches rely on rather simple *Named Entity Extraction* (NEE) techniques (Chau et al. 02), which result in the retrieval of terms ignoring other semantically relevant information available in the document.

The identification of terms that correspond to the entities seems adequate for the construction of the network. However, it is evident that the extraction and maintenance of additional information regarding the characteristics of the extracted entities provides several advantages. First, since similar information may come from more than one source, it may be useful to provide means for the monitoring of the sources with respect to the quality of the information they provide (completeness, accuracy, relevancy etc). In the long term, analysts will be able to prioritize their sources as a means to minimize the information overload effect and improve the quality of the knowledge base. Second, this enriched semantic information could be utilized to assist analysts in detecting deceptive terrorist identities (Wang et al. 04) by comparing disparate semantic elements extracted from these sources.

In contrast to other approaches, where entities of interest belong to a rather restricted and predetermined set (persons, organizations, addresses, crimes etc), Parmenides provides a tight coupling of the rule-based IE system with the domain ontology (Vasilakopoulos et al. 04). The ontology-based approach followed provides extensibility with respect to the easy enrichment of the set of entities of interest.

The ontology was designed by the NLP experts in cooperation with the domain analysts using the Protégé platform. In short, the entities modelled among others include: *terrorist groups*, *organizations*, *people* (roles and positions), and *locations*.

Apart from entities, the ontology provides means to model events (Figure 1) and relations between entities of interest, which can be discovered semi-automatically using the IE system.

Figure 1: Event ontology classes

The modelling of terrorist-related events in the ontology includes victims/casualties (types and number), terrorist means (comprehensive classification of the various means of terrorist attacks), and terrorist targets (transportation, public places, persons etc).

The extraction of events/relations enhances the network construction process by providing a better understanding of the actual strengths of the links connecting two or more entities. Furthermore, analysts may proceed manually to define relationships among entities of interest (static or temporal) using the annotation editor (Figure 2).

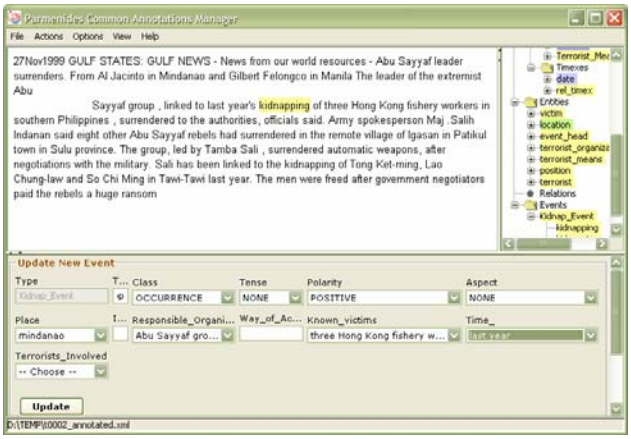


Figure 2: Event Annotation

As far as the temporal dimension of the networks (entities and relationships) is concerned, three types of temporal annotations are available through the annotation editor:

- *Temporal Expressions* (TIMEX) through annotation of text spans corresponding to temporal elements and providing their temporal values

- *Time-stamping of Events* by assigning the corresponding annotated TIMEX to the appropriate slot of the event template
- *Temporal ordering of Events* by manually creating or inferring relations among already annotated events.

Time related classes included in the ontology are loosely based on TimeML[4] and adapted to the needs of this case study.

3.3 Network Creation

The last step of the CS construction process deals with the actual computation of the strengths of the relations among the extracted entities. This process is very computationally intensive and efficient algorithms have been proposed (Ng et al. 01). They build on previous work and consider the extracted terms of equal importance computing the strength of their relationship on the basis of the frequency of their co-occurrence in the document collection.

First, the term frequency (*tf*) and document frequency (*df*) are computed representing the occurrences of the term *j* in document *i*, and the number of documents in which the term j occurs. Then, a combined weight d_{ij} is computed using the following function

$$d_{ij} = tf_{ij} \text{ x } \log\left(\frac{N}{df_j} \text{ x } w_j\right)$$

where N represents the number of documents and w_i the weight of words in the term j, since some term types are more important than others (e.g. crime types). Co-occurrence analysis is based on the asymmetric cluster function (Chen & Lynch 92), where the similarity weights from both terms are computed, and extremely frequent terms are penalized using a function similar to the inverse document frequency function.

However, this approach has several disadvantages. First, the co-occurrence of two terms in the same document does not imply a relationship between the entities they represent. For example, one term may be involved in the description of an actual event, while another term may appear in a section of the same document where historical information is provided. To this end, events extracted from the documents can be used to amplify the relations between entities. The slots of events (e.g. participating actors) are filled using a function that assigns higher scores to candidate entities that appear in close proximity to the phrase that triggers the event.

Second, low co-occurrence frequency does not necessarily imply the absence or low strength of a relationship between two entities. We should consider that terrorist networks are more vulnerable when they activate hidden links in order to obtain the resources (knowledge, training, weapons etc) they need. Therefore, it seems rational to define the weight of such relationships on the basis of the importance of the action/event that links the entities.

Third, temporal proximity in the participation of disparate events between entities can provide an indicative measure of the strength of a relationship between these entities. In detail, analysts may be able to identify otherwise hidden links when for specific time intervals the co-occurrence frequency presents notable fluctuations.

Finally, it is self evident that the simple co-occurrence weighting scheme needs several improvements, since its initial purpose was to assist users with the automatic generation of thesaurus resources. Additionally, the incorporation of semantic information in the computation of the link strength seems more appropriate in order to differentiate among relationships involving a diverse set of terrorism related events involving their temporal proximity.

4 Terrorist Network Analysis

The Parmenides framework employs temporally-oriented techniques to assist analysts with the creation and monitoring of the constructed networks, as well as the discovery of the characteristics of the actors and existing subgroups.

The analysis was performed on a corpus of 1000 documents, describing terrorist activity taking place throughout the Philippines, originating mostly from newspaper reports, ranging in date from the late 1990s to soon after the turn of the Millennium.

4.1 Metadata Queries

In the light of new evidence, analysts often need to re-examine already extracted knowledge juxtaposed to the incident documents. Therefore, it becomes imperative to provide them with links between the extracted evidence and the original documents, in order to facilitate the efficient information retrieval and analysis. Parmenides allows the formulation of queries posed on the metadata repository providing a mechanism to transparently switch between the two views (Figure 3).

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

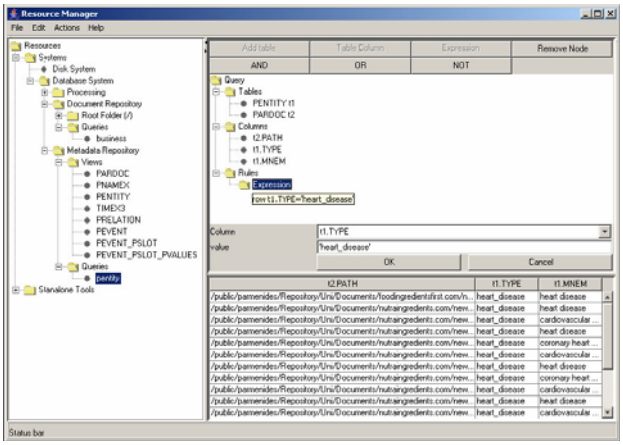


Figure 3: Querying the Metadata Repository

Taking into account the dynamic nature of the terrorist networks applying the construction technique to the entire document collection on one hand will result in unacceptable response times, while on the other hand may not reveal actors or interactions that have been enabled only for a short period of time, thus will remain hidden. Given the results of a query, analysts can proceed to the construction of a network focusing on a specific incident and/or entities. Furthermore, multiple interlinked networks can be constructed each highlighting different processes using the Meta-Matrix approach (Carley 03). This will provide analysts with insights of the conditions under which these networks are activated; hence they can be easily exposed.

4.2 Metadata Versioning

As new information becomes available, the understanding of how the network is structured develops and the main actors and possible patterns are discovered. However, the validity of these findings is subject to the quality of the information provided - usually fragmentary and intentionally incomplete - as well as to the experience of the analyst.

Analysts need to locate and examine past incidents in the document repository and update the extracted information through the manual addition/removal of temporal links, in order to run and validate scenarios. However, this may be repeated several times and previous work should be preserved to allow the monitoring of how information is revealed as well as for training purposes. In Parmenides, a versioning system allows the automatic tracking and evolution of the extracted knowledge; hence the constructed networks. The latest version of a document reflects the insights of the analysts as they have been shaped according to the latest evidence.

4.3 Ontology Evolution

Data mining techniques are utilized in the Parmenides framework to assist users with the semi-automatic building and maintenance of the domain ontology. In detail, RELFIN (Schaal et al. 05) employs clustering techniques to monitor the evolution of the entities in the document collection, producing clusters of concepts that can be used to enrich the ontology.

Furthermore, these clusters can be compared to the existing subgroups in the constructed networks, in order to make sure that information has not been ignored when networks were constructed. Additionally, the Parmenides Concept Monitor - PCM (Spiliopoulou & Baron 05) monitors the statistical metrics of the produced clusters and informs the analyst when deviations above the desired thresholds occur.

4.4 Link Discovery via Clustering

One of the primary objectives of the analysts was to be able to record relevant information on the objectives, targets and means used in past terrorist activities that would help them study the social-ideological status of the terrorist groups. This information has been modelled in the ontology and captured via rule-based NLP system.

Using this information in a series of data mining experiments analysts were able to trace and cluster[5] all recorded terrorist groups or individuals along their "profiles". Each profile is characterized by a number of items such as way of action, cultural/ideological characteristics, means used in the attacks, communicational characteristics, training, religion, nationality/language, areas of action etc. The produced clusters can assist analysts in the identification of possibly hidden links between seemingly unrelated terrorist groups and organizations.

For example, two terrorist groups have attacked similar targets, using *similar means*, in the wider area. It is also known that the first received financial support for the purchase of weapons from an organization. Given the similarity of the two groups, an analyst may hypothesize on the relationship of the second with the funding organization or investigate whether the two groups are actually the same entity.

4.5 Frequent Event Episodes

Although, it is difficult to predict and prevent the manifestation of terrorist actions, Parmenides utilizes a

[5] The EnVisioner data mining tool was used http://www.neurosoft.gr/en/products/envi.asp?actcat=products&actbul=envis

frequent episode mining algorithm similar to the one developed by (Mannila et al. 97). A framework for data pre-processing is responsible for the extraction and preparation of the accumulated event instances in the metadata repository (Figure 4).

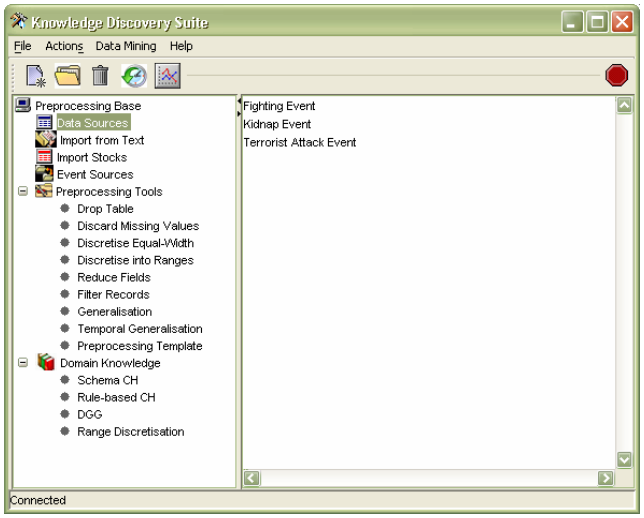


Figure 4: Data Pre-processing & Event Miner

Event data are filtered, cleaned, generalised (using ad hoc taxonomies) and transposed to the desired temporal granularity. Then analysts can utilize the produced patterns in order to make predictions for future terrorist attacks as well as build scenarios on the activation conditions (conditions when links become stronger usually when members work towards the completion of a goal) in the terrorist networks.

Furthermore, this framework allows the incorporation of external events into the analysis, offering the analysts the flexibility to combine data extracted from the repository and data existing in internal databases accumulated over the years, where the original documents are not available in electronic form.

The produced sequences of events can be used as a means for the identification of links that have gone unseen between the entities participating in them. Finally, analysts can use the patterns produced for different groups to identify similarities that may denote how one group influences the other in terms of the characteristics of their terrorist actions.

5 Conclusions

The present work investigated the area of intelligence analysis with respect to construction and monitoring of terrorist networks from an information management perspective. The proposed framework deals with the automatic collection, consolidation, and management of heterogeneous document collections under a single XML-based representation scheme. A set of improvements on existing automatic network creation process (CS) are proposed, in order to enhance the derivation of meaningful relationships among the entities in the network. Information management techniques supporting the analysis deal with the dynamic nature of the networks and provide means for the incorporation of uncertainty using a versioning and metadata querying mechanisms. Semi-automatic ontology construction and monitoring enhance the development of the analysts' understanding for the domain. Finally, temporal data mining is employed to discover frequent patterns of interaction upon which new scenarios will be build towards the creation of new insights about the structure and inner-workings of terrorist networks.

Acknowledgments:

The author is sponsored by the State Scholarship Foundation of Greece. The Parmenides project is co-funded by the European Commission (contract No. IST-2001-39023) and the project partners, as well as the Swiss Federal Office for Education and Science (BBW/OFES). Please see http://www.crim.co.umist.ac.uk/parmenides for additional information.

References:

(Carley 03) K. Carley, *Dynamic Network Analysis*, Committee on Human Factors, National Research Council, pp. 133-145, 2003.

(Chau et al. 02) M. Chau, J. Xu, and H. Chen. *Extracting meaningful entities from police narrative reports.* Proceedings of the Second National Conference on Digital Government Research, Los Angeles, CA, 20–22 May, 2002.

(Chen & Lynch 92) H. Chen, K. J. Lynch, *Automatic construction of networks of concepts characterizing document databases*, IEEE Transactions on Systems, Man and Cybernetics, 22(5), pp. 885 – 902, 1992.

(Harper & Harris 75) W. R. Harper, D. H. Harris, *The application of link analysis to police intelligence*, Human Factors, 17(2), pp. 157-164, 1975.

(Klerks 01) P. Klerks, *The network paradigm applied to criminal organizations: Theoretical nitpicking or a relevant doctrine from investigators? Recent developments in the Netherlands*, Connections, 24(3), pp. 53-65, 2001.

(Krebs 01) V. Krebs, *Mapping networks of terrorist cells*, Connections, 24(3), pp. 43 – 52, 2001.

(Mannila et al. 97) H. Mannila, H. Toivonen and A. I. Verkamo, *Discovery of frequent episodes in event sequences, Data Mining and Knowledge Discovery*, 1(3), pp. 259-289, 1997.

(Ng et al. 01) C.-Y. Ng, J. Lee, F. Cheung, B. Kao and D. W.-L. Cheung, *Efficient Algorithms for Concept Space Construction*. Proceedings of the 5th Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer-Verlag, pp. 90 – 101, 2001.

(Rinaldi et al. 03) F. Rinaldi, J. Dowdall, M. Hess, J. Ellman, G. P. Zarri, A. Persidis, L. Bernard and H. Karanikas, *Multilayer annotations in Parmenides*. Proceedings of the Workshop on Knowledge Markup and Semantic Annotation (K-CAP2003), Sanibel, Florida, USA, 2003.

(Schaal et al. 05) M. Schaal, R. Muller, M. Brunzel and M. Spiliopoulou, *RELFIN - Topic Discovery for Ontology Enhancement and Annotation*, Proceedings of the European Semantic Web Conference (ESWC 2005), Heraklion, Greece, 2005.

(Sheth et al. 05) A. Sheth, B. Aleman-Meza, I. Arpinar, C. Halaschek, C. Ramakrishnan, C. Bertram, Y. Warke, D. Avant, F. Arpinar, K. Anyanwu and K. Kochut, *Semantic Association Identification and Knowledge Discovery for National Security Applications*, Special Issue of Journal of Database Management on Database Technology for Enhancing National Security, 16(1), pp. 33-53, 2005.

(Sparrow 91) M. K. Sparrow, *The application of network analysis to criminal intelligence: An assessment of the prospects*, Social Networks, 13, pp. 251-274, 1991.

(Spiliopoulou & Baron 05) M. Spiliopoulou, S. Baron, *Temporal Evolution and Local Patterns*, Local Patterns Detection, LNAI volume 3539, Springer-Verlag, Berlin, pp. 190-206, 2005.

(Vasilakopoulos et al. 04) A. Vasilakopoulos, G. P. Zarri and K. Zervanou, *Ontology-enablement of a system for semantic annotation of digital documents*. Proceedings of the 3rd International Semantic Web Conference (2004).

(Wang et al. 04) G. Wang, H. Chen and H. Atabakhsh, *Automatically Detecting Deceptive Criminal Identities*, Communications of the ACM, 47(3), pp. 71-76, 2004.

(Xu & Chen 04) J. Xu, H. Chen, *Fighting organized crimes: using shortest-path algorithms to identify associations in criminal networks*, Decision Support Systems, 38, pp. 473-487, 2004.

(Xu & Chen 05) J. Xu, H. Chen, *Criminal Network Analysis and Visualization*, Communications of the ACM, 48(6), pp. 101-107, 2005.

Using Hearst's Rules for the Automatic Acquisition of Hyponyms for Mining a Pharmaceutical Corpus

Michael P. Oakes
School of Computing and Technology
University of Sunderland
David Goldman Informatics Centre
St. Peter's Campus, Sunderland, SR6 0DD, England
michael.oakes@sunderland.ac.uk

Abstract

Fully Automatic Thesaurus Generation (ATG) seeks to generate useful thesauri by mining a corpus of raw text. A number of statistical approaches, based on term co-occurrence, exist for this, but in general they are only able to estimate the strength of the relationship between two terms, not its nature. In this paper we implement Hearst's method of discovering the hyponymy relations which are the building blocks of hierarchical thesauri. We start with the Scrip corpus of newsfeeds in the domain of psychology, and were able to discover an estimated 400 useful term relationships.

1 Introduction

A domain-specific thesaurus such as MeSH (MEDLINE) or the Derwent Drug File (DDF) gives an overview of the extent of the domain, and the categories, relations and named entities within it. They typically consist of lists of terms organised according to a semantic hierarchy. Electronic thesauri are used in document retrieval or indexing systems, for expanding queries when searching for information or the selection of a preferred form of a given search term. Experiments such as the Worm Community System have shown that the thesaurus is an excellent memory-jogging device which supports learning and serendipitous browsing. Thesauri prevent users from becoming overwhelmed by the sheer amount of available information, and the "classical vocabulary problem, which results from the diversity of expertise and backgrounds of systems users" (Chen et al., 95).

Although a number of successful commercially-available thesauri created by large teams of human experts are available, in general manual thesaurus generation is prohibitively costly. Grefenstette (94) writes that the ideal might be to use knowledge-poor approaches, starting from just the raw corpus - "if the ultimate goal of ATG (Automatic Thesaurus Generation) is the deduction of semantic relationships exclusively from free text corpora". ATG is thus an example of knowledge discovery in text databases or text data mining.

Most existing methods for automatic thesaurus generation are statistical, and rely on the co-occurrence of a pair of terms within a common "window" of text, which may be a fixed number of words, within the same syntactic clause, or within a common document in a large collection of documents. Details of such approaches were given first by Salton in 1989, and more recently by Pereira et al. (93) and Kageura et al. (00). For each word pair in the corpus vocabulary, such methods are able to generate a numeric score to estimate semantic relatedness. However, to my knowledge, such statistical methods are unable to discriminate between different types of semantic relations, such those held in the manually-produced WordNet (Fellbaum 98) thesaurus: synonym (same as), antonym (opposite to), hyponym (is_a), meronym (part of) entailment (where one entails the other, e.g. "buy" and "pay") and troponym (the two concepts which entail each other must happen at the same time).

The "thesauri" produced by purely statistical approaches tend to be networks of interrelated terms with no sense of hierarchy, whereas the most widely-used thesauri terms for narrower concepts near

the bottom and terms for broader concepts nearer the top, with all terms connected via a single generic root concept. Sometimes the hyponymy relation is described as "parent-child", where the parent is the broader term (hypernym) and the child is the narrower or more specific term (hyponym).

2 Hearst's method

In a novel alternative approach, Hearst (92) produced an automatic lexical discovery technique that uses lexico-syntactic patterns to find instances of hyponymy relations (such as "apirin is_a drug") between noun phrases identified in a free text corpus. She used a parser to identify the noun phrases (NP) in the text, but in this paper a simpler approach was taken: entities related by the hyponymy relation consisted either of single words, or of single words preceded by "the", "a" or "an". One of Hearst's lexico-syntactic patterns is given below, where NP means a noun phrase, {}* means that the enclosed sequence may repeat any number of times, {} denotes an optional sequence, and (a|b) means either a or b may occur in the sequence at that point:

such NP as {NP,} {(or|and)} NP*
as in **… works by such authors as Herrick, Goldsmith and Shakespeare.**

When a sentence containing this pattern is found, the following hyponymy relations can be inferred:

hyponym (author, Herrick)
hyponym (author, Goldsmith)
hyponym (author, Shakespeare)

This approach has the advantage over statistical methods of determining term-term relatedness, which rely on multiple occurrences of a term pair to be in proximity with one another, in that only one occurrence of the pattern need be found for the relation to be identified (Grefenstette 94).In formulating this approach, Hearst had two main motivations: to avoid the need for pre-encoded knowledge and to produce a technique which is applicable over wide range of texts. This enables a text-mining approach whereby an ontology of hyponymic relations can be derived from a corpus of raw text. The set of lexico-syntactic patterns indicating the hyponymy (the hyponym is the narrower term) were chosen to satisfy the following desiderata:

1. They occur frequently and in many text genres
2. They (almost) always indicate the relation of interest.
3. they can be recognised with little or no pre-encoded knowledge

In this paper we use Hearst's rules to discover the homonymic relations in a free-text collection of Scrip. Scrip is an electronic daily news bulletin, distributed to those working in the pharmaceutical industry. It covers a number of topics relevant to the industry, including product launches, licensing, forthcoming meetings, personnel data, announcements by regulatory authorities, company relations and clinical trials. The data set used here consisted of all issues covering the period January to March 1999, a total of 631, 269 words including HTML mark-up. Scrip is a trademark of PJP publications Ltd. (See http://www.pjpub.co.uk).

3 Experiment on *Scrip*

Hearst's lexico-syntactic rules were adapted slightly and encoded in a program written in Perl. The set of rules used in this implementation are given below, with examples of patterns found in the Scrip corpus.

NPn = the|a|an + one word

(1.1) *NP1 such as NP2*
… diseases such as hepatitis …
→ hyponym (disease, hepatitis)

(1.2) *NP1 such as NP2 (and|or) NP3*
… cities such as Beijing and Guangzhou …
→ hyponym (cities, Beijing),
hyponym (cities, Guangzhou)

(1.3) *NP1 such as NP2, NP3 (and|or) NP4*

… infections such as bronchitis, sinusitis or pneumonia …
→ hyponym (infections, bronchitis), hyponym (infections, sinusitis), hyponym (infections, pneumonia)

(2.1) *such NP1 as NP2*

Pattern (2.1) yielded no useful matches in the Scrip corpus; two spurious matches were "dispensing such gifts as a marketing tool" → hyponym (gifts, marketing), and "such factors as tenderness" → hyponym (factor, tenderness), where "factors" was too broad a term to be useful.

(3.1) *NP1 {,} (or|and) other NP2*
… vaccines, or other injectables…
→ hyponym (injectables, vaccine)

(3.2) *NP1, NP2 {,} (or|and) other NP3*
… royalties, fees, and other revenues…
→ hyponym (revenues, royalties), hyponym (revenues, fees)

(3.3) *NP1, NP2, NP3 {,} (or|and) other NP4*
… Italy, Canada, the US and other countries …
→ hyponym (countries, Italy), hyponym (countries, Canada), hyponym (countries, US)

(4.1) *NP1 {,} {including|especially} NP2*
… cytokines, including BNF …
→ hyponym (cytokines, BNF)

(4.2) *NP1 {,} {including|especially} NP2 {or|and} NP3*
… technologies including ATLAS and SCAN …
→ hyponym (technologies, ATLAS), hyponym (technologies, SCAN)

4 Discovered Relations

The broader terms most often matched by the rules are listed below in bold type, along with five examples of discovered hyponyms in italics. The figure on the left is the number of times the broader term was found to act as a hypernym in the corpus. Unless otherwise indicated, the hyponyms could all be meaningfully grouped under their parent term in a thesaurus, and thus Hearst's rules and the Scrip corpus together form the basis for a useful taxonomy of the field of pharmacology.

53 **products** - the hyponyms were a mixture of names of drugs and companies, e.g. *ACE, Astra, Calcichew, Cognex, Glaxo,* so to be useful we need a way of subdividing these.
45 **countries** - e.g. *Africa, America, Brazil, Canada, China.*
34 **diseases** - e.g. *Alzheimer, Ebola, HIV, Marburg, asthma.*
28 **areas** - (research) e.g. *ENT, asthma, biotechnology, cancer, cardiovascular.*
18 **conditions** - all hyponyms were names of body parts or diseases e.g. *acne, arthritis, balloon, colon, diabetes.*
17 **issues** - this category was too broad to be useful, containing such diverse terms as *HIV, IP, Ukraine, adherence, compulsory.*
16 **companies** e.g. *Bristol, Centeon, David, Eskom, Glaxo.*
16 **markets** - all hyponyms were names of countries, *e.g. Asia, Australia, Canada, Denmark, EC.*
13 **bodies** (regulatory) e.g. *EC, National, WHO, WTO, World.*
13 **effects** (side effects) e.g. *bleeding, dizziness, headache, hot, loss.*
11 **drugs** e.g. *ciclosporin, cocaine, corticosteroids, heroin, leukotriene.*
11 **factors** - too broad a category to be of use, containing such terms as *construction, degree, drugs, goodwill, higher, hypertension.*

Other interesting categories which were discovered included **disorders:** *Parkinson, chronic, diabetes, hepatitis, hypertension, idiopathic, sequelae, shock, stroke;* **technologies:** *ATLAS, SCAN, gene, genomics, laser, libraries, taste* and **cancers:** *breast, colon, colorectal, lung, non, ovarian, pancreatic.*

It would be difficult to print out the entire set of results in a neat, strict indented hierarchy using a recursive algorithm, as some terms had a large number of parent terms which tended not to be very descriptive (e.g. *cancer* had the hypernyms

diseases, areas, illnesses, pathology, terminal, serious, and *UK* had the hypernyms **countries, markets, member, sites, year, 1980s**). The word most commonly matching the hyponym slot was "those", suggesting that this word should be incorporated into the original definition of a noun phrase as an alternative to "the".

5 Bootstrapping Approach to Search for New Patterns

Hearst proposed, but did not implement, a bootstrapping approach to learning new lexico-syntactic patterns indicating hyponymy. The method is to gather a list of terms for which the lexical relation is known to hold, e.g. hyponym (countries, Bulgaria). The environments where this pair of terms occur syntactically near one another should be recorded, and the most common ones can then be used as patterns that indicate the relation of interest. This approach was implemented in Perl, starting with a list of all non-spurious hypernym-hyponym pairs (as edited subjectively) with a frequency of 2 or more. The most frequently occurring of these were:

6 diseases - cancer
6 countries - US
5 bodies - WHO
5 countries - UK
4 diseases - HIV
4 agreements - trade
4 countries - Canada

Whenever both the parent and child terms were found in the same sentence in the Scrip corpus, a record was kept of which term was the broader (bt) and which was the narrower (nt) and the intervening words. For example, if a sentence containing "diseases including cancer" was found "bt including nt" would be stored. The frequency of each of the stored strings was found at the end of the program, the most frequent being assumed to provide the most useful contexts. The most frequent learned contexts are listed below:

37 nt bt
20 nt and other bt
20 bt such as nt
12 bt, including nt
8 bt, including the nt
6 bt such as the nt
6 bt, such as nt
5 bt including nt
4 bt and nt
4 bt in nt
4 bt outside the nt
4 bt including the nt

This experiment was less successful in that we were only able to relearn some of the patterns originally developed by Hearst, with the exception of "bt and nt" (considered unreliable because it will pick up sibling terms rather than parent and child) and "bt in nt" and "bt outside the nt", which show how terms are related spatially rather than reveal hyponymy. Even though "nt bt" was the most common learned pattern, it is clearly not reliable to assume that every word in a text must be a hyponym of the following word.

6 Conclusion

It has been demonstated that Hearst's rules for discovering hyponymic relations for automatic thesaurus generation were highly effective when working with Scrip, a corpus of pharmaceutical newsfeeds. Altogether 1054 unique hyponymy relations were discovered, and taking the first 200 relations to be learned as a sample, 83 of the relations were deemed useful. This suggests that about 400 useful relations were learned in total. A more rigorous method of evaluating this approach would compare the relations learned with those in an existing humanly generated thesaurus in the same domain, such as the Derwent Drug File (DDF) for pharmacology. Recall would be the (number of hyponyms found both by Hearst's method and in the DDF thesaurus) divided by (the number of hyponyms in the DDF thesaurus), and Precision would be the (number of hyponyms found both by Hearst's method and in the DDF thesaurus) divided by the (number of hyponyms found by Hearst's method).

References

(Chen et al. 95) Chen, H., Yim, T., Fye D., and Schatz, B. (1995). *Automatic thesaurus generation for an electronic community system*. Journal of the American Society for Information Science 46(3): 173-195.

(DDF) Derwent Drug File, Thompson Scientific, 14 Great Queen Street, London WC2B 5DF, England.

(Fellbaum 98) Fellbaum, C. (1998). *A lexical database of English: The mother of all WordNets*. Special Issue of Computers and the Humanities, ed. P. Vossen, pp. 209-220.

(Grefenstette 94) Grefenstette, G. (1994). *Explorations in automatic thesaurus discovery*. Norwell MA: Kluwer Academic Publishers.

(Hearst 92) Hearst, M. A. (1992). *Automatic acquisition of hyponyms from large text corpora*. Proceedings of the 4th International Conference on Computational Linguistics (Nantes, France): COLING 1992.

(Kageura et al. 00) Kageura, K., Tsuji, T., and Aizawa, A. N. (2000). *Automatic thesaurus generation through multiple filtering*. Proceedings of the 18th International Conference on Computational Lnguistics (COLING 2000), Vol 1: 397-403.

(MEDLINE) MEDLINE, US Library of Medicine, http://medline.cos.com

(Pereira et al. 93) Pereira, F., Tishby, N. and Lee, L. (1993). *Distributional clustering of English words*. Proceedings of the Association for Computational Linguistics (ACL 93): 183-190.

(Salton 89) Salton, G. (1989). *Automatic Text Processing*. Reading MA: Addison-Wesley.

A Comparison Between a Rule-Based and a TBL-Based Approach for Temporal Element Extraction

Argyrios Vasilakopoulos and **William J. Black**
School of Informatics
University of Manchester
Sackville Street, Manchester M60 1QD
United Kingdom
mcaikav2@co.umist.ac.uk, w.black@manchester.ac.uk

ABSTRACT

Text Mining is a relatively modern area of research that combines the disciplines of both Information Extraction (IE) and Knowledge Discovery and Management. In this respect, the importance of IE is straightforward, as this constitutes the primary stage of text analysis and produces the basic information for subsequent analysis (data mining). One rather neglected type of IE analysis is the extraction of temporal information, i.e. recognition of temporal expressions and their meaning. This paper presents the comparison of two popular approaches for the extraction of Temporal Elements as defined in the TimeML specification: a rule-based one, using human authored rules, and a machine learning one, following Brill's TBL paradigm.

Keywords

Temporal Information, Machine Learning, Transformation-based Learning, Temporal Expression, Event, Signal.

1. Introduction

The extraction of temporal information is one of the targets of several stages during an Information Extraction Analysis pipeline. Traditionally, temporal information has been recognized and extracted as part of higher level analyses such as Named Entity Recognition and Template Element Extraction (see MUC Guides [1], [2]). Recently, however, the interest of a large part of the research community has been focused on the temporal aspects of NLP and IE, and research is being carried out towards the following three directions:

- Temporal Information Representation: Research towards the creation and standardization of convenient ways for representing time in NL texts.
- Temporal Expression Extraction: This refers to the task of extracting and normalizing (assigning a temporal value - meaning) temporal expressions.
- Event Temporal Anchoring and Ordering: This is the task of assigning timestamps to and temporally ordering events found in NL texts.

The purpose of this paper is to investigate the suitability of human authored rules (translated into finite state structures) and machine learning for carrying out the task of *temporal element* recognition. By temporal elements we mean time expressions and events found in NL texts. In this respect, our work could be included in the second category in the above classification, although we do not deal at the current stage with the normalization of the extracted information.

The rule based system we have been experimenting with is based on linguistic rules

written according to a context sensitive grammar formalism (Cafetiere [3]) and is currently being used for carrying out various IE tasks such as domain specific named entity recognition and event extraction with appropriate slot filling[1]. Regarding the ML-based algorithm we have been experimenting with the Transformation-Based Learning algorithm initially proposed in [4]. This is an approach based on supervised learning that has been extensively used for various tasks such as: PP attachment disambiguation[5], part-of-speech tagging[6], text chunking[7], dialog act tagging[8], named entity recognition[9], etc.

This paper is organized as follows: in the next section we briefly present the Cafetiere formalism and the way we have used it for writing appropriate rules. Section 3 describes the ML algorithm outlining certain implementation issues. Section 4 discusses our results and the final section consists of our conclusions and our thoughts about future work.

2. Rule-based Approach

The first of the two approaches we experimented with is the application of hand-crafted linguistic rules for recognizing and normalizing temporal expressions and extracting events and signal (connective) words. The experiments were based on the Cafetiere Environment (see [10]) and the rules authored according to the Cafetiere formalism (for a full description of the formalism please refer to[3]). Briefly, the Cafetiere rules are of the following format:

```
A => B \ C / D >> E;
```

In the above expression **C** is the textual span to be assigned the annotation **A** given **B** and **D** (the left and right context respectively). **E** is an optional constituent that links the current annotation to an already recognized entity (antecedent) and is used to provide a solution to coreference. Examples of rules that match an event and a date can be found at the appendix A. Rule A.1 indicates the recognition of a personnel event (addition/resignation) for a person. Please note that this rule is intended to extract only the event-denoting phrase and certain TimeML specific additional features. Rule A.2 describes a rule that matches an absolute time expression, calculating its temporal value at the same time. More specifically, it fires when the actual text pattern is *<month_full|month_abbr.> (.) <day> (,) <year>*. The calculation of the actual value string, according to the format proposed in TimeML, is created by the *assign* feature of the LHS of the rule. This feature causes the creation of a new feature named *value* whose value is the concatenation of the variables *_year, _month* and *_day* and the string literals '-' in the way shown above. The variables themselves hold respective information: the *_year* variable holds the actual year number, the *_month* variable holds the ordinal number for the month acquired from the ontology[2] (i.e. 11 for November) and the *_day* holds the token (number) that indicates the day part of the absolute time expression (date). Finally, rule A.3 indicates the use of the coreference operator (>>) to calculate the temporal value for the word "*today*". This is done by making "*today*" corefer to the timestamp of the document found in the *dateline* part of the document.

3. Simple and Multidimensional TBL

[1] The Cafetiere Environment is used in the Parmenides project. Project website: http://www.crim.co.umist.ac.uk/parmenides.

[2] For the purposes of Information Extraction of the temporal expressions we have constructed a knowledge base that contains basic temporal information (days, months, adverbials, etc.) in Protégé.

The second experimentation has been based on two different versions of the Transformation-based Error-Driven Learning approach proposed by Brill in [4]. The first is an implementation of the original algorithm in Java as described in [6]. The second implementation is a variant of the approach similar to [11]. Our approach deviates from the one described in [11] in that it does not perform any "multi-targeted" learning, but instead, it is based on an extended set of features for every single token and targets only in one classification. In our case, the classification for both algorithms is based on the B-I-O notation. According to this, B- indicates the beginning and I- an intermediate token of an entity of interest, while O is used to mark any other token of no importance to the task. The entity categories we use are three: *event, timex* and *signal*, hence the annotation of the following short text:

Token	Classification
Two	B-TIMEX
days	I-TIMEX
ago	I-TIMEX
,	O
Biotech	O
announced	B-EVENT
the	O
. . .	. . .

The original TBL algorithm proposes the use of the *word* and *tag* features to be used in the *transformation rules*. Our approach also involves the *pos, orth* and *onto* features. For the *pos* feature we use the Penn Treebank tagset for this experiment, although any tagset can be used. The values for the *orth* feature are:{*lowercase, capitalized, caphyphenated, lhyphenated, uppercase, multicap, upperdotted, initial, initialdot, arithmetic, doublequote, apostrophe, number, numberrange, bracket, punct, space, other*} and are self explanatory regarding their meaning. The values for the *onto* feature come from the temporal ontology mentioned earlier.

The TBL lexicon holds the frequencies for the tags for every single token of the training corpus as originally described, but the templates are extended to include the additional features. Appendix B contains the set of templates we have been experimented with. The following two rules have been generated by the two variants of the algorithm and indicate the use of the extended templates:

```
O -> I-TIMEX :
tag I-TIMEX @ [-1] & tag I-TIMEX @ [1].

O -> B-TIMEX :
pos CD @ [0] & word in @ [-1].
```

The first of the above two transformation rules is generated using the standard TBL version. Its meaning is to change the tag of the current word from O to I-TIMEX if both words on its left and right have been classified as I-TIMEX. The second rule demonstrates the use of the *pos* feature and its meaning is to mark something as B-TIMEX if it is currently marked up as nothing (O), it is a number (CD) and follows the word *in.* This second rule is generated from the multidimensional variant of TBL.

4. Results

For our experiments we have used the Timebank corpus [12]. To our knowledge, this is the most complete temporally annotated corpus available and contains 186 documents from the financial domain. In order to evaluate the performance of the algorithms, we have initially transformed the corpus in the CAS format [13] and split it in two parts: the training corpus containing the nine tenths of the original corpus (167 documents) and the test corpus consisting of 19 documents. The training corpus has been used for training the ML algorithms and writing patterns for the Cafetiere Environment, and the test corpus for performing the comparisons. The Cafetiere rule set consists of 59 human authored rules, while both TBL-

based algorithms have been trained using a score threshold of 3 and accuracy threshold of 0.75. Both rule sets of the TBL-based algorithms consist of 300 rules (the first 300 rules have been kept). The following Table 1, Table 3 and Table 2 show the results of all experiments:

	GS	Rule-Based System				
		Found	Corr	R	P	F
Events	944	661	472	50	71.41	58.82
Times	155	168	134	86.45	79.76	82.97
Signals	210	1168	205	97.61	17.55	29.75

Table 1: Results for the Cafetiere System.

	GS	Original TBL				
		Found	Corr	R	P	F
Events	944	740	540	57.20	72.97	64.13
Times	155	183	111	71.61	60.66	65.68
Signals	210	183	183	44.29	54.39	48.82

Table 2: Results for the original TBL implementation.

	GS	Multidimensional TBL				
		Found	Corr	R	P	F
Events	944	729	559	59.22	76.68	66.83
Times	155	168	114	73.55	67.86	70.59
Signals	210	168	92	43.81	57.86	49.86

Table 3: Results for the multidimensional TBL.

Our results and conclusions are comparable to the ones stated in [14]. Regarding various other researchers ([15], [16], [17]) our results (rule-based) are also comparable, although a straight comparison cannot be achieved due to the different corpora and target annotation schemata used. According to our results, the rule-based system outperforms the other two ML-based implementations for the time expression extraction task. However, the ML-based algorithms seem to yield better results when it comes to the recognition of event and signal phrases. The reason for this is the fact that time expressions are multi-word annotations having some specific format, whereas the events and signals are mostly single-token annotations. In this respect, the TBL algorithms, which work on per token basis, fail to mark up correctly both boundaries of the temporal expression. This is also evident in Table 4 where a token-based evaluation is performed. This table shows the tokens that have been correctly marked up according to the entity they belong to.

	GS	Multidimensional TBL				
		Found	Corr	R	P	F
Events	1189	790	624	52.48	78.99	63.06
Times	441	476	395	89.57	82.98	86.15
Signals	269	184	118	43.87	64.13	52.10

Table 4: Token-based evaluation for the multidimensional TBL.

From the above table it is evident that the ML algorithms we used do not perform as well as the rule-based ones when they are to extract multi-word annotations. On the contrary they are better in extracting single-word annotations such as event-denoting phrases and signal words.

Another outcome from our experiments is the difference between the performances of the two versions of the TBL algorithm. The multidimensional version produces better results from the original TBL version and it seems that it can perform better for identifying the boundaries of long (multi-span) annotations. This is evident if we compare the results of the two TBL variants where we will notice that the difference in the performances is maximum for the recognition of time expressions. Further experimentation with modified templates for the multidimensional TBL, enrichment of the temporal ontology with events and even consideration of further features (such as lemma information) could possibly yield even better results.

5. Conclusions

In this paper we described our experiments on the investigation of the use of a human crafted rule based system and two machine learning based implementations for the extraction of temporal elements (time expressions, event-denoting phrases and signal words) from natural language texts. As a very first investigation on this comparison, we have not tried to perform any time expression normalization, which is left as future work. Our current experiments showed two things: First, the superiority of the human generated patterns for recognizing complex time expressions compared to the better single-span element recognition achieved by the ML based approaches. Second, the better performance of the multidimensional TBL algorithm, which is due to the use of additional information apart from the classification tag originally considered.

Our plans for future work include the enrichment of the ontology used with information about events and the use of additional features and modified templates for the TBL variant we have already implemented. Additionally, we shall deal with the recognition of the actual temporal value for the extracted time expressions using again the options of human crafted rules and ML-based systems.

Acknowledgements

This work was funded by the Parmenides Project (contract No. IST-2001-39023).

References

[1] N. Chinchor and P. Robinson, "MUC-7 Information Extraction Task Definition (version 5.1)," presented at Message Understanding Conference, 1997.

[2] N. Chinchor and E. Marsh, "MUC-7 Named Entity Task Definition (version 3.5)," presented at Message Understanding Conference Proceedings, 1997.

[3] W. J. Black, J. McNaught, A. Vasilakopoulos, K. Zervanou, and F. Rinaldi, "CAFETIERE: Conceptual Annotations for Facts, Events, Terms, Individual Entities and RElations," UMIST, Manchester 2003.

[4] E. Brill, "Error - Driven Learning and Natural Language Processing: A case study in Part-of-Speech tagging," *Computational Linguistics*, 1995.

[5] E. Brill and P. Resnik, "A Transformation-Based Approach to Prepositional Phrase Attachment.," presented at COLING 1994, Kyoto, Japan, 1994.

[6] A. Vasilakopoulos, "Improved Unknown Word Guessing by Decision Tree Induction for POS Tagging with TBL," presented at CLUK, Edinburgh, 2003.

[7] L. A. Ramshaw and M. P. Marcus, "Text Chunking Using Transformation-Based Learning.," presented at The ACL Third Workshop on Very Large Corpora., 1995.

[8] K. Samuel, S. Carberry, and K. Vijay-Shanker, "Dialog Act Tagging with Transformation-Based Learning.," presented at COLING/ACL 1998, 1998.

[9] W. J. Black and A. Vasilakopoulos, "Language Independent Named Entity Classification by Modified Transformation Based Learning and by Decision Tree Induction," presented at CoNLL, Taipei, 2002.

[10] A. Vasilakopoulos, M. Bersani, and W. J. Black, "A Suite of Tools for Marking Up Temporal Text Mining Scenarios," presented at LREC 2004, Lisbon, Portugal, 2004.

[11] R. Florian and N. Grace, "Multidimansional Transformation-Based Learning.," presented at 5th Computational Natural Language Learning Workshop (CoNLL-2001), Toulouse, France, 2001.

[12] D. Day, L. Ferro, R. Gaizauskas, P. Hanks, M. Lazo, J. Pustejovsky, R. Sauri, A. See, A. Setzer, and B. Sundheim, "The TIMEBANK Corpus," presented at Corpus Linguistics, 2003.

[13] F. Rinaldi, J. Dowdall, M. Hess, J. Ellman, G. P. Zarri, A. Persidis, L. Bernard, and H. Karanikas, "Multilayer Annotations in Parmenides.," presented at The K-CAP 2003 Workshop on Knowledge Markup and Semantic Annotation., Sanibel, Florida, USA, 2003.

[14] D. Ahn, S. F. Adafre, and M. d. Rijke, "Extracting Temporal Information from Open Domain Text: A comparative Exploration.," presented at the 5th Dutch-Belgian Information Retrieval Workshop (DIR 2005). 2005.

[15] N. Vazov, "A System for Extraction of Temporal Expressions from French Texts Based on Syntactic and Semantic Constraints," presented at ACL 2001 Workshop on Temporal and Spatial Information Processing, Toulouse, France, 2001.

[16] E. Saquete, P. Martinez-Barco, and R. Munoz, "Recognizing and Tagging Temporal Expressions in Spanish," presented at Workshop on Annotation Standards for Temporal Information in Natural Language (LREC), Palmas, Canary Islands, Spain, 2002.

[17] G. Puscasu, "Framework for Temporal Resolution," presented at LREC 2004, Lisbon, Portugal, 2004.

APPENDIX A – Cafetiere rules for temporal element extraction

A.1) # . . . the (immediate) addition/resignation of X . . .

```
[sem=NP, type=PNAMEX, eventClass=_type, tense=NONE,  aspect=NONE,
polarity=POSITIVE, voice=NONE, oids=_oids, rulid=nom1]=>
[token="the"],
[token=__ignored]?
\
[lookup<=personnel_event],
/
[token="of"],
[sem=person]
;
```

A.2) # November (Nov.) 16, 2004

```
[sem=date, type=timex, assign="value=concat(_year,'-',_month,'-
',_day)", rid=rdate1]=>
\
[lookup=month, value=_month],
[token="."]?,
[pos=CD, token=_day],
[token=","],
[token="????", pos=CD, orth!=lowercase, token=_year]
/
;
```

A.3) # Rule that uses Coreference

```
[sem=timex, type=timex, rulid=gen2, value=_v]=>
\
[token="today"]
/
>>[value=_v, zone=dateline];
```

APPENDIX B – Extended set of templates for the multidimensional TBL

```
A -> B tag C @ [ -1 ] .
A -> B tag C @ [ 1 ] .
A -> B tag C @ [ -2 ] .
A -> B tag C @ [ 2 ] .
A -> B tag C @ [ -1 -2 ] .
A -> B tag C @ [ 1 2 ] .
A -> B tag C @ [ -1 -2 -3 ] .
A -> B tag C @ [ 1 2 3 ] .
A -> B tag C @ [ -1 ] & tag D @ [ 1 ] .
A -> B tag C @ [ -1 ] & tag D @ [ -2 ] .
A -> B tag C @ [ 1 ] & tag D @ [ 2 ] .
A -> B word C @ [ 0 ] & tag D @ [ -2 ] .
A -> B word C @ [ 0 ] & word D @ [ -2 ] .
A -> B word C @ [ -1 ] .
A -> B word C @ [ 1 ] .
A -> B word C @ [ -2 ] .
A -> B word C @ [ 2 ] .
A -> B word C @ [ -1 -2 ] .
A -> B word C @ [ 1 2 ] .
A -> B word C @ [ 0 ] & word D @ [ -1 ] .
A -> B word C @ [ 0 ] & word D @ [ 1 ] .
A -> B word C @ [ 0 ] & tag D @ [ -1 ] .
A -> B word C @ [ 0 ] & tag D @ [ 1 ] .
A -> B word C @ [ 0 ] .
A -> B word C @ [ 0 ] & tag D @ [ 2 ] .
A -> B word C @ [ 0 ] & word D @ [ 2 ] .
A -> B tag C @ [ -1 ] & pos D @ [ 1 ] .
A -> B tag C @ [ -1 ] & pos D @ [ -2 ] .
A -> B tag C @ [ 1 ] & pos D @ [ 2 ] .
A -> B word C @ [ 0 ] & orth D @ [ -2 ] .
A -> B orth C @ [ 0 ] & pos D @ [ -2 ] .
A -> B onto C @ [ -1 ] .
A -> B onto C @ [ 1 ] .
A -> B onto C @ [ -2 ] .
A -> B onto C @ [ 2 ] .
A -> B onto C @ [ -1 -2 ] .
A -> B onto C @ [ 1 2 ] .
A -> B pos C @ [ 0 ] & word D @ [ -1 ] .
A -> B pos C @ [ 0 ] & word D @ [ 1 ] .
A -> B pos C @ [ 0 ] & tag D @ [ -1 ] .
A -> B pos C @ [ 0 ] & tag D @ [ 1 ] .
A -> B pos C @ [ 0 ] & tag D @ [ 2 ] .
A -> B pos C @ [ 0 ] & word D @ [ 2 ] .
A -> B onto C @ [ 0 ] & word D @ [ -1 ] .
A -> B onto C @ [ 0 ] & word D @ [ 1 ] .
A -> B onto C @ [ 0 ] & tag D @ [ -1 ] .
A -> B onto C @ [ 0 ] & tag D @ [ 1 ] .
A -> B onto C @ [ 0 ] & tag D @ [ 2 ] .
A -> B onto C @ [ 0 ] & word D @ [ 2 ] .
```

Textual Information Extraction Using Structure Induction

Alexandre Saidi
LIRIS-CNRS (UMR 5205)
Ecole Centrale de Lyon, Mathematics and Computer Science Department
B.P. 163. 69134 Ecully - France
Alexandre.Saidi@ec-lyon.fr

Abstract

Given the huge quantity of the current available textual information, *Text Mining* process aims the task of searching useful knowledge in a natural language document.

When dealing with a free-format textual corpus where the linguistic rules are not respected, the time consuming morpho-syntactic analysis is not of a great help. However, text mining techniques process may exploit sub-structures in the text.

In this paper, we report how the Grammatical Induction can help to extract the (partial) structure of the sub-languages used in a text. We present the practical contribution of the Grammatical Induction by reporting an Information Extraction process applied to a fragmented announcement corpus.

1 Introduction

Textual databases constitute the major part of the current available information. Significant research work concentrates on the Information Extraction (IE) from these databases.

Given a textual corpus, the information extraction process applied to the texts by techniques of *Text Mining* (e.g. (Fayyad'96), (Hearst'97), (Tan), (Grishman'97), (Ahonen'98)) consists on the search for no-explicit information in these texts. As an example, Text Mining can extract significant information like the research directions of a university from a corpus of seminar announcement.

In a basic approach, IE task would be tedious if no *a priori* structural information about the text is available. On the other hand, given the cost of a syntactical analysis, an IE process based on a whole morpho-syntactic analysis of documents would not often be realistique. When dealing with free-format texts, such analysis would not be of a great interest in the text mining process usually based on key patterns.

In the case of free format texts, the rules of linguistic grammars are seldom respected. These texts rather tend to transmit information with few words without using entities such as determinant, verb and other punctuation.

In the current paper, we are interested in the structures of sub-languages present in free-format texts. Given the structure of the sub-language representing e.g. the *address* in the advertisement of an exposure on *Egypt* that will take place in *Paris* may avoid concluding too quickly (and wrongly) on the place of the exposure upon the simple presence of *Egypt* city name.

Text Mining research field has been focused on since 1991 through MUC programs. However, it is still domain specific and time-consuming to build a new system or to adapt an existing one to a new domain. Although symbolic and statistical methods have been applied in some IE systems (e.g. (Califf et Mooney'97), (Huffman'96)), not a lot have combined Grammatical Inference with (naive) statistical information.

Techniques of Grammatical Inference (GI) ((PR'82), (Miclet)) promise to be useful in this field by accompanying the process of *Text Mining* to exploit the (partial) morpho-syntactic structure of patterns (or of sub languages) with a minimum amount of information on the contents structure. These techniques attempt to induce the structures of a source data (flow of signs) by a set of production rules of a regular grammar[1]. The induced grammar being an element of a (language-inclusion) lattice, then the text mining is concerned by an informed search (seen as generalisation) within this lattice carrying required information and semantics.

In this paper, we consider the case of a textual base (with free syntax) of seminar announcements. Note that various formats of announcement are possible. A first example of such announcement is given below.

```
Seminar of the Institute of Nuclear physics of Lyon
problem of the mode conversions
Yves  Colin de Verdiere
```

[1] In the Text Mining field, one is interested in the (so-called *surface*) structure of the sub-language usually governed by *regular* grammatical rules.

```
Fourier Institute of Grenoble
14:30 H - Room 27
Paul Dirac Building
```

The aim of the processing of these announcements is to extract various information such as the *Date* or the *Subject* in a seminar. Finals measurements like the research fields of a university (or a researcher, etc.) can then be extracted. In the supervised process we consider, the text mining task applied to such a corpus could break up into several phases.

In this process whose goal is to extract slot fillers, the important templates slot fillers are already defined by an expert[2] : he/she knows in advance which kind of information is contained (and sought) in the base. The principal phases of this process is briefly described below. Note that a **L** means that the action takes place during the *Learning* phase and a **T** means during the *Test* phase :

Preprocessing (LT) transformation and homogenisation of the characters, sentence extraction, suppression of some common words and punctuation in the text, etc.

Morphological Analysis (LT) extraction of lexeme and basic lexical classes; constitution of a dictionary/lexicon of terms and keywords in various slots (e.g. the *institute* for an announcement) starting from positive examples;

Partial syntactic analysis (LT) regrouping of the lexeme, constitution of simple and partial syntactic entities according to the structures of the sublanguages;

Grammatical Inference (L) training of the grammar of sublanguages from positive examples together with the description of negative examples;

Statistic analysis (Bayesian) (L) extraction of measurements, frequencies, weight and probabilities on the (couple of) patterns in the sample set;

Adding (semantical) actions (L) to the induced grammar using the results of the Bayesian analysis[3]

[2]The expert in this domain is just a scientifique-researcher familiar with such seminar announcements

[3]a semantical action is a term from the *syntax directed* and the *Attributed Grammars* paradigm which denotes (no syntactic) actions based on the attribute values. Distinguished from the pure syntactical analysis, such actions take place in a production rule if the rule applied.

Added to this process is a postprocessing and decision making phase (e.g. classification of unprocessed free zones, etc.) that will complete the process of learning. For other corpus, we may also need to annotate samples in the same phase.

It is also appropriate to note that examples can be incomplete. For instance, the *Hour* may be missing within an announcement or it can be expressed in a different form (for example, by the "Friday afternoon" expression).

In the reminder of this paper, we will describe the interest of the Grammatical Inference (section 3) and the Bayesian analysis (section 4) with respect to the textual IE task. We describe examples of announcements database we considered and give some aspects of the realization in the section 7.

2 Seminar Announcement Corpus

We consider a textual database of seminar announcements. Announcements may have several formats. Below, there are several examples of seminar announcement we made up via the WEB. Some examples are complete : examples 1(given above), 2 and 3 contain significant informations. In example 4, the *Address* and the *Place* are missing. In example 5, the *Speaker* is not given. Note also that these examples were originally in French. We give hereafter some of their English translation.

```
2- Seminar
   Conference Room, 1st floor,  IRIGM   Build.
   Thursday April 11 2002, 14h30,  Yves Mheust,
   ENS Paris
   Flow of Stokes in a rough open fracture
3- Seminar in Toulouse:  Migration towards the
                free,   Utopia or reality?
   By Romance Nicolas.  05/16/2003 at 13:15
4- seminar:  security and Internet
   Paul-Andre Pays,  Tuesday, Jan 25 1996, 15:15
5- Arithmetic Seminar
   Thursday February 25 at 11h
   in  Kampe  de Feriet room,  M2  Build.
```

2.1 Slots and Fillers

The following slots are defined for the seminar announcements processing (abreviations are further used in the paper).

$<Sub>$	the (general) *Topic* and the *Subject* of the seminar,
$<Org>$	the organiser, i.e. a university, lab.,...
$<Adr-Plc>$	the address and/or the place where the seminar takes place,

$<Sp>$	the person who will make the talk,
$<OrgSp>$	the organisation of the Speaker (e.g. the lab. of the Speaker),
$<Date>$	the date of the seminar,
$<Hr>$	the beginning hour (or the time range) of the seminar.

An announcement starts with the *seminaire* (seminar) keyword.

3 Grammatical Inference

In the IE applied to natural language texts, there are major differences between the Sentence Analysis and traditional parsers . The goal of syntactic analysis in an IE system is not to produce a complete parse tree for each sentence in the text. Instead, the system needs only to perform partial parsing: that is, it needs only to construct as much structures as the IE task requires.

Current methods (see e.g. (Weischedel), (Appelt'95)) use generally global constraints to resolve local ambiguities. But because of inevitable gaps in grammatical and lexical coverage, full sentence parsers may end up making poor local decisions about structures in their aim to create a parse spanning the entire sentence.

Furthermore, the syntactic analysis in a text mining process is avoided for several more reasons:
- the cost and the complexity of this analysis,
- the very few use of the results of this analysis (the goal is not to correct errors or to translate the text),
- the texts may not follow the correct and complete syntax (of French in our case), etc.

A partial parser looks up for fragments of text that can be reliably recognised, e.g., *noun* and *verb* groups. Because of its limited coverage, a partial parser can rely on general pattern-matching techniques, particularly finite-state machines, to identify these fragments deterministically based on pure local syntactic elements. Partial parsing is well suited for information extraction applications for an additional reason : the ambiguity resolution decisions that make full parsing difficult can be postponed until later stages of processing where top-down expectations from information extraction task can guide the system's actions.

In our seminar announcement, the subject is similar to a *noun group* but may not follow its rigorous syntax. Then, the inference stage helps, in this case, to retain *effective* rules used in the examples. Therefore, the corresponding text mining process will rather be a syntax directed process.

Starting from a sample basis (positive examples and negative cases description, see the section 5.1), the Grammatical Inference (GI) induces production rules of a regular grammar[4] (a deterministic finite state automaton, DFA) of this sample set. In the test phase, the sentences presented to the grammar will be regarded as pertaining (or not) to the language generated by induced grammar.

The Grammatical Inference carries out a classification of the sentences (*accept* or *reject* means belonging or not to a given language) but, in its original form, it does not handle the semantics of these constructions. Hence, Bayesian measures will guide the process by predicting the slot to be submitted to the grammar. The IE process is then achieved with more precision and reliability (see also (Freitag'97)).

4 Naive Bayesian use

Several techniques of text mining use the Bayesian analysis that (even in its naive form) gives interesting results. In the method known as *naive Bayesian*, the document is presented as a vector of characteristics (e.g. various sections of an announcement). Other presentations such as *bag of words* consider the text in the form of a collection of words where any internal structure (physical, logical, morpho-syntactic or semantics) is inhibited.

The Bayesian rule is recalled below. Given a hypothesis (e.g. to have such a section of the class C in such a context inside a seminar announcement) and an example of announcement E over C, we have:

$$Pr(C/E) = \frac{Pr(E/C)\,.\,Pr(C)}{Pr(E)}$$

The idea is to express the weighted probability of the membership of a pattern or a sub-language within a class C according to the characteristic of the text E and those of other texts classified as such.

To summarise the current process, key patterns leading to recognise the various (but not all) fillers of an announcement are first defined during the

[4]We note that the Context-Free grammar induction is an actual and active research filed facing hard constraints making the general Context Free induction problem no-decidable.

training stage. Together with the key patterns, the frequency measurements and the regular production rules will help to decide (nearly *naively classify*) a section of the announcement. During the test phase, a pattern p first gets a probability to belong to a slot filler by the presence of a deterministic keyword (100 %) and/or by the probability (from the frequency table) of its (possibly left and right) context. p is then submitted to the induced grammar according to these probabilities. Failure cases are postponed to the postprocessing step[5]. Further more, the process uses the backtracking in order to re-consider other possibilities (see section 7 for the *Sub* filler).

5 The Application of the GI

It is easy to note that a simple textual search cannot be appropriate for extracting knowledge from our seminar announcements. Methods of knowledge extraction based on the Bayesian analysis allow to predict the position of a given information in the text together with its average length (see e.g. (Freitag'97)). This technique, based on the learning of the position of a section (e.g. the $<Sub>$ section) would not be appropriate here because the format of announcements are free. In addition, an announcement can be incomplete. Thus, getting the induced grammar of e.g. the $<Adr - Place>$ section will make it possible to analyse the content of that sub-language.

Obvisiously, the Inference engine applied to the entire announcement gives the sequential structure of an announcement: the sequence of various slots or sub-languages. This does not bring any relevant information. One obtains a grammar that confirms that the format of an announcement is in a free-format. Instead, the grammatical inference is used in various sub-languages (e.g. the *heading* or the *subject* of an announcement) that may contain relevant information. As an example, the heading can contain a topic, a subject or an organiser that can possibly extend in the reminder of the announcement. The subject (*Sub*) can add precise details to the Topic of the seminar and vice versa. Such complementary data are registered both in the frequency table and hard coded in the production rules. The sequence of operations is governed by the key patterns, the probabilities from the frequency table (table 1) and, finally, by the production rules.

A quite brief presentation of the applied GI process is given in the section 11. It may be noted that if the Grammatical Induction is processed only upon positive examples (the set I_+ set below), the result tends to over-generalise the language induced. Hence, the expert may express negative descriptions that are representative of the *words* that must be rejected[6]. For example, he may state that a *seminar announcement heading containing the Hour value* must be rejected. The following example contains some negative examples for an announcement heading (the set I_-).

5.1 An Example

As an example, the results of the grammatical inference on the **heading** of announcement follows. The grammar below partially describes what the headings of the sample set contained. Hence, the following I_+ does not cover all possible headings in all seminar announcements, but those of the sample set. The GI engine received a set of (more than) 100 positive examples (see section 8) from the test corpus.

I_+={'SDON', 'S:T', 'S', 'ST','SDT','SV:T','SN:T', ...}
I_-={'Sa', 'SS', 'S::T', 'S::L', 'S::N','SDD','SOO', 'S', 'Sa:', 'SD:', 'SaVV', ...} where

S : the "séminaire" keyword (seminar in English),

D : $<Det>$, a determinant (e.g., *'du'*, *'de la'*, *'des'*) like *'of'* or *'of the'* in English

T : $<Thème>$, an exposed $Topic - Subject$ (e.g. *Algorithm*, *Complexity , Internet and Security*, etc.),

N :$<Nom>$, a Noun, e.g. name of a research laboratory ,

O : $<Org>$, an organisation name (e.g. *institute, laboratory, university, school...*)

V : Ville, name of a City, e.g. *Toulouse*

':' : this character,

'à' : this character (stands for *'at'* or *'in'*, ... in English).

The induced grammar accepts the language L_+: the induced language for the I_+ and rejects those of L_- (the induced language of I_-). The final induced automaton accepts the language given below[7]. The rules that reject unsuitable constructions (i.e. words in L_-) are not reported here for

[5] e.g. in the case of ambiguity (or failure on p), if a pattern p' ($p' \neq p$) has been successfully recognised to fulfil a slot filler, the pattern p is *tried* against other related sections. Several lookup may be necessary in more complex cases. A *blind* application of the induced production rules is the last chance.

[6] For the seminar announcement case, negative examples are quite straightforward.

[7] Notation: $(X \| Y)$ means (X or Y). The dot denotes the monoid concatenation and ε denote the empty string.

the sake of clearness. However, one may observe that a rejection takes place in the induced DFA when a derivation (upon a token) leads to a final *failure* state F_- (see section 5).

The language of the induced finite state automaton The language induced from the set $I = (I_+ \cup I_-)$ for the *heading* part of announcements is given below. Recall that this definition gives only the successful derivation paths.

L+ = "Séminaire" . L1
L1 = ϵ || (':' || '') . L3 || $<Nom>$. L5 || $<Thème>$. L6
L3 = $<Org>$. L6 || ($<Thème>$ || $<Ville>$) . L1
L5 = ':'. L3 L6 = ϵ || $<Nom>$. L1

Nota Bene: the induced grammar being a logical grammar (called DCG), predicates expressing the constraints and other actions are then added to its rules (see the example below). For example, while recognising (in their context):
- a $<Thème>$ may contain a part of the *Subject*; then the value corresponding to the *Subject* will be added to the $<Sub>$ filler;
- for a $<Ville>$, the corresponding value will be added to $<ADR - Place>$ filler[8].

Other possible adjustments are achieved during the post-processing phase.

5.2 An example: the *Date* analyser

Below, some of the induced grammar rules (together with their semantical actions) for the $<Date>$ filler are given. The lack of any part of a *Date* (e.g., the *day-name*) is not reported here[9].

< Date >:: ["*date*"][" : "]["*le*"][$<Day - name>$]
[$<Mid - day>$]["*le*"] $<Day>$ [$<Sep>$]
$<Month>$ [$<Sep>$] $<Year>$.
<Mid − day>:: $<Word>$ {\$1 ∈ {"*matin*"} ;
\$1 ∈ {"*matin*"} ; *add*(*part_of_Heure*, "8*h* − 12*h*", 100)
OR \$1 ∈ {"*après − midi*"} ;
add(*part_of_Heure*, "14*h* − 18*h*", 100)}.
< Month >:: $<Number>$ {\$1 ∈ {1..12} ;
add(*part_of_Date*, \$1, 100)}
|| $<Word>$ {\$1 ∈ {"*jan*".."*dec*"} ;
add(*part_of_Date*, \$1, 100)}.
<Day−name>:: $<Word>$ {\$1 ∈ {"*lun*".."*sam*"} ;
add(*part_of_Date*, \$1, 100)}.
< Day >:: $<Number>$ {\$1 ∈ {1..31};
add(*part_of_Date*, \$1, 100)}.
< Year >:: $<Number>$ {\$1 ≥ 1990};
add(*part_of_Date*, \$1, 100)}.
< Sep >:: '/' || ' :' || ' −' || ... −− *a separator*

Nota Bene: the value 100 (parameter of the predicate *add*) indicates the confidence coefficient of the value assigned to the filler. Here, the case of $<Date>$ is rather simple and follows a known format. We may however note that the presence of "matin/après-midi" (AM/PM in English) of the $<Date>$ will complete the $<Hr>$ slot filler.

6 Frequency Measurements

Considering a sample set of 100 examples, the percentage values are given below ($OrgSp$ abbreviates $'Organiser - Speaker'$, $Pres$ stands for $'Present'$, Sub for $'Subject'$, Plc for $'Place'$, Hr for $'Hour'$ and Sp for $'Speaker'$):

	Sub	Org	Date	Hr	Plc	Adr	Sp	OrgSp	End	Pres
Ance	14	9	41	4	14	14	9	0	0	100
Sub	0	0	4	0	9	0	23	0	9	45
Org	0	0	4	0	4	0	0	0	0	9
Date	0	0	0	77	9	0	4	0	4	95
Hr	9	0	4	0	41	0	18	0	14	86
Plc	4	0	23	0	0	36	4	0	23	91
Adr	4	0	9	0	4	0	0	0	32	50
Sp	4	0	9	0	4	0	0	36	4	59
OrgSp	14	0	0	4	4	0	0	0	14	36

Table 1: Frequency table of various sections in the seminar database

In the above table, a cell C_ij gives the frequency of the column j that followed the line i in the training set. The *Pres* column (the last one) gives the frequency (the Presence) of each line in the training set. We add to this table two other values: 77% of the announcements contain a *Topic* in their heading, and 18% of the headings contain an indication on the organiser (Org). The *present (pres)* column indicates that e.g. the *Sub* is present only in 45% of the announcements. The cells containing 0% are of a particular interest because they give indications on the cases that do not occur. For example, $<OrgSp>$ never follows the heading of an announcement.

As an example, we apply the conditional probability to the section *Sub* of the example of section 1 where the slot of the second line is not determined. This example shows how the post-processing will help deciding the slots filler. Given the table 1 above, the probability so that the unknown section (2nd line of this example) in this announcement be a *Subject* (surrounded by the *Heading* and the *Speaker*) is 12%. However, this announcement does not contain a Subject in its heading and, the Speaker is the successor of a *Subject* in $\frac{23}{45}$ cases. Therefore, the filler is predicted at 23% (weighted 51%) to be the *Subject*.

Note that the strongest probability of the section that follows the *heading* is the *Date* section. However, one can recognise a *Date* by the keywords in the induced grammar.

[8] In the $<Adr - Plc>$ context.

[9] $[x]$ means an optional x; \$k denotes the value of the k^{th} literal (a.k.a. *yacc* compiler compiler).

The depth of the Morpho-Syntactic analysis is a system parameter. In some cases (see e.g. (Appelt'95), the (partial) linguistic class from this filler can be extracted giving a (partial) Noun Group (even without any initial determinant).

7 Results for the Example

This section describes briefly of the experimentation on the seminar announcements corpus.

For the grammatical induction, the GI process is applied in the morphological step result in order to learn to reject useless combinations[10].

Textmining applied to the seminar database produced the following results for the above seminar example (confidence coefficient for a filler value is reported beside if less than 100, the original database is in French):

Org	= "Institut de Physique Nucleaire de Lyon"	
Sub	= "Le probleme des conversions de modes"	(51)
Sp	= "Yves Colin de Verdiere"	(51)
OrgSp	= "Institut Fourier Grenoble"	(61)
Hr	= "14:30 H"	
Adr_Plc	= "Salle 27-Rez de chaussee-Bt. Paul Dirac"	
Adr	= "Institut de Physique Nucleaire de Lyon"	
Date	= ""	

8 Performances Evaluation

Several textual IE systems, notably those of MUCs, involved large training corpora with over a thousand documents and their templates (see e.g. (Grishman'97)). However, such large consistent training corpora may be expansive since they would not be available for most real tasks. Users may only be willing to prepare a few dozen examples of filled templates.

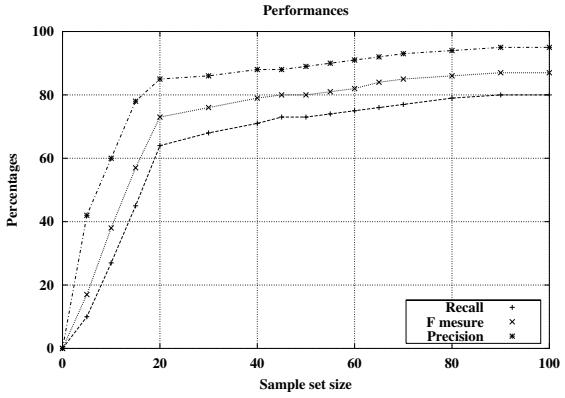
Experiments with smaller training collections, such as the 100 documents provided for MUC-6 suggest that fully automated learning techniques, when provided only with text examples and associated templates, and with minimal automatic syntactic generalisation may not be able to achieve sufficient coverage (see e.g. (Fisher'96)).

We paid a special attention to the over generalisation pitfall of the GI engine. An amount of work was done in testing the GI engine on several different corpora (bibliography, abstract, table of content, etc.) in order to improve the induction algorithm. The GI engine is parametric such that several[11] different degrees of generalisation

[10]Here, some linguistic knowledge is required to eliminate useless lexical class combinations from morphological analysis.

[11]Three for the moment

Figure 1: *Performance evaluation*



can be fixed (by variating the constraints over the language-inclusion lattice of automata). The output automaton is then tested against the training set and the one (that accepts all positive examples rejecting all negative one) with least number of states is chosen. One may observe that the refinement operator is hard-coded within the the *Congruence Predicate* of section 5 and the above (and the following) parameters.

In addition, another parameter is available in the GI engine the turns on-off the so-called *enrichment* issue[12]. The GI engine is described in (Saidi'03).

However, we are aware that larger sample sets (other domain specific corpora such as resume scanning) are needed to improve the system. Larger sample set has however an inconvenience. Recall that the search space is given by the lattice of language-inclusion specified by the GI process and illustrated by the *Congruence Predicate*. This search space grows exponentially with the size of sample set I.

Starting with 300 examples, we applied then a ten-fold cross validation and observed that the results were not significantly changed for more examples.

Metric used : using a given corpus of announcement, evaluation metrics are based on the filler presence and its prediction.

$$Precision = \frac{Number\ of\ Correctly\ assigned\ slots}{Number\ of\ assigned\ slots}$$

$$Recall = \frac{Number\ of\ Correctly\ assigned\ slots}{Number\ of\ correct\ present\ slots}$$

In addition, an harmonic measure called F-measure (see e.g. (Lehnert et Sundheim'91)) is used to give the mean of the above values:

[12]Simply speaking, the question is whether one can learn positive derivation from negative examples.

$$F - measure = \frac{Precision \times Recall}{\frac{1}{2}(Precision + Recall)}$$

The diagram of the figure 1 shows the performance percentages we obtained. As one may observe for the seminar announcements corpus, it is not surprising to have high performance values (95% and 80%) given the intended slots and the relative low risk of error. The system is quite domain specific and may even be enhanced. However, appropriate modifications are needed in order to apply it to other kind of corpus. Work is currently done to adapt the system to other corpus.

9 The Related Work

Several textual IE system have been proposed since the focus on researches started by MUC program of DARPA (e.g. (DARPAs), (Lehnert et Sundheim'91)).

The use of pattern dictionary is common to many systems. Some uses clustering to create patterns by generalising those identified by an expert (see e.g. (Soderland-1)). The dictionnary we use in the present work contains basically keywords (and their lexical class) that are used during the analysis.

Syntactic information can be used as in Autoslog ((Riloff'93), (Riloff'96)) that uses a set of general syntactic patterns validated by an expert. Among these systems, some uses advanced syntactic analysis to identify the relationship between the syntactic elements and the linguistic entities (e.g. in (Huffman'96)). This analysis is costly (when the semantic information is not used) and may limit the system specially if linguistic rules are not respected (like in our seminar examples).

In many IE systems, human interaction is highly required through different phases of training. Machine Learning techniques like decision trees are used ((McCartht et Lehnert'95)) to extract coreferences using the annotated coreference examples.

Among these systems, the current work is closed to PAPIER system ((Califf et Mooney'97)). RAPIER is an ILP system that takes pairs of documents and filled templates and induces rules that directly extract fillers for the slots in the template. This system uses constraints on words and part-of-speech tags surrounding the fillers' left and right contexts. To some extent, our system can be seen from this point of view since, as mentioned in the GI section 3, our grammatical Inference engine implements this technique implicitly. In addition these results should be compared with those of the *Named-Entity* research work (see e.g. (Bikel'99), (Palmer'97)) and aims to learn *names* by identifying all named locations, persons, organisations dates and so on.

10 Conclusions

We presented an IE system that fills slots of a template associated to seminar announcements using Grammatical Inference and Bayesian measurements.

Once the template slots are filled, current techniques of Data Mining can be applied to the database made up (see e.g. (Ahonen'98)).

Two directions are possible for the future of the current system: improving the current one and/or extending it to other corpora (like resume scanning, job announcement, marine's weather announcement, etc.).

Fore the first one, we actually work on the WEB HTML documents. In such a semi-structured document, HTML tags are of very *limited use*. In fact, HTML tags are format mark-ups and not content mark-ups. However, one can use these tags to bound various sections of a document. Grammatical Induction needs to concentrate on limited sections. Otherwise, the complexity of the GI algorithm discourages any computation. Another motivation was that HTML text database is immediately available. Here, once such information are gathered using these tags, the remaining of the task of IE (inside the text) should be done using the above system in order to extract useful information from these documents. Partial results on this corpus shows that poor HTML documents (with the simplest tags like /title, /head and /body) are not of great interest. We currently apply a HTML parser (realized by ECL students) that would reformat the HTML documents prior to any analysis. Partial results are not yet encouraging.

For the second direction, the best effort is made to extend the system to a job announcement corpus (and marine weather announcements). We have kindely been suggested to work also on resume scanning.

References

H. Ahonen, O. Heinonen, M. Klemettinen, and A. Verkamo. *Applying data mining techniques for descriptive phrase extraction in digital documents*. In Proc. Advances in Digital Libraries (ADL98), Santa Barbara, CA, 1998.

D. Appelt et all. : *FASTUS system : MUC-6 test results and analysis*. In Proc. 6^th MUC, 1995, Morgan Kaufmann.

D. Bikel'99, R. Schwartz, R. Weischedel *An Algorithm that Learns What's in a Name*, 1999

M.E. Califf, R.J. Mooney, *Relational Learning of Pattern-Match Rules for I.E.*, Proc. of AAAI Symposium on Applying Machine Learning to Discourse Processing, 1997.

ed., *Proc. of 4th and 5th DARPA Message Understanding Evaluation and Conference*. Morgan Kaufman. 1992, 1993.

U. fayyad et all *From DataMining to Knowledge discovery : An overview*. in Advances in Knowledge Dicovery and DataMining, MIT Press, Cambridge, Mass 1996.

D. Fisher'96 et all. *Description of UMass system as used for MUC-6*. In Proc.. 6^th MUC. 1996, Morgan Kaufmann.

D. Freitag'97. *Using Grammatical Inferenec to Improve Precision in*. ICML'97. 1997.

H.S. Fu. *Syntactic Pattern Recognition and Applications*. Prentice Hall, N.Y. 1982.

R. Grishman'97, *Information Extraction: Techniques and Challenges*, http://citeseer.nj.nec.com/grishman97information.html.

M.A. Hearst'97, *Text Data Mining : Issues, Techniques and Relationship to Information Axxess*, Presentation Notes for UW/MS Workshop on data mining, July 1997.

Theodore W. Hong, Keith L. Clark, *Using Grammatical Inference to Automate Information Extraction from the Web*, Lecture Notes in Computer Science - 2168, 2001.

S. B. Huffman'96, *Learning information extraction from xamples*. in Wermter, Riloff, Scheler ed. Berlin, 1996.

W. Lehnert, B. Sundheim, *A performance evaluation of text-analysis technologies*, AI Magazine 12(3), 1991.

D.D. Lewis, W.A/ Gale , *A sequential algorithm for training text classifier*, Proc. of the 7th Int. Conf. on Research and Development in Information Retrievial, 1994.

J. McCartht, W. Lehnert, *Using decision trees for coreference resolution.*, in Proc. of 4th Int. Conf. on IA, 1995.

L. Miclet. *Grammatical Inference*, Syntactic and Structural Pattern Recognition. H. Bunk and SanFeliu eds. World Scientific.

E. Califf, R.J. Mooney, *Induction of first order decision lists : Results on learning the past tense of English verbs*, Journal of AI Research., 1995

D.D. Palmer'97, D.S. Day, *A Statistical Profile of the Named Entity Task*, in proc. of th 5th. Conf. on Applied Natural Language Processing, Washington D.C., 1997 ACL.

E. Riloff, *Atomatically constructing a dictionnart for information extraction tasks*. in Proc. of 11th National Conf. on AI, 1993.

E. Riloff, *Atomatically generating extraction patterns from untagged text.*. in Proc. of 13th National Conf. on AI, 1996.

A. Saidi. *A Constraint satisfaction framework for Documents Recognition*, Workshop on Multimedia Discovery and Mining , ECML-PKDD 2003.

S. Soderland, D. Fisher'96, J. Aseltine and W. Lehnert, *Crystal : inducing a conceptual dictionnary*, in Proc. of the 14th Int. Conf. on AI, 1995

S. Soderland, D. Fisher'96, J. Aseltine and W. Lehnert, Srystal : *Issues in inductive learning of domain specific text extraction rules*, in LNCS in AI, 1996.

A.H. Tan. *Text Mining : The state of the art and the Challenges*. Proc. PAKDD'99 workshop on Knowledge Discovery from Advanced Databases. Beijing, 1999.

R. Weischedel *BBN : Description of the PLUM system as used for MUC-6*. In Proc. 6^th MUC. 1995, Morgan Kaufmann.

11 Annexe

A quite brief presentation of the applied GI process is given below (details in (Saidi'03)).

Given the sample sets I_+ (positive examples) and I_- (negative examples descriptions), one deterministic finite state automaton (DFA) is associated to each example of $I = I_+ \cup I_-$. During the GI process, states of these automata are merged according to the following predicate :

Predicate Congruence(r_1, r_2) :
adds constraints to the constraint store θ
Let r_1 and r_2 be the above rules (transitions) with $\alpha,\beta \in \Sigma$
$r_1 : [\alpha] \times s'_1 \rightarrow s_1$ $\quad$ $r_2 : [\beta] \times s'_2 \rightarrow s_2$

(1) if s_1 and s_2 are different final states in $(F_+ \times F_-)$ then add $[s_1] \neq [s_2]$.
(2) if $[\alpha] = [\beta]$ then add $([s'_1] = [s'_2] \Rightarrow [s_1] = [s_2])$ *(The DFA condition)*
(3) if $[\alpha] \neq [\beta]$ then add $[s_1] \neq [s_2]$

Here, F_+ (resp. F_-) is the set of final states for the positive examples I_+ (resp. I_-). $[\alpha]$ denotes the equivalence class of $\alpha \in \Sigma$. Identically, $[s_i]$ denotes the equivalence class of the state s_i. The aim of this predicate is to compute the equivalence classes of the states and to create a constraint store θ on the final DFA. Then, given these constraints[13] (that describe a lattice of automata), we pick up a solution that minimises the number of the states, accepting *words* of the *language* of I_+ (a.k.a. L_+) and rejecting those of L_-.

Givens the rules r_1 and r_2 above, the application of the Congruence predicate can produces 3 different configurations (i.e. $[s'_1]=[s'_2] \wedge [s_1]=[s_2]$, $[s'_1]=[s'_2] \wedge [s_1] \neq [s_2]$, $[s'_1] \neq [s'_2] \wedge [s_1] \neq [s_2]$).

Although $[\alpha]=\alpha$ in its simplest form, we introduced the notion of equivalence class for the alphabet using the lexical class function $CL(\alpha)=[\alpha]$ where:

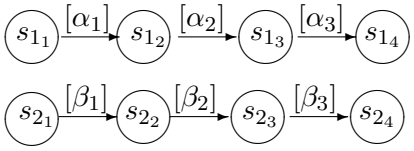
$[\alpha]=[\beta]$ iff $\alpha = \beta$ or $CL(\alpha)=CL(\beta)$, $\alpha,\beta \in \Sigma$.

For example, different city names are considered equivalent. In addition, two (possibly different) organizations (university, research laboratory) are equivalent.

Note that if we consider α_1 (resp. β_1) as the *left context* of α_2 (resp. β_2) and α_3 (resp. β_3) as its *right context*, we will cover, to some

[13] The set θ contains constraints on integers, floats and boolean expressions. The current system uses GNU-Prolog Constraint Logic Programming environment (http://www.inria.fr/)

extent, the case studied in (Califf et Mooney'97):



Applying the Congruence predicate to this case will produce 5 different configurations (depending on the equivalence classes of α_i,β_i) with various number of states in which the final induced minimal DFA has 4 states. Constraint store then will decide the final induced DFA considering all transitions and the negative examples.

It may be noted that the Grammatical Induction processed only upon positive examples (I_+) tends to over-generalize L_+ (see e.g. (Ahonen'98)). Hence, the expert may express negative descriptions that are representative of the *words* that must be rejected. For example, he may state that a *seminar announcement heading containing the Hour value* must be rejected. The I_- set of the section 5.1 contains some negative examples for an announcement heading.

Learning sure-fire rules for Named Entity Recognition

Enrique Alfonseca and Maria Ruiz-Casado

Department of Computer Science
Universidad Autónoma de Madrid
28049 Madrid
{Enrique.Alfonseca,Maria.Ruiz}@uam.es

Abstract

This paper describes a procedure for obtaining and generalising, automatically, patterns that can be used as *sure-fire* rules, i.e. rules that have a very high precision (albeit, possibly, low recall) for Named Entity Recognition. The experiments performed on the CoNLL-2003 training and test corpora show that, for people and locations, the patterns obtained attain very high precisions just by themselves. The precisions obtained for organisations and miscellaneous entities are somewhat lower, a fact which indicates that they should not be used without the aid of auxiliary gazetteers listing common entities belonging to those two classes.

1 Introduction

Named Entity (NE) Recognition is usually defined as the task of identifying and annotating instances of particular Named Entity categories, such as people, organisations or locations, inside unrestricted text. It was originally defined as a subtask of Information Extraction (IE) in the Message Understanding Conferences (MUC)[1], competitions in which most of the early research in NE recognition was accomplished (MUC6 95; MUC7 98). Since then, there have appeared many new applications of NE recognition, including Text Summarisation, Question Answering, Ontology Population with instances, and semantic annotation for ontology-based search engines.

Therefore, the interest on improving the current technology has not decreased. The MUC competitions have been followed by the CoNLL-2002[2] and CoNLL-2003[3] conferences (Tjong-Kim-Sang & Meulder 03), which addressed NE recognition from texts written in Spanish, Dutch, English and German; and the Automatic Content Extraction (ACE) program[4], which includes tasks on Entity Detection and Tracking across documents, and Time Expressions Recognition and Normalisation.

We can distinguish three types of systems for NE Recognition:

- Knowledge-based systems, which are based in the use of rules, patterns or grammars (Califf 98; Soderland 99; Freitag 00; Maynard *et al.* 02; Arevalo *et al.* 04). These rules are generally handcrafted, so there exist pattern languages, such as JAPE (Cunningham *et al.* 02), that simplify the task of writing the rules and the parsers.
- Those that apply Machine Learning techniques, either alone or in combination, including Memory-Based Learning, Maximum Entropy models and Hidden Markov Models (Freitag & McCallum 00; Klein *et al.* 03; Florian *et al.* 03; Kozareva *et al.* 05), Error-Driven Transformation-Based Learning (Black & Vasilakopoulos 02), boosting algorithms (Carreras *et al.* 03), and Support Vector Machines (Isozaki & Kazawa 02; Mayfield *et al.* 03; Li *et al.* 05).
- Those that combine knowledge-based methods and ML techniques (Mikheev *et al.* 98; Mikheev *et al.* 99).

In the MUC-7 competition (MUC7 98), the best results were obtained by (Mikheev *et al.* 98). It attained a precision of 93.39%, close to the 96.95% obtained by the worst human annotator. It can be considered a hybrid system with several stages. In the first one, the words in the texts were looked up in gazetteers (lists of common place, people and organisation names). Every time a candidate entity was found in a list, the system applied rules with a very high precision (they call *sure-fire rules*) before annotating it. For instance, the following rules

```
Xxxxx+ is a?  JJ* PROF
   shares of Xxxxx+
      Xxxxx+ area
```

indicate that one or more consecutive capitalised words can be considered as a person, an organisation or a location if they appear in the place of `Xxxxx+` in these three patterns, respectively. If the system, for example, found the word *Washington*, which may appear in the gazetteers for people and locations, it checked whether the context matched any of the rules for people and for places. If it is seen in the phrase *in the Washington area*, then the sure-fire rule for locations triggers, and it is marked up. After the sure-fire rules had applied on the corpus, the system continues with a maximum-entropy model.

In summary, we can consider that there are some contexts which strongly indicate the presence of a particular Named Entity. Therefore, the approach of using these *sure-fire rules* to annotate entities before

[1] http://www.itl.nist.gov/iaui/894.02/related_projects/muc/proceedings/muc_7_toc.html
[2] http://www.cnts.ua.ac.be/conll2002/ner/
[3] http://www.cnts.ua.ac.be/conll2003/ner/
[4] http://www.nist.gov/speech/tests/ace/

executing a second step, possibly based on Machine Learning techniques, seems very sound. However, a disadvantage of the pattern-based annotation is that the rules are generally designed manually and, thus, they should be difficult to port to new domains and to new kinds of entities.

This work addresses that problem by proposing a new algorithm for automatically extracting and generalising the sure-fire rules when a training corpus is available. The objective is to find a set of high-precision patterns with which a few Named Entities inside a corpus can be annotated, so the corpus provided to the ML techniques already contains some entities identified and classified inside it. This new approach will be incorporated soon to the *wraetlic* tools[5].

This paper is structured as follows: first, Section 2 describes the procedure used for extracting and generalising the sure-fire rules; next Section 3 describes the evaluation performed and the results obtained. Finally, Section 4 summarises the conclusions and describes open lines for future work.

2 Procedure

The general procedure to obtain the *sure-fire* rules is the following: first of all, for every appearance of a kind of entity (e.g. people) in the training corpus, we extract its context window. Next, in a second step, we generalise those contexts to obtain the patterns shared by them. The following sections describe these steps in detail.

2.1 Pattern extraction

The purpose of the first step is to extract a set of very specific context patterns obtained from the training corpus. We have worked with the CoNLL-2003 English dataset, which was built from the REUTERS corpus. In this dataset, all the words are annotated with part-of-speech tags. The sentences are chunked in noun, verb, prepositional and adverbial phrases, and four kinds of entities are annotated: person (`PER`), organisation (`ORG`), location (`LOC`), and miscellaneous names (`MISC`).

In this step, we simply look for all the instances of each of the four entity types, and extract a context around them. The context we have used is a window that includes up to five words to the left of the entity, and five words to its right. The window never jumps over sentence boundaries. For this experiment, we have only considered the words and their p-o-s tags; chunking information is discarded. Figure 1 shows several example contexts extracted for each of the four kinds of entities. The abbreviations `-BOS-` and `-EOS-` mark the beginning-of-sentence and the end-of-sentence, respectively.

Furthermore, for each entity type, we also collect, from the training corpora, the sequences of part-of-speech tags of every known entity. So, for instance, `NNP NNP` is a common sequence for people, as they are usually represented with two proper names (first name and family name), and `NNP CC NNP` will be a common sequence for organisations, as they sometimes include conjunctions in their names.

[5] http://www.ii.uam.es/~ealfon/eng/research/wraetlic.html

2.2 Pattern generalisation (I): Algorithm

The previous patterns might be directly applied on the test corpus to annotate entities inside it. However, for the moment, most of them are far too specific and the probability that we will find exactly the same 10-word context in a new text is very small. Therefore, we would like to substitute the patterns that have commonalities with more general patterns with a larger coverage. The following algorithm is used in order to generate the final set of generalised patterns:

1. Store all the patterns in a set $\mathcal{P}$.
2. Initialise a set $\mathcal{D}$ as an empty set.
3. While $\mathcal{P}$ is not empty,
   (a) For each possible pair of patterns, calculate the distance between them (described in the next section).
   (b) Take the two patterns with the smallest distance, p_i and p_j.
   (c) Remove them from $\mathcal{P}$.
   (d) Obtain the generalisation of both, p_g.
   (e) If the precision of p_g in the training corpus is over a threshold θ, add p_g to $\mathcal{P}$.
   Otherwise, add p_1 and p_2 to $\mathcal{D}$.
4. Return $\mathcal{D}$

The previous algorithm is repeated, separately, to generalise the patterns for people, locations, organisations and miscellaneous names. The output is the set containing all the rules that have been obtained by combining pairs of original rules. The purpose of the parameter θ is to ensure that we do not generalise patterns that are too different, resulting in rules that match in many places in the texts with a low precision. If we set θ to a high value, say 0.9, we are guiding the search towards high-precision rules; on the other hand, if we set it to a lower value, e.g. 0.75, then we'll obtain a smaller set of rules, which will have a larger coverage but a lower precision.

The next sections describes how the distance between the patterns is calculated for step (3a), and the procedure to generalise them in step (3d).

2.3 Pattern generalisation (II): Edit distance calculation

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

The procedure used to obtain a similarity metric between two patterns, consists of a slightly modified version of the dynamic programming algorithm for *edit-distance* calculation (Wagner & Fischer 74). This procedure has already been used successfully to extract

Person:

by/IN Hendrix/NNP 's/POS former/JJ girlfriend/NN ENTITY ,/, who/WP lived/VBD with/IN him/PRP
green/JJ light/NN to/TO Prime/NNP Minister/NNP ENTITY to/TO call/VB snap/VB elections/NNS ,/,
elections/NNS ,/, its/PRP$ general/JJ secretary/NN ENTITY told/VBD reporters/NNS ./. -EOS-/-EOS-
-BOS-/-BOS- ENTITY ,/, who/WP as/IN Israel/NNP 's/POS
-BOS-/-BOS- ENTITY is/VBZ winding/VBG up/RP his/PRP$ term/NN
He/PRP will/MD be/VB replaced/VBN by/IN ENTITY ,/, a/DT former/JJ Israeli/JJ envoy/NN

Location:

China/NNP 's/POS top/JJ negotiator/NN with/IN ENTITY ,/, Tang/NNP Shubei/NNP ,/, as/IN
-BOS-/-BOS- ENTITY accused/VBD Israel/NNP on/IN Wednesday/NNP of/IN
Iraqi/JJ forces/NNS were/VBD ousted/VBN from/IN ENTITY in/IN the/DT 1991/CD Gulf/NNP War/NNP
positions/NNS in/IN Qasri/NNP region/NN in/IN ENTITY province/NN near/IN the/DT Iranian/JJ border/NN
expected/VBN to/TO travel/VB to/TO the/DT ENTITY before/IN Monday/NNP ,/, "/" Nabil/NNP
minister/NN Shimon/NNP Peres/NNP in/IN the/DT ENTITY town/NN of/IN Ramallah/NNP on/IN Thursday/NNP

Organisation:

the/DT talks/NNS ,/, the/DT official/NN ENTITY news/NN agency/NN quoted/VBN Tang/NNP Shubei/NNP
executive/JJ vice/NN chairman/NN of/IN the/DT ENTITY ,/, as/IN saying/VBG late/RB on/RB
the/DT year-earlier/JJ period/NN ,/, the/DT ENTITY said/VBD on/IN Thursday/NNP ./. -EOS-/-EOS-
-BOS-/-BOS- ENTITY won/VBD 77,719/CD registrations/NNS ,/, slightly/RB
-BOS-/-BOS- Third/JJ was/VBD ENTITY with/IN 35,563/CD registrations/NNS ,/, or/CC

Miscellaneous:

-BOS-/-BOS- ENTITY farmers/NNS denied/VBN on/IN Thursday/NNP there/EX
,/, but/CC expressed/VBD concern/NN that/IN ENTITY government/NN advice/NN to/TO consumers/NNS to/TO
to/TO Ukraine/NNP this/DT week/NN by/IN ENTITY Vice/NNP President/NNP Lien/NNP ./. -EOS-/-EOS-
-BOS-/-BOS- ENTITY July/NNP car/NN registrations/NNS up/RB 14.2/CD

Figure 1: Example patterns extracted from the training corpus for each kind of entity.

patterns for identifying hyperonymy and meronymy relationships inside text (Ruiz-Casado *et al.* 05). The *edit distance* between two strings A and B is defined as the minimum number of changes (character insertion, addition or replacement) that have to be done to the first string in order to obtain the second one. The algorithm can be implemented as filling in a matrix $\mathcal{M}$ with the following procedure:

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

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

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

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

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

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

In these equations, $\mathcal{M}$[i,j] will contain the edit distance between the first i elements of A and the first j elements of B. Equation (1a) indicates that, if A and B are both empty strings, the edit distance should be 0. Equations (1b) and (1c) mean that the edit distance between an empty string, and a string with N symbols must be N. Finally, equation (1d) uses the fact that, in order to obtain a string[6] $A\sigma$ from a string $B\gamma$, we may proceed in three possible ways:

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

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

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

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

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

- Replacing `a` by `nice`.

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

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

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

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

Figure 2: Example of the edit distance algorithm. A and B are two word patterns; $\mathcal{M}$ is the matrix in which the edit distance is calculated, and $\mathcal{D}$ is the matrix indicating the choice that produced the minimal distance for each cell in $\mathcal{M}$.

- Removing `kind`.

2.4 Pattern generalisation (III): Algorithm

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

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

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

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

This pattern may match phrases such as *It is a kind of*, *It is nice of*, *It is a subset of*, or *It is a type of*. As can be seen, the generalisation of these two rules produces one that can match a wide variety of sentences, so we should always take care in order not to over-generalise.

2.5 Pattern generalisation (IV): Generalisation with part-of-speech tags

As shown in the previous example, sometimes, when two patterns are combined, the result is too general and matches more contexts than expected. Part-of-speech tags have been used to modify the edit distance calculation, in a way such that the edit distance of two patterns which do not differ in their sequences of part-of-speech tags will remain small even though their words are all different.

Our patterns are, therefore, sequences of terms annotated with part-of-speech labels, as in the following examples:

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

The calculation is modified in the following way: the system only allows replacement actions if the words from the two patterns A and B belong to the same general part-of-speech (nouns, verbs, adjectives, adverbs, etc.). Also, if this is the case, we consider that there is no edit distance between the two patterns. The d function, therefore, is redefined as:

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

The insertion and deletion actions are defined as before. Therefore, patterns (a) and (b) above would have an edit distance of 3: two deletions, `a` and `kind`, and one insertion, `nice`. Note that it is not possible to do any replacement, because those words have different p-o-s tags. The result of their generalisation is:

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

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

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

2.6 Application of the generalised patterns for NE recognition

Finally, given a set of patterns for a particular named entity, the procedure for annotating is straightforward:

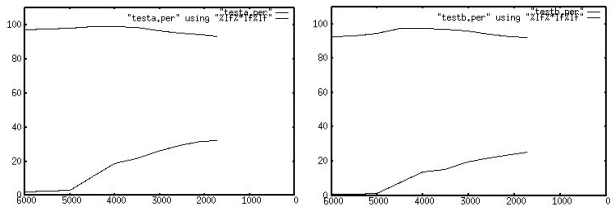


Figure 3: Recall and precision identifying people on test sets A and B, depending on the number of rules during the generalisation.

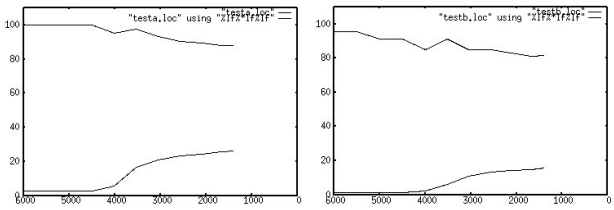


Figure 4: Recall and precision identifying locations on test sets A and B, depending on the number of rules during the generalisation.

1. For any of the rules in the set $\mathcal{D}$, defined for a particular entity type,
2. For each sentence in the corpus:
   (a) Look for the left-hand side of the rule in the sentence.
   (b) Look for the right-hand side of the rule afterwards in the sentence.
   (c) Take the words that are in between. If the sequence of part-of-speech tags has been seen in the training corpus for that kind of entity, annotate it.

For instance, the following pattern

```
,/, ENTITY announced|said/VBD */* I|We|he|it/PRP
came|did|have|think|wanted/VBD
```

matches with the sentence *Today, John Smith announced Mary he wanted a car*. First of all, it is necessary to find, in a sentence, a comma. Later on, the program finds that *announced Mary he wanted* matches the last part of the pattern. In between we find the words *John Smith*, with p-o-s tags `NNP NNP`, which is valid for people according to the training corpus. Therefore, *John Smith* will be annotated as a person inside that sentence.

3 Evaluation and results

We have tried the rules on the CoNLL evaluation data in several experiments:

- Applying the whole set of rules obtained with the rule generalisation procedure.
- Applying a pruned set of rules that only included those that applied at least a certain number of times on the training data.
- Applying the pruned set of rules, and adding a simple heuristic about a few words that were clearly mis-tagged.

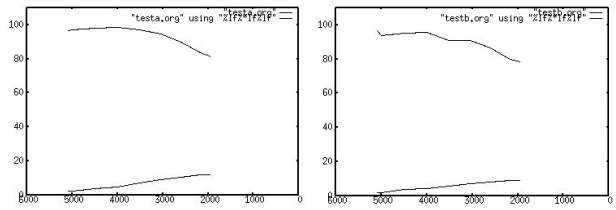


Figure 5: Recall and precision identifying organisations on test sets A and B, depending on the number of rules during the generalisation.

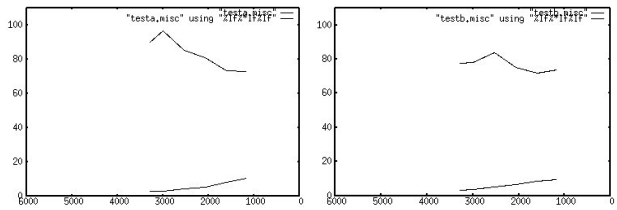


Figure 6: Recall and precision identifying miscellaneous entities on test sets A and B, depending on the number of rules during the generalisation.

The results are described in the following sections.

Direct application of the rules After extracting the patterns from the CoNLL training data, they have been generalised using the previous algorithm. We have set the threshold θ to 0.9, because we want to obtain patterns with high precision. In this way, we can be sure that all the rules obtained at the end have a precision of at least 0.9 when applied on the training corpus.

The CoNLL competition provided two different test corpora, named A and B. Figures 3, 4, 5, and 6 show how the precision and the recall of the set of patterns varies as we generalise them.

For instance, in the case of people, the system starts with roughly 6000 very specific patterns, which, applied on test set A, have a very high precision (96.88%), but a very low recall (1.68%). Note that the precision is not 100%, i.e. a few patterns may make mistakes. This is because not all the patterns contain a 10-word window if the Named Entity was very near the sentence boundaries. In the extreme case, an extracted pattern might be

```
-BOS-/-BOS- ENTITY -EOS-/-EOS-
```

which will surely have a low precision on any corpus.

As the patterns are being generalised by twos, the total number of patterns in the set decreases. It can be seen that precision drops slightly down to 93.5%, and recall increases up to 32.19%. With test set B, the performance is rather similar.

The case of locations is also alike to that of people. In test set A, when the system finishes generalising the patterns, the resulting set of rules attains a precision of 87.8% and a recall of 26.35%. In this case as well we can consider that the precision of the rules has been preserved from the training set to the test sets, as it is very near the value of 90% that we had set as threshold

for the generalisation. The results for test set B are slightly worse, 81.5% precision and 15.9% recall.

The results obtained by the rules for organisations also behave in the same way: before generalising, precision is near 95% and recall is very small; and, as we proceed, recall improves and precision decreases. With the final set, the precisions obtained are 81.4% on set A, and 78.3% on set B; and the recall is 11.8% and 8.9%, respectively.

Finally, the patterns for the miscellaneous entities have proven the most difficult to learn. This is the only case in which the most specific patterns, without any generalisation, do not have a very high precision. For instance, in test B, the original patterns obtained from the training corpus just attain a precision of 77.78%. After they have been generalised, precision drops to 73%, and recall reaches 10.5%. This set is particularly difficult because it does not contain a very precise kind of entities; it includes things such as names of the inhabitants of countries (e.g. German) or illnesses (e.g. Bovine Spongiform Encephalopathy).

Application of the pruned set of rules In a second experiment, we have pruned the obtained set of rules to remove those that applied just a few number of times in the training corpus. The motivation for this experiment is that, if a rule applies many times in the training corpus with a very high precision, then we can be confident that it will continue having a high precision in the test corpus; whereas if the rule has only applied once in the training corpus, we have much less evidence to assert that it will extract correct entities in other corpus.

We have varied the threshold for discarding a rule from 1 to 8. In this way, in the first run, we keep all the patterns that applied more than once in the training corpus; in the second run, those that applied more than twice; and so on. The results can be seen in Table 1, columns 5 and 6. The pruning increases the precision for all entities, at the cost of a small decrease in recall.

Application with a heuristic Finally, a manual observation of the results allowed us to identify the most common errors for people and for locations. Firstly, the obtained patterns may extract mistakenly people's titles together with their names. For instance, the pattern

`17-year-old/JJ ENTITY and|or/CC`

may extract, from the sentence

`17-year-old Secretary John Smith and`

the entity `Secretary John Smith`, while it will be tagged in the test set as just `John Smith`. Secondly, in the case of locations, we found that some patterns for locations were able to extract weekday and month names as well. Therefore, in this last run, we post-processed the corpus to remove all the people titles (found in a list) and weekday and month names. The results for test set A are shown in Table 1. It

				No heuristic		Heuristic	
Type	**T**	**Patterns**		**Prec.**	**Recall**	**Prec.**	**Recall**
PER	0	1720	1720	93.53	32.19	95.40	32.63
	1	1720	391	94.20	29.10	95.41	29.37
	2	1720	262	94.46	28.72	95.52	28.94
	3	1720	199	95.13	28.61	96.20	28.83
	4	1720	157	95.63	28.50	96.71	28.72
	5	1720	120	95.77	28.28	96.86	28.50
	6	1720	102	96.12	28.23	97.22	28.45
	7	1720	93	96.46	28.07	97.57	28.28
	8	1720	81	96.64	28.07	**97.75**	28.28
LOC	0	1387	1387	87.84	26.35	90.98	26.35
	1	1387	382	93.43	20.14	95.36	20.14
	2	1387	257	93.15	19.98	95.08	19.98
	3	1387	183	93.49	19.54	95.48	19.54
	4	1387	146	93.72	19.49	95.72	19.49
	5	1387	115	93.93	19.38	95.96	19.38
	6	1387	89	93.90	19.27	95.93	19.27
	7	1387	76	94.10	19.11	95.90	19.11
	8	1387	67	94.57	18.94	**96.13**	18.94
ORG	0	1943	1943	81.44	11.78	81.96	11.86
	1	1943	425	80.00	8.95	80.67	9.02
	2	1943	266	80.42	8.58	81.12	8.65
	3	1943	191	80.77	7.83	81.54	7.90
	4	1943	146	80.00	7.16	80.83	7.23
	5	1943	120	80.83	7.23	81.67	7.31
	6	1943	102	83.64	6.86	84.55	6.94
	7	1943	82	86.41	6.64	87.38	6.71
	8	1943	70	88.12	6.64	89.11	6.71
MISC	0	1166	1166	72.93	10.52	72.93	10.52
	1	1166	271	78.21	6.62	78.21	6.62
	2	1166	184	78.21	6.62	78.21	6.62
	3	1166	141	78.08	6.18	78.08	6.18
	4	1166	111	79.41	5.86	79.41	5.86
	5	1166	89	85.48	5.75	85.48	5.75
	6	1166	70	85.25	5.64	85.25	5.64
	7	1166	53	84.48	5.31	84.48	5.31
	8	1166	48	87.27	5.21	87.27	5.21

Table 1: Results after pruning the sets of rules, on test set A. Columns indicate entity type, the threshold (T) for the pruning, and the precision and the recalled obtained with the pruned set of rules with or without the heuristic. T=0 means no pruning.

can be seen that the precision of the rules for people and locations are somewhat improved, and become similar to Mikheev's results (99% for people and 96% for locations after just the sure-fire rules; 97% for people and 93% for locations after the complete run of his system). In a few cases, the precision for organisations changes slightly, due to the fact that something that has been untagged as a person or a location may next be tagged as an organisation if a pattern for ORG matches with its context.

The final results for test set B are similar: for people, the precision is 97.07%. In the case of locations, the precision obtained was worse (88.70%), due to a single pattern that classified all football teams as locations. The removal of that pattern boosts the precision for locations up to 96.88%.

4 Conclusions and future work

This paper describes a procedure for obtaining from a corpus, and next generalising, automatically, patterns that can be used as *sure-fire* rules in an Information Extraction system, in a similar way as the one described in (Mikheev *et al.* 98). The system obtains the generalised patterns for one entity in ∼3 hours in a Pentium 2.4GHz, and the test corpora can be tagged in less than one minute.

We have shown that, in our case, the patterns learnt for people and for locations have a very high precision, combined with a simple heuristic, even without the use

of gazetteers of people and location names. (Mikheev *et al.* 99) quantifies the precision of a sure-fire rule in the rage 96-98%, so these fall inside the interval. In the case of organisations and miscellaneous entities, precision is just 87-89%, so it might be better to use them combined with gazetteers to increase their precisions. At this point of the work, we may only recommend to use the system for people and locations, as the patterns for the other entities should be improved a little more. On the other hand, the recall is low, probably due to a sparse-data problem. We believe that the training corpus includes a minimal fraction of all the possible contexts in which an entity can appear.

Concerning these results, it should be noted that (Mikheev *et al.* 98) just applied the *sure-fire* rules when the entity that matched the rule *also* appeared in a gazetteer. (Mikheev *et al.* 99) reports that the precision decreases somewhat if a small gazetteer is used, and may decrease very much (in the case of locations) if no gazetteer is present. In our case, even without using a gazetteer, precision is very high for people and locations, and rather high for organisations. We expect that, combined in this way with lists of people, companies and place names, the precision of the rules will be even higher than the one obtained.

For future work, we would like to extend the generalisation procedure, to see if both the precision and recall of the rules can be further improved if a more expressive encoding is used. Along this line, we plan (a) to be able to substitute disjunctions of words with the same part-of-speech, such as `big|small|large/JJ` by any word of that p-o-s, `?/JJ`, something that is currently not implemented; (b) to extend the generalisation with semantic classes, so patterns such as

```
The/DT Spain|France|Italy|Japan/NNP Minister/NNP
```

can be substituted by

```
The/DT {country:1}/NNP Minister/NNP
```

using the hyperonymy relationship in WordNet, in a similar way as in (Soderland 99); (c) to test the system in unsupervised settings. From a set of seed words that we know to pertain to a given Named Entity class (e.g. person names or locations names), we could learn and generalise the patterns and, using a bootstrapping procedure, apply those patterns to augment the set of seed words from which to extend the set of patterns; (d) to test the effect of gazetteers on the accuracy of the learnt rules; and (e) to test the influence of this system as an initial step in a complete system.

5 Acknowledgements

This paper has been funded by CICYT, project numbers TIC2002-01948 and TIN2004-03140. We wish to thank the anonymous reviewers for their comments, which have been very useful for improving this paper.

References

(Arevalo *et al.* 04) M. Arevalo, M. Civit, and M. A. Martí. MICE: A module for named entity recognition and clasification. *International Journal of Corpus Linguistics*, 9(1):53–68, 2004.

(Black & Vasilakopoulos 02) W. J. Black and A. Vasilakopoulos. Language-independent named entity classification by modified transformation-based learning and by decision tree induction. In *Proceedings of CoNLL-2002*, pages 159–162, Taipei, Taiwan, 2002.

(Califf 98) Mary Elaine Califf. *Relational Learning Techniques for Natural Language Extraction*. PhD thesis, University of Texas at Austin, 1998.

(Carreras *et al.* 03) X. Carreras, L. Márquez, and L. Padró. A simple named entity extractor using adaboost. In *Proceedings of CoNLL-2003*, pages 152–155, Edmonton, Canada, 2003.

(Cunningham *et al.* 02) H. Cunningham, D. Maynard, K. Bontcheva, V. Tablan, and C. Ursu. The GATE user guide, 2002.

(Florian *et al.* 03) R. Florian, A. Ittycheriah, H. Jing, and T. Zhang. Named entity recognition through classifier combination. In *Proceedings of CoNLL-2003*, pages 168–171, Edmonton, Canada, 2003.

(Freitag & McCallum 00) Dayne Freitag and Andrew McCallum. Information extraction with HMM structures learned by stochastic optimization. In *AAAI/IAAI*, pages 584–589, 2000.

(Freitag 00) Dayne Freitag. Machine learning for information extraction in informal domains. *Machine Learning*, 39(2-3):169–202, 2000.

(Isozaki & Kazawa 02) Hideki Isozaki and Hideto Kazawa. Efficient support vector classifiers for named entity recognition. In *Proceedings of the 19th international conference on Computational Linguistics*, pages 1–7, Morristown, NJ, USA, 2002. Association for Computational Linguistics.

(Klein *et al.* 03) D. Klein, J. Smarr, H. Nguyen, and C. Manning. Named entity recognition with character-level models. In *Proceedings of CoNLL-2003*, pages 180–183, Edmonton, Canada, 2003.

(Kozareva *et al.* 05) Z. Kozareva, O. Ferrández, and A. Montoyo. Combining data-driven systems for improving named entity recognition. In *Natural Language Processing and Information Systems*, volume 3513 of *Lecture Notes in Computer Science*, pages 80–90. Springer, 2005.

(Li *et al.* 05) Y. Li, K. Bontcheva, and H. Cunningham. Using uneven margins SVM and perceptron for information extraction. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, pages 72–79, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.

(Mayfield *et al.* 03) James Mayfield, Paul McNamee, and Christine Piatko. Named entity recognition using hundreds of thousands of features. In Walter Daelemans and Miles Osborne, editors, *Proceedings of CoNLL-2003*, pages 184–187. Edmonton, Canada, 2003.

(Maynard *et al.* 02) D. Maynard, H. Cunningham, K. Bontcheva, and M. Dimitrov. Adapting a robust multi-genre ne system for automatic content extraction. In *Artificial Intelligence: Methodology, Systems, and Applications*, volume 2443 of *Lecture Notes in Artificial Intelligence*, pages 264–273. Springer-Verlag, 2002.

(Mikheev *et al.* 98) A. Mikheev, C. Grover, and M. Moens. Description of the ltg system used for muc-7. In *Proceedings of 7th Message Understanding Conference (MUC-7)*, 1998.

(Mikheev *et al.* 99) A. Mikheev, M. Moens, and C. Grover. Named entity recognition without gazeteers. In *Proceedings of the Ninth Conference of the European Chapter of the Association for Computational Linguistics*, pages 1–8, Bergen, Norway, 1999.

(MUC6 95) MUC6. *Proceedings of the 6^{th} Message Understanding Conference (MUC-6)*. Morgan Kaufman, 1995.

(MUC7 98) MUC7. *Proceedings of the 7^{th} Message Understanding Conference (MUC-7)*. Morgan Kaufman, 1998.

(Ruiz-Casado *et al.* 05) M. Ruiz-Casado, E. Alfonseca, and P. Castells. Automatic extraction of semantic relationships for wordnet by means of pattern learning from wikipedia. In *Proceedings of NLDB-05*, 2005.

(Soderland 99) S. Soderland. Learning information extraction rules for semi-structured and free text. *Machine Learning*, 34(1–3):233–272, 1999.

(Tjong-Kim-Sang & Meulder 03) E. F. Tjong-Kim-Sang and F. De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL-2003*, pages 142–147, Edmonton, Canada, 2003.

(Wagner & Fischer 74) R.A. Wagner and M.J. Fischer. The string-to-string correction problem. *Journal of Assoc. Comput. Mach.*, 21, 1974.

Intelligence Analysis through Text Mining

Alessandro ZANASI
Via G.B.Amici, 29 - 41100 Modena - Italy
Phone Number: +39 059 237634
Fax Number: +39 059 220093
Email: alessandro.zanasi@temis-group.com

Overview

In a world where all the organizations (from government and from corporate world), thanks to internet, are "global" organizations, only those ones with the knowledge to answer the global needs may survive to a fierce global struggle. The amount of this new knowledge available through the Web, the Intranets and other sources is increasing, but the capacity of reading and analyzing it remains constant. Only the organization who knows how to retrieve, analyze and turn into actionable intelligence documents, web pages and, generally, the so called *open sources*, are able to acquire and maintain a competitive advantage.

The world of intelligence is reshaping itself, since that the world is requiring a different intelligence: dispersed, not concentrated; open to several sources; sharing its analysis with a variety of would-be partners, without guarding its secrets tightly. And open to the contribution with the best experts, also outside government or corporation [1].

The role of *competitive intelligence* has assumed great importance not only in the corporate world but also in the government one, largely due to the changing nature of national power. Today the success of foreign policy rests to a significant extent on industrial and financial power, and industrial and financial power in turn is dependent on science and technology and business factors [2].

Intelligence has long been deemed necessary for government agencies as well as for corporate enterprises that require up-to-date, accurate data. The need for true intelligence has become increasingly crucial because of information explosion.

Information density doubles every 24 months and its costs are halved every 18 months, creating a non linear increase in technological advances supporting the global information infrastructure [3]. Too much information is available, and too much is contradictory; it is necessary a help coming from information technology to process this information and to extract knowledge.

These are the reasons by which the technologies which allow the reduction of information complexity and overload have become important in the government as in the corporate intelligence world.

The information revolution is the key enabler of economic globalization. It was the information revolution that undid the Soviet Union, which could produce roads and dams thanks to planning and brute force, but could not induce innovation in computer chips.

The age of information is also the emergence of the so called *market-state* [4]. Where the various competing systems of the contemporary nation state all took their legitimacy from the promise to better the material welfare of their citizens, the market state offers a different covenant: it will maximize the opportunity of its people.

The global society of market states will face lethal security challenges that its habits of intense competition do not naturally suit it to deal with and will dramatically change the roles of government and of private actors and of intelligence.

Currently governments power is being challenged from both above (international commerce, which erodes what used to be thought of as aspects of national sovereignty) and below (terrorist and organized crime challenge the state power from beneath, by trying to compel states to acquiesce or by eluding the control of states).

Tackling these new challenges is the role of the new *government intelligence*.

We present here some methodologies and results obtained by different organizations (from corporate and government world) using text mining technology.

References

[1] Treverton G.F., Reshaping National Intelligence in an Age of Information, Cambridge University Press, 2001.
[2] Bottom N.R., Gallati R.R.J., Industrial Espionage: Intelligence Techniques and Countermeasures, Stoneham, MA, Butterworth, 1984.
[3] Lisse W., The Economics of Information and the Internet, Competitive Intelligence Review, Vol.9 (4), 1998.
[4] Bobbitt P., The Shield of Achilles: War, Peace, and the Course of History, Knopf, 2002.

Biography

- ESRAB – European Security Research Advisory Board member
- University Professor (Modena University and Malta University in Roma, where he is coordinating its Intelligence and Security Study Center, under the Presidency of Hon. Scotti, ex-Interior and Foreign Affairs minister of Italian Government).
- He graduated in nuclear engineering at Bologna Univ., specialized in probability theory and financial engineering in Paris and in business administration in Modena.
- Carabinieri officer (Ret.) charged of electronic intelligence at Rome Scientific Investigations Center, professional consultant for Italian courts and information broker.
- IBM, where he spent 16 years holding different consulting and research

positions in business intelligence in Italy, France (Paris) and USA (San Jose), leaving as IBM Market Intelligence analysis responsible in South Europe.
- Bologna Data Mining Center (http: open.cineca.it/datamining) coordinator. This centre was born as a partnership between IBM and Cineca (Italian government supercomputer center) to deepen the knowledge in data mining applications.
- META Group Inc. Int'l Program Director, analyst in business intelligence and CRM areas.
- TEMIS Text Mining Solutions SA co-founder in 2000.
- SCIP (Society of Competitive Intelligenec Professionals) European Advisory Board member and Rome chapter coordinator.
- Since 1998 he is European Commission reviewer and evaluator of proposals and projects.
- Fluent in 4 languages, for more than twenty years, as an intelligence analysis specialist, he has been teaching, working and consulting for corporations and governments and giving public and private workshops in Europe, North and South America.

Past Experience

- Modena and Reggio Emilia, and Bologna, University Professor ("Data Analysis and Data Mining" course)
- University of Malta Master in Intelligence professor ("Business Intelligence" and "Applications of Business Intelligence" courses)
- Chairman and/or speaker at several workshop and/or conferences on intelligence or business intelligence topics (the next chairmanship: "Data and Text Mining and their Business Applications" (*7th Int'l Conference on Data Mining*) – Prague, Czech Republic – (July, 11-13) http://www.wessex.ac.uk/conferences/2006/data06/)

Relevant publications

- Author or editor of seven books on quantitative intelligence (available through Amazon.com) in English. Among them:
    - "Text Mining and its Applications to Intelligence, CRM and KM", Wit Press, 2005
    - "Discovering Data Mining", Ed.Prentice Hall, 1998;
- Several papers on intelligence topics. Among them:
    - "Open sources automatic analysis for corporate and government intelligence" Wit Press, 2005
    - "Virtual Communities: Human Capital and Other Personal Characteristics Extraction", Wit Press, 2005
    - "Email, chatlines, forums, newsgroups: a continuos opinion surveys source thanks to text mining" – This paper received the Best Paper Award and was published on "Excellence in Research", 2003, Esomar, Holland
    - "Competitive Intelligence through data mining public

sources", Competitive Intelligence Review, March 1998, John Wiley

Relevant presentations

- "Text Mining for corporate and government intelligence" presented at Poitiers (France, 27-28 Jan.2005) at 1st European Symposium on Competitive Intelligence, under the patronage of French 1st Minister (Mr.Raffarin).
-
- "Text Mining for Automatic Intelligence" – workshop presented at SCIP Europe Conference – London, 2003, October
- "New forms of war, new forms of intelligence: Text Mining" – presented in 2001, March at "Intelligence in XXI century", Priverno (Italy) under the patronage of Italian Presidency of Republic

Intelligence Analysis through Text Mining

Alessandro ZANASI
Via G.B.Amici, 29 - 41100 Modena - Italy
Phone Number: +39 059 237634
Fax Number: +39 059 220093
Email: alessandro.zanasi@temis-group.com

Overview

In a world where all the organizations (from government and from corporate world), thanks to internet, are "global" organizations, only those ones with the knowledge to answer the global needs may survive to a fierce global struggle.
The amount of this new knowledge available through the Web, the Intranets and other sources is increasing, but the capacity of reading and analyzing it remains constant. Only the organization who knows how to retrieve, analyze and turn into actionable intelligence documents, web pages and, generally, the so called *open sources*, are able to acquire and maintain a competitive advantage.

The world of intelligence is reshaping itself, since that the world is requiring a different intelligence: dispersed, not concentrated; open to several sources; sharing its analysis with a variety of would-be partners, without guarding its secrets tightly. And open to the contribution with the best experts, also outside government or corporation [1].

The role of *competitive intelligence* has assumed great importance not only in the corporate world but also in the government one, largely due to the changing nature of national power. Today the success of foreign policy rests to a significant extent on industrial and financial power, and industrial and financial power in turn is dependent on science and technology and business factors [2].

Intelligence has long been deemed necessary for government agencies as well as for corporate enterprises that require up-to-date, accurate data. The need for true intelligence has become increasingly crucial because of information explosion.

Information density doubles every 24 months and its costs are halved every 18 months, creating a non linear increase in technological advances supporting the global information infrastructure [3]. Too much information is available, and too much is contradictory; it is necessary a help coming from information technology to process this information and to extract knowledge.

These are the reasons by which the technologies which allow the reduction of information complexity and overload have become important in the government as in the corporate intelligence world.

The information revolution is the key enabler of economic globalization. It was the information revolution that undid the Soviet Union, which could produce roads and dams thanks to planning and brute force, but could not induce innovation in computer chips.

The age of information is also the emergence of the so called *market-state* [4]. Where the various competing systems of the contemporary nation state all took their legitimacy from the promise to better the material welfare of their citizens, the market state offers a different covenant: it will maximize the opportunity of its people.

The global society of market states will face lethal security challenges that its habits of intense competition do not naturally suit it to deal with and will dramatically change the roles of government and of private actors and of intelligence.

Currently governments power is being challenged from both above (international commerce, which erodes what used to be thought of as aspects of national sovereignty) and below (terrorist and organized crime challenge the state power from beneath, by trying to compel states to acquiesce or by eluding the control of states).

Tackling these new challenges is the role of the new *government intelligence.*

We present here some methodologies and results obtained by different organizations (from corporate and government world) using text mining technology.

References

[1] Treverton G.F., Reshaping National Intelligence in an Age of Information, Cambridge University Press, 2001.
[2] Bottom N.R., Gallati R.R.J., Industrial Espionage: Intelligence Techniques and Countermeasures, Stoneham, MA, Butterworth, 1984.
[3] Lisse W., The Economics of Information and the Internet, Competitive Intelligence Review, Vol.9 (4), 1998.
[4] Bobbitt P., The Shield of Achilles: War, Peace, and the Course of History, Knopf, 2002.

Biography

- ESRAB – European Security Research Advisory Board member
- University Professor (Modena University and Malta University in Roma, where he is coordinating its Intelligence and Security Study Center, under the Presidency of Hon. Scotti, ex-Interior and Foreign Affairs minister of Italian Government).
- He graduated in nuclear engineering at Bologna Univ., specialized in probability theory and financial engineering in Paris and in business administration in Modena.
- Carabinieri officer (Ret.) charged of electronic intelligence at Rome Scientific Investigations Center, professional consultant for Italian courts and information broker.
- IBM, where he spent 16 years holding different consulting and research

positions in business intelligence in Italy, France (Paris) and USA (San Jose), leaving as IBM Market Intelligence analysis responsible in South Europe.
- Bologna Data Mining Center (http: open.cineca.it/datamining) coordinator. This centre was born as a partnership between IBM and Cineca (Italian government supercomputer center) to deepen the knowledge in data mining applications.
- META Group Inc. Int'l Program Director, analyst in business intelligence and CRM areas.
- TEMIS Text Mining Solutions SA co-founder in 2000.
- SCIP (Society of Competitive Intelligenec Professionals) European Advisory Board member and Rome chapter coordinator.
- Since 1998 he is European Commission reviewer and evaluator of proposals and projects.
- Fluent in 4 languages, for more than twenty years, as an intelligence analysis specialist, he has been teaching, working and consulting for corporations and governments and giving public and private workshops in Europe, North and South America.

Past Experience

- Modena and Reggio Emilia, and Bologna, University Professor ("Data Analysis and Data Mining" course)
- University of Malta Master in Intelligence professor ("Business Intelligence" and "Applications of Business Intelligence" courses)
- Chairman and/or speaker at several workshop and/or conferences on intelligence or business intelligence topics (the next chairmanship: "Data and Text Mining and their Business Applications" (*7th Int'l Conference on Data Mining*) – Prague, Czech Republic – (July, 11-13) http://www.wessex.ac.uk/conferences/2006/data06/)

Relevant publications

- Author or editor of seven books on quantitative intelligence (available through Amazon.com) in English. Among them:
  - "Text Mining and its Applications to Intelligence, CRM and KM", Wit Press, 2005
  - "Discovering Data Mining", Ed.Prentice Hall, 1998;
- Several papers on intelligence topics. Among them:
  - "Open sources automatic analysis for corporate and government intelligence" Wit Press, 2005
  - "Virtual Communities: Human Capital and Other Personal Characteristics Extraction", Wit Press, 2005
  - "Email, chatlines, forums, newsgroups: a continuos opinion surveys source thanks to text mining" – This paper received the Best Paper Award and was published on "Excellence in Research", 2003, Esomar, Holland
  - "Competitive Intelligence through data mining public

sources", Competitive Intelligence Review, March 1998, John Wiley

Relevant presentations

- "Text Mining for corporate and government intelligence" presented at Poitiers (France, 27-28 Jan.2005) at 1st European Symposium on Competitive Intelligence, under the patronage of French 1st Minister (Mr.Raffarin).
-
- "Text Mining for Automatic Intelligence" – workshop presented at SCIP Europe Conference – London, 2003, October
- "New forms of war, new forms of intelligence: Text Mining" – presented in 2001, March at "Intelligence in XXI century", Priverno (Italy) under the patronage of Italian Presidency of Republic

Unsupervised text mining for designing a virtual web environment

Marie-Laure Reinberger
University of Antwerp - CNTS,
Universiteitsplein 1, B-2610 Wilrijk - Belgium,
tel.: +32-3- 820.2766; fax: +32-3-820.2762
marielaure.reinberger@ua.ac.be

Abstract

We report on an a set of experiments carried out in the context of the Flemish OntoBasis project. Our purpose is to extract semantic relations from text corpora in an unsupervised way and use the output as preprocessed material for the construction of ontologies from scratch.

We have worked on a corpus taken from Internet websites and describing the megalithic ruin of Stonehenge. Using a shallow parser, we select functional relations, such as the syntactic structure subject-verb-object. Those functional relations correspond to what we call a "lexon". The selection is done using prepositional structures and frequency measures in order to select the most relevant lexons. Therefore, the paper stresses the choice of patterns and the filtering carried out in order to discard automatically all irrelevant structures.

The adequacy of the relations extracted has been evaluated manually.

Keywords: unsupervised text mining, semantic relation extraction, ontology.

1 Introduction

The development of ontology-driven applications is currently slowed down due to the knowledge acquisition bottleneck. Therefore, techniques applied in computational linguistics and information extraction (in particular machine learning) are used to create or grow ontologies in a period as limited as possible with a quality as high as possible. Sources can be of different kinds including databases and their schemes - e.g. [Volz *et al.* 04], semi-structured data (XML, web pages), ontologies and texts.

This paper wants to report on a research effort on the learning of ontologies from texts during the Flemish IWT OntoBasis project [1]. The experiments concern the extraction of conceptual relationships. For this aim, the results of shallow parsing techniques are combined with unsupervised learning methods.

Although the results we are reporting here have been evaluated manually, the final purpose of this study is to establish an ontology-based method for designing virtual web environments.

2 Unsupervised learning

The aim of this study is to extract automatically semantic information from text in order to allow the building of a graphical representation of Stonehenge.

To get to this, our purpose is to build a repository of lexical semantic information from text, ensuring evolvability and adaptability. This repository can be considered as a complex semantic network. We assume that the method of extraction and the organisation of this semantic information should depend not only on the available material, but also on the intended use of the knowledge structure. There are different ways of organising this knowledge, depending on its future use and on the specificity of the domain. Currently, the focus is on the discovery of concepts and their conceptual relationships, although the ultimate aim is to discover semantic constraints as well. We have opted for extraction techniques based on unsupervised learning methods [Reinberger *et al.* 03; Reinberger *et al.* 04; Spyns *et al.* 04; Reinberger & Spyns 05] since these do not require specific external domain knowledge such as thesauri and/or tagged corpora [2]. As a consequence, the portability of these techniques to new domains is expected to be much better

[1] see http://wise.vub.ac.be/ontobasis

[2] Except the training corpus for the general purpose shallow parser.

3 Material and methods

The ***linguistic assumptions*** underlying this approach are

1. the principle of selectional restrictions (syntactic structures provide relevant information about semantic content), and

2. the notion of co-composition [Pustejovsky 95] (if two elements are composed into an expression, each of them impose semantic constraints on the other).

The fact that heads of phrases with a subject relation to the same verb share a semantic feature would be an application of the principle of *selectional restrictions*. The fact that the heads of phrases in a subject or object relation with a verb constrain that verb and vice versa would be an illustration of *co-composition*. In other words, each word in a noun-verb relation participates in building the meaning of the other word in this context [Gamallo *et al.* 01; Gamallo *et al.* 02]. If we consider the expression "write a book" for example, it appears that the verb "to write" triggers the informative feature of "book", more than on its physical feature. We make use of both principles in our use of clustering to extract semantic knowledge from syntactically analysed corpora.

In a specific domain, an important quantity of semantic information is carried by the nouns. At the same time, the noun-verb relations provide relevant information about the nouns, due to the semantic restrictions they impose. In order to extract this information automatically from our corpus, we used the ***memory-based shallow parser*** which is being developed at CNTS Antwerp and ILK Tilburg [Buchholz 02; Buchholz *et al.* 99; Daelemans *et al.* 99] [3]. This shallow parser takes plain text as input, performs tokenisation, POS tagging, phrase boundary detection, and finally finds grammatical relations such as subject-verb and object-verb relations, which are particularly useful for us. The software was developed to be efficient and robust enough to allow shallow parsing of large amounts of text from various domains.

Example:

[NP-SBJ-1 The/DT Sarsen/NNP lintels/NNS NP-SBJ-1] [VP-1 form/VBP VP-1] [NP-OBJ-1 a/DT continuous/JJ circle/NN NP-OBJ-1] PNP [PP around/IN PP] [NP the/DT top/NN NP] PNP ./.

Different methods can be used for the ***extraction of semantic information*** from parsed text. Pattern matching [Berland & Charniak 99] has proved to be an efficient way to extract semantic relations, but one drawback is that it involves the predefined choice of the semantic relations that will be extracted. On the other hand, clustering [Lin 98; Pantel & Lin 02] only requires a minimal amount of "manual semantic pre-processing" by the user but an important amount of data. But both allow for an unsupervised process [Caraballo 99; Cimiano *et al.* 03]. Here, as we consider a small corpus (4k words), we have uniquely employed pattern matching. More precisely, we are using pattern matching methods on syntactic contexts in order to also extract previously unexpected relations.

The initial *corpus* is formed of descriptions of Stonehenge collected from websites. But as some texts were providing historical information about the building of Stonehenge over the centuries and the modifications that took place in the arrangement of stones over the years, some of them disappearing or being moved, we had to adapt the existing descriptions to keep only information relating to what Stonehenge looked like at a certain moment of its history. Therefore, we had to discard some sentences and the Stonehenge we aim to represent is the arrangement referred to by historians as the fourth age of Stonehenge. At the same time, it appeared that the material available on the Internet was not complete enough and was lacking practical information concerning the disposition of the stones, as that information was provided by pictures displayed on the websites. Therefore, we have completed the corpus with literal descriptions based on those sketches. Those descriptions represent less than a tenth of the final corpus.

As mentioned above, the shallow parser detects the subject-verb-object structures, which gives us the possibility to focus in a first step on the term-verb-term relations with the terms appearing as the head of the subject and object phrases. This type of structure features a functional relation between the verb and the term appearing in object position. We call this functional relation a lexon. The pattern we are using

[3] See `http://ilk.kub.nl` for a demo version.

is checking also the existence of a prepositional structure following the direct object noun phrase (NP):

NP_Subj-V-NP_Obj[-Prep-NP]

with Prep for preposition, V for verb, Subj for subject and Obj for object.

As intransitive constructions contain also a lot of functional information, we rely on the chunks output to extract the verbal structures involving intransitive verbal structures:

NP-V-Prep-NP[-Prep-NP]

By using both those patterns, we retrieve a set of functional relations that we have now to sort out, as some of them contain mistakes due to errors at the syntactic level or during the relation extraction process.

Therefore, in a next step, we create a new set of relations, using this time information contained in prepositional structures. Those patterns are more frequent in the corpus and easier to detect. The prepositional patterns are defined using word frequencies and a predefined prepositions set focusing on spatial relations determining dimensions, distances and positions: in, of, within, inside, into, around... The pattern used with respect to the chosen preposition is:

NP-Prep-NP[-Prep-NP]

Some prepositional structures are then selected using a frequency measure. Only the structures including one of the N most frequent nouns are kept. In order to perform this ranking, the NPs are lemmatized.

The last selection process consists in using the prepositional structures selected to filter out incorrect functional relations from the previous set. We select the lexons that contain at least one NP appearing in a prepositional pattern.

Eventually, a last filtering operation consists in comparing the strings two by two and always keep the longest one, therefore the most complete. For example, the sentence: "The bluestones are arranged into a horseshoe shape inside the trilithon horseshoe." will produce two lexons (1)and (2).

(1) bluestones arranged into horseshoe shape

(2) bluestones arranged into horseshoe shape inside trilithon horseshoe

The lexon (1) will be discarded and the lexon (2) will be kept.

4 Results

Our extraction process results in different kind of relations.

We retrieve a small amount of relations that refer to world knowledge:

- bottom of stone
- shape of stone
- block of sandstone
- centre of circle

But our main interest lies in more specific spatial relations and more generally information related to the disposition of the stones, the shapes of the different arrangements of stones, as well as the positions of the different stone arrangements, one in respect to the other. At the same time, some more general relations like 1. and 2., can allow us to check or confirm more precise ones, or just acknowledged the existence of an element of the monument.

1. ring of bluestones
2. central part of monument
3. monument comprises of several concentric stone arrangement
4. Bluestone circle outside Trilithon horseshoe
5. Bluestone circle inside Sarsen Circle
6. Slaughter Stone is made of sarsen
7. 100 foot diameter circle of 30 sarsen stone

We give below some examples of bad relations we are extracting. Those relations are either incomplete, irrelevant or unfocused:

- Altar Stone is in front
- 120 foot from ring
- rectangle to midsummer sunrise line of monument

Those erroneous relations are due to long or complex sentences on which mistakes happen. Those errors can take place during the syntactic analysis, because of the bad tagging of an word that is unknown to the parser (such as "trilithon"

for example) and will lead to a wrong analysis of the whole chunk. They can also take place during the pattern matching, if the syntax of the sentence is unusual and does not fit the definition of the pattern we are using.

We extract also a lot of correct relations that we did not use as they were not relevant for the purpose of representing Stonehenge graphically:

- Aubrey Holes vary from 2 to 4 foot in depth
- Stonehenge stands on open downland of Salisbury Plain
- bluestone came from Preselus Mountain in southwestern Wale
- carving on twelve stone

Those relations provide us with information concerning the provenance of the stones, the location of Stonehenge itself, the sizes and weights of the stones, as well as some information describing carvings on stones.

But we need to know how many of those relations are really useful, and how well we can perform in building a graphical representation of Stonehenge, only using those relations.

5 Evaluation

The evaluation of the relations has been performed manually. It aims at comparing the amount and accuracy of the information contained in the corpus with the quality of the information contained in the relations extracted automatically. This evaluation has been carried out by two people that are not specialists of Stonehenge. Their prior knowledge consisted in being able to define Stonehenge as an ancient set of stones in England. They were unable to draw it without extra information.

One of them was only using the corpus while the other one was only referring to the relations. They have been asked to draw Stonehenge as if it was seen from above, taking into account the relative positions of the stones, the patterns, the distances, the orientation, the nature of the stones. They were told to ignore any information concerning the particular shape of a stone, details concerning the appearance of a stone (such as carvings).

In case of incomplete information, they were asked to be as precise as possible. If an object was only mentioned, without its position being precisely defined, they were asked to represent it by choosing the spot randomly. Please note that, although they were using the same drawing application, the evaluators were free to choose the objects and colours to represent the stones and other elements composing Stonehenge, which explains some differences in the appearance of the drawings.

The comparison of the drawings shows that the set of relations is complete enough and contains enough information concerning the disposition of the stones to draw a clearly recognisable Stonehenge. The relations are precise enough to determine the general structure and most of the stone arrangements with correct relative positions. The orientation of the monument, with its opening to the North East, is present also.

We are principally missing information about distances between the different structures, especially the different circles. The information concerning the avenue and the lintels is lacking also from the set of relations

Globally, the relations we are extracting automatically allow the initiation of a graphical representation of Stonehenge, but they need enrichment for a complete and accurate drawing of the monument. We will discuss in the next section the nature of the missing information and the possible reasons of this absence.

6 Discussion

Our automatic extraction process allows us to retrieve informations related to the positions of stones, their amounts, sizes, weights, as well as their composition or their shape.

The semantic information can often be double checked. This is due to the fact that the corpus is composed of the content of various documents appended. They provide different descriptions of Stonehenge containing of course some common information. In some cases, a similar information can also be retrieved in a same sentence with two different patterns.

We mainly miss information concerning distances, and elements such as the avenue or the rings of pits...

The principal reason for this phenomenon is

the length of some sentences resulting in a wrong syntactic analysis or in a partial detection of the pattern, information appearing not in all description, therefore not rated high enough and considered as not relevant by the system.

One solution would be to use more patterns, and to define them more precisely, but we want to keep on working in an unsupervised way and rely on the amount of relations extracted and their quality.

An important amount of world knowledge is lacking, as this information is not provided in descriptions aimed at readers who know what a stone might look like and that it will tend to lie on the ground...

7 Conclusion and future work

In respect to the representation of Stonehenge and the manual evaluation which are described above, we would have needed more material, and especially more accurate textual descriptions of the site. This would certainly improve the quality of the relations and increase the amount of information they convey.

All necessary information is not (always) available on the Internet. Visual representations require precise and exhaustive descriptions that are not aimed at human beings. For this reason, collecting enough relevant material proved not to be an easy task in that particular domain.

On the other hand, a task-based evaluation with an automatic system building virtual environments would be a far better way to evaluate our relations, as the human bias is difficult to control. But an automatic drawing of the site would require also the inclusion of all information related to world knowledge in the system, or a human intervention as an intermediary step between the extraction of the relations and the virtual environment building system.

We are also planning to try to improve the syntactic analysis by training the shallow parser on a corpus containing information such as semantic relations. We are expecting a better output, especially concerning the detection of subjects and objects, as well as the possibility to detect more specific structures such as locations or temporal complements.

Acknowledgements This research has been carried out during the OntoBasis project (IWT GBOU 2001 #10069), sponsored by the IWT Vlaanderen (Institution for the Promotion of Innovation by Science and Technology in Flanders).

References

(Berland & Charniak 99) Matthew Berland and Eugene Charniak. Finding parts in very large corpora. In *Proceedings ACL-99*, 1999.

(Buchholz 02) Sabine Buchholz. Memory-based grammatical relation finding. In *Proceedings of the Joint SIGDAT Conference EMNLP/VLC*, 2002.

(Buchholz *et al.* 99) Sabine Buchholz, Jorn Veenstra, and Walter Daelemans. Cascaded grammatical relation assignment. In *Proceedings of EMNLP/VLC-99*, pages 239–246, 1999.

(Caraballo 99) Sharon A. Caraballo. Automatic construction of a hypernym-labeled noun hierarchy from text. In *Proceedings ACL-99*, 1999.

(Cimiano *et al.* 03) P. Cimiano, S.Staab, and J.Tane. Automatic acquisition of taxonomies from text: Fca meets nlp. In *Proceedings ATEM03*, 2003.

(Daelemans *et al.* 99) Walter Daelemans, Sabine Buchholz, and Jorn Veenstra. Memory-based shallow parsing. In *Proceedings of CoNLL-99*, pages 53–60, 1999.

(Gamallo *et al.* 01) Pablo Gamallo, Alexandre Agustini, and Gabriel P. Lopes. Selection restrictions acquisition from corpora. In *Proceedings EPIA-01*. Springer-Verlag, 2001.

(Gamallo *et al.* 02) Pablo Gamallo, Alexandre Agustini, and Gabriel P. Lopes. Using co-composition for acquiring syntactic and semantic subcategorisation. In *Proceedings of the Workshop SIGLEX-02 (ACL-02)*, 2002.

(Lin 98) Dekang Lin. Automatic retrieval and clustering of similar words. In *Proceedings of COLING-ACL-98*, 1998.

(Pantel & Lin 02) Patrick Pantel and Dekang Lin. Discovering word senses from text. In *Proceedings of ACM SIGKDD-02*, 2002.

(Pustejovsky 95) James Pustejovsky. *The Generative Lexicon*. MIT Press, 1995.

(Reinberger & Spyns 05) Marie-Laure Reinberger and Peter Spyns. Unsupervised text mining for the learning of dogma-inspired ontologies. In Paul Buitelaar, Philipp Cimiano, and Bernardo Magnini, editors, *Ontology Learning from Text: Methods, Applications and Evaluation., Advances in Artificial Intelligence,*. IOS Press, 2005.

(Reinberger *et al.* 03) Marie-Laure Reinberger, Peter Spyns, Walter Daelemans, and Robert Meersman. Mining for lexons: applying unsupervised learning methods to create ontology bases. In *Proceedings ODBASE03*, pages 803–819. Springer-Verlag, 2003.

(Reinberger *et al.* 04) Marie-Laure Reinberger, Peter Spyns, A.Johannes Pretorius, and Walter Daelemans. Automatic initiation of an ontology. In *Proceedings ODBASE04*, pages 600–617. Springer-Verlag, 2004.

(Spyns *et al.* 04) Peter Spyns, A.Johannes Pretorius, and Marie-Laure Reinberger. Evaluating dogma-lexons generated automatically from a text corpus. In *Proceedings EKAW2004 - Worshop on the Application of Language and Semantic Technologies to support Knowledge Management Processes*, 2004.

(Volz *et al.* 04) R. Volz, S. Handschuh, S. Staab, L. Stojanovic, and N. Stojanovic. *Unveiling the hidden bride: deep annotation for mapping and migrating legacy data to the semantic web*. 2004.

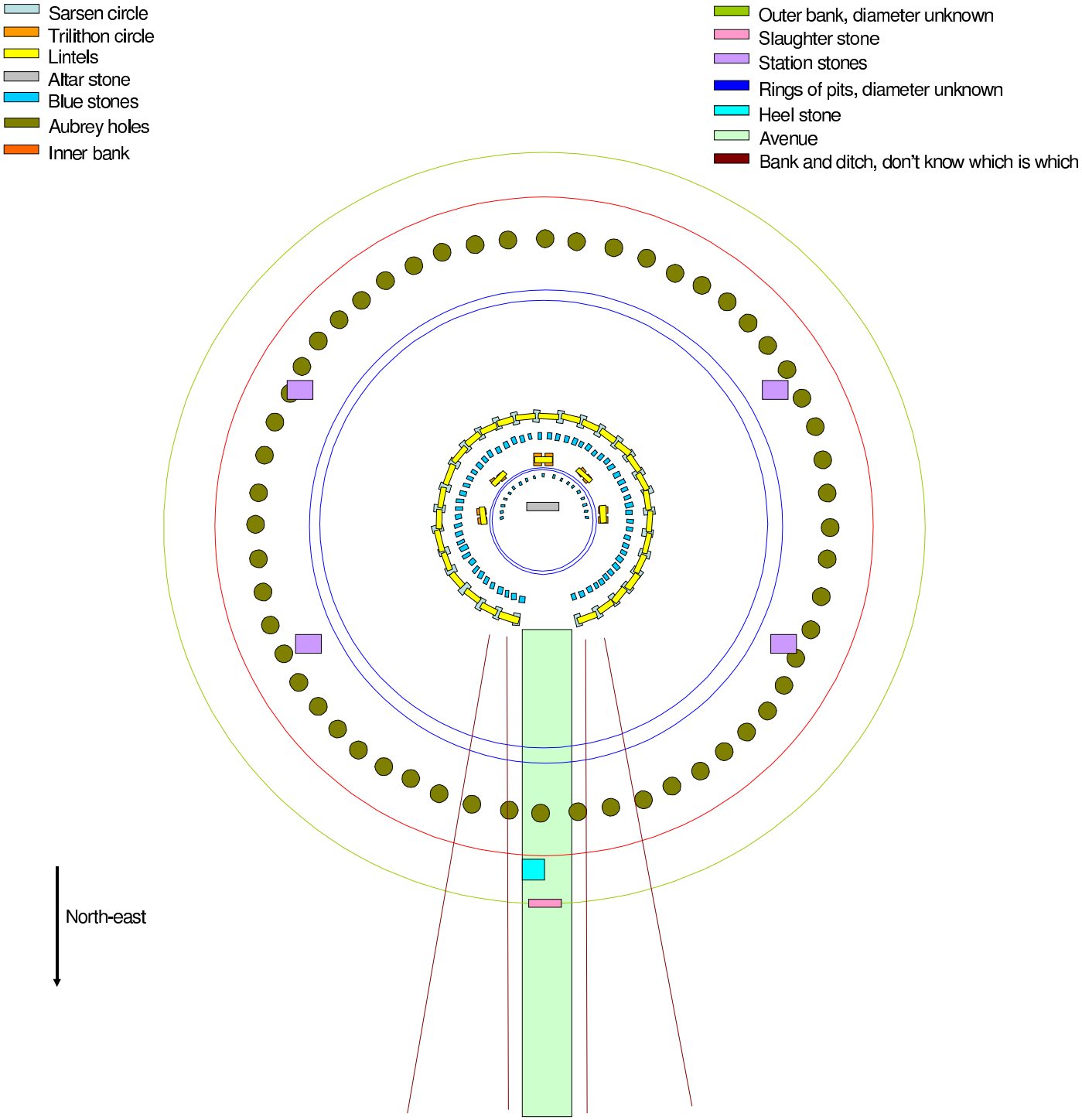


Figure 1: *Evaluation of the corpus: The main stone arrangements including the bluestones and the trilithon horseshoe as well as the bluestones and the Sarsen circles are found in the middle of the structure, in yellow and blue on the picture. Far around, we find the Aubrey holes, Station stones and banks. The Avenue as well as the rings of pits are also represented.*

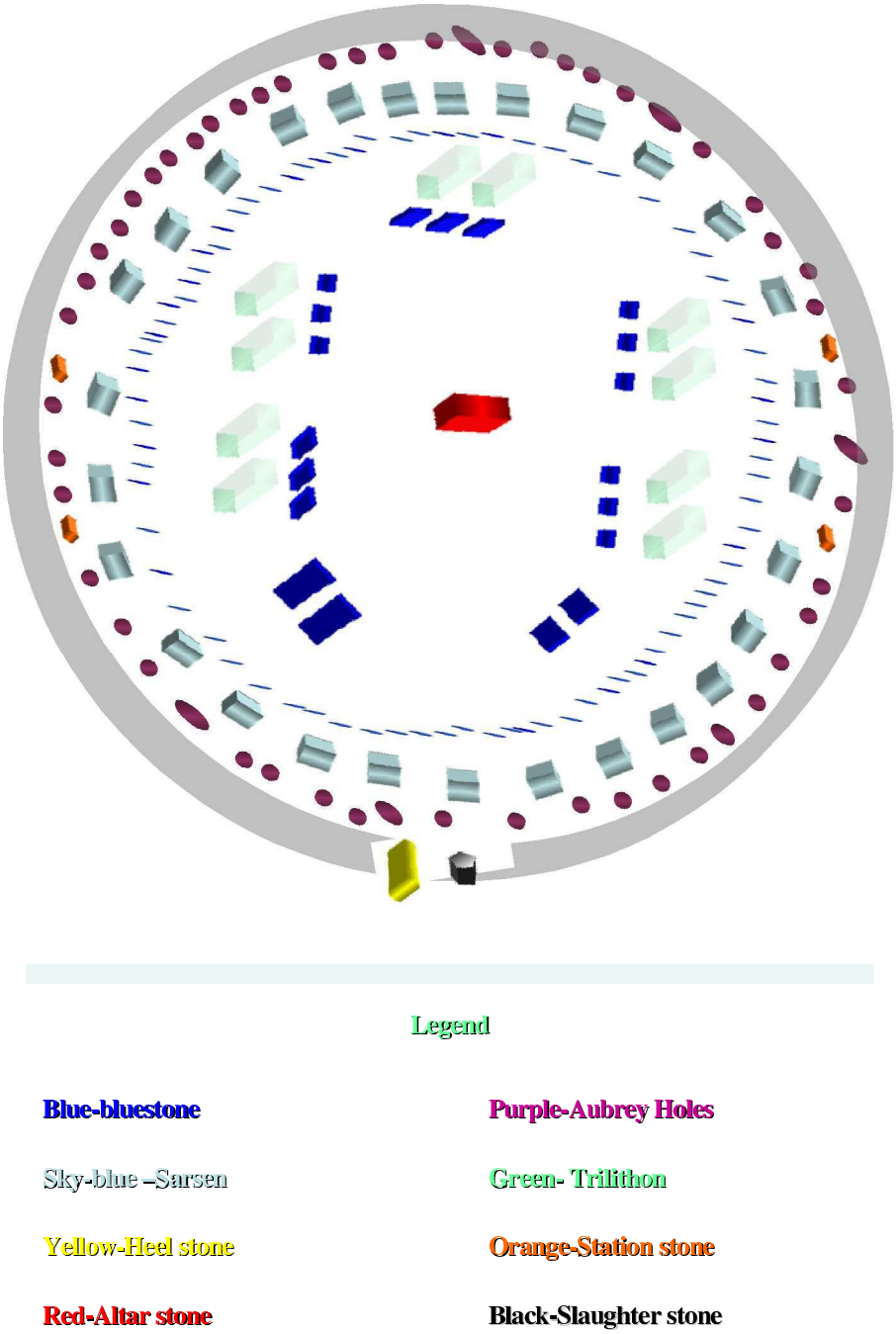


Figure 2: *Evaluation of the relations: The proportions between the central stone arrangements (horse-shoes and circles of stones) are missing. The outer bank (in grey), the Aubrey holes and Station stones are too close to the main stone arrangements. The Avenue and the rings of pits are missing, as well as the lintels on the trilithons and Sarsen stones. The orientation is correct, with the heel stone and the slaughter stone placed at the entrance of the monument, on the North-East side.*

OLE — A New Ontology Learning Platform

Vít Nováček[1] and **Pavel Smrž**[2]
[1]Faculty of Informatics, Masaryk University
Botanická 68a, 602 00 Brno, Czech Republic
E-mail: xnovacek@fi.muni.cz
[2]Faculty of Information Technology, Brno University of Technology
Božetěchova 2, 612 66 Brno, Czech Republic
E-mail: smrz@fit.vutbr.cz

Abstract

Ontologies are commonly considered as one of the essential parts of the Semantic Web, providing a theoretical basis and implementation framework for conceptual integration and information sharing among various domains. In this paper, the main principles and current highlights of the OLE project development are presented and discussed. OLE stands for Ontology LEarning. The purpose of the project is to develop a system for bottom-up automatic generation and merging of domain specific ontologies, representing particular domains of human scientific knowledge. In order to gain ontology concepts from textual resources, various methods can be used. Presently, the method of automatic acquisition of semantic relations is examined. The method and its implementation details are described here, provided with brief comparative overview of other methods used for the development of domain specific ontologies. When merging automatically created ontologies, a need of uncertain information representation arises. The main ideas of representing uncertainty in ontologies and related remarks for future work on the OLE project are mentioned in this paper as well.

1 Introduction

In the field of computer science, an ontology is understood as a formal and machine readable representation of a concept set, stratified in classes and including some relations among particular concepts and their classes. Ontologies are able to provide a comprehensive representation of information related to a particular subdomain of human knowledge. Such a representation can be utilised for an efficient semantic querying upon the subdomain objects, resource relevance measurement, interoperability of different systems and many other tasks. Ontologies also play the major role in the Semantic Web vision.

The basic approach to the ontology building is the manual definition of domain conceptualisation. This task is usually performed by a group of domain experts. Various elaborated tools support the work; the most popular ones are Protégé, WebODE and OntoEdit. A comprehensive survey of such ontology engineering frameworks can be found in (Arpirez *et al.* 03).

The manual creation of ontologies presents a tedious work, is error-prone and the results are often too subjective. Moreover, it is infeasible to organise a group of experts for each possible domain. This led to the idea of automatic extraction of ontologies from available resources.

A method that can be used for ontology acquisition from texts was sketched by M. A. Hearst in (Hearst 92). It is based on the automatic pattern-based extraction of particular semantic relations. The hyponymy or *is-a* relation serves as a basis for the natural sub-concept/super-concept hierarchy of ontology classes. The notion of the automatic extraction of hyponymical constructs from textual data can be adopted for any other semantic relation, although the applied techniques may differ. Methods based on token co-occurrence can be employed to gather sets of concepts belonging to the same class. Various modifications of these two generic techniques are presented in (P. Pantel 04).

The OLE platform introduced in this paper takes advantage of the automatic acquisition methods. It enables creating the core taxonomy of an ontology subdomain in the bottom-up manner, from ontologies with a very simple structure to more complex ones, in a continual iterative process. It is also able to extend, refine and update ontologies with respect to new data.

A miniontology for each input resource is created first. It consists of concepts and classes gained from the given resource. The miniontologies are integrated into the current ontology on the fly. The process of ontology merging and alignment embodies the application of uncertainty representation methods. The emerging BayesOWL framework (Y. Peng 05) — a probabilistic extension of OWL — provides tools for this task.

OLE differs from other ontology-learning systems also in its accent on modularity and flexibility. Virtually any method of automated knowledge acquisition can be employed as an independent part of the OLITE module. Section 3 gives details describing such an integration. Presently, the method of pattern-based extraction of semantic relations along with dynamic pattern learning is examined.

The rest of the paper is organised as follows. The next section presents a brief overview of the OLE architecture. Section 3 discusses one of the essential parts of OLE — the OLITE module. The efficiency of the platform is demonstrated by the results given in Section 4. Sections 5 and 6 compare OLE with other available systems and indicate the future directions of our research.

2 OLE Architecture

2.1 Design Considerations

The design of OLE has been influenced by the need for autonomy, efficiency and precision of the resulting platform. The following list summarises the major requirements:

- The tool should support the user-friendly interactive way of ontology acquisition, but also the fully automatic process of knowledge mining that can run without any human assistance.
- The efficiency of ontology acquisition is crucial, for the system will process gigabytes of data.
- The precision is preferred over the recall. Even if the number of the extracted conceptual structures will be relatively low (compared to the number of relations a human can identify in the same resource), it will be balanced by the extensive quantity of resources available.
- The relations between concepts stored in the resulting ontology do not need to be precise — the explicit uncertain knowledge representation is one of the essential parts of OLE. The loss of exactness is balanced by the increased fuzzy precision of the whole process.

2.2 System Components

The modular architecture of OLE is given in Figure 1.

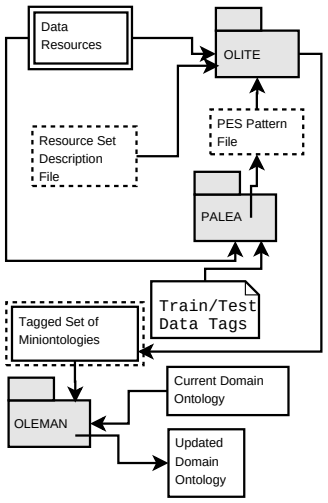


Figure 1: The modular architecture of the OLE platform

The **OLITE** module processes plain text and creates the miniontologies from the extracted data. The following sources are related to the OLITE module:

- *Data Resources* – documents provided by external tools (document classifiers, existing databases of related resources etc.).
- *Resource Set Description File* – a XML (RDF) encoded annotation of the resource files; it is read by the OLITE module in order to supply the extracted concept sets with category affiliation and other information.
- *PES Pattern File* – the definition of the semantic relation patterns.
- *Tagged Set of Miniontologies* – the output of the OLITE module in the form of miniontologies which correspond to the respective documents.

The **PALEA** module is responsible for learning of new semantic relation patterns. A simple method of frequency analysis is integrated in the current implementation.

The **OLEMAN** merges the miniontologies resulting from the OLITE module and updates the base domain ontology. The uncertain information representation techniques are employed in the phase of ontology merging[1]. The module can

[1]Automatic matching of ontologies is a complex task which is not tackled in this paper. Relevant information on ontology merging and alignment can be found in (Doan

be used as a rudimentary ontology manager as well.

2.3 Implementation Remarks

All the OLE software components are implemented in the Python programming language. A special attention has been paid to the object oriented design. Another reason for choosing Python was the wide range of freely available relevant modules and application interfaces[2].

Python as an interpreted language can be inefficient for the implementation of some part of the OLE platform. Special tools improving the computational efficiency of the Python code are available. For example, we are going to take advantage of Psyco (Psy05) which is similar to the Java just-in-time compiler.

3 OLITE Module

3.1 Text Preprocessing

OLITE processes English plain-text documents and produces the respective miniontologies. To increase the efficiency, the input is preprocessed with the aim to reduce irrelevant data. Shallow syntactic structures that appear in the semantic pattern are also identified in this step.

The preprocessing consists of *splitting of the text into sentences*, *eliminating irrelevant sentences*, *text tokenization*, *POS tagging and lemmatization*, and *chunking*. The first two steps are based on regular expressions and performed in one pass through the input file. The possible relevance — the presence of a pattern — is detected by matching "core words" of the patterns.

The next phases of preprocessing depend on utilisation of NLTK natural language toolkit (NLT05) with custom-trained Brill POS tagging algorithm (Brill 94) and regular expression chunking incorporated. Moreover, the usage of NLTK toolkit (which allows users to train their own POS taggers from annotated data and easily create efficient chunking rules) enables to adapt the whole OLE system even for other languages than English in future.

Fast regular expression-based chunking is then performed on the tagged sentences. The resulting file is stored in the form of a simple annotated vertical text — sentence elements with individual lines in the form of token/tag/lemma triples separated by the tab character. Tags conform to the Brown corpus (see (Bro05) for details).

3.2 Concept Extraction and Miniontology Generation

Any extraction algorithm can be integrated into OLITE in the form of a plug-in. Such a plug-in is responsible for the concept extraction, precise (or fuzzy) assignment of a class or a property and passing of gained information further in order to build an output miniontology.

The currently implemented pattern-driven concept extraction process accepts patterns in a special form. The designed universal pattern-specification format allows new patterns to be easily added in the future.

The patterns are loaded and compiled from a separate PES file. PES stands for *Pattern Extended Specifications*. The syntax is similar to extended regular expressions, a few new symbols with higher level semantics are added. A chunk in a pattern is defined by a special expression — one of the NX, VX, AX, or UC character groups, representing a noun, verb, adjectival chunks or an unchunked text respectively. The expression can be amended by the '+' sign, indicating a sequence of same chunks. Chunk representation is enclosed in `~ ~`. The core words are enclosed in `% %`. All other elements of the extended regular expressions syntax are accepted by the internal PES compiler.

The *is-a* pattern in the form of:

```
NP1 {‘‘,’’} ‘‘such as’’ NPList2
```

is transformed into the PES expression:

```
~NX~,? %such as% ~NX+~
```

The concepts are extracted from the chunked text, using the abstract regular expression matching represented by internal objects. The abstract regular expression matching means that the expressions involved are not compared as pure strings. Rather, they are taken as abstract objects bearing raw information on what are they representing (a chunked sentence or a pattern) and regular expression based operations to be launched mutually upon their content. However, any other extraction method can be easily incorporated as an independent plug-in.

When the concepts are extracted, they are implicitly bound by the respective semantic relation.

et al. 03) and in (Ehrig & Staab 04). An introduction into the uncertain information in ontologies is given in (Y. Peng 05).

[2]For example, the NLTK natural language toolkit (NLT05) is used

The importance of direct mapping between the concept hierarchies and the corresponding semantic relations is obvious. If we can discover the relations in the resource data, then we can create the respective taxonomy of the resulting ontology directly in the bottom-up manner (from ontologies with very simple structure to more complex ones in a continual iterative process). And when such a process is fully automated, we can acquire and update our ontologies dynamically from the submitted real world data. The ontologies reflecting the particular resources are merged into the current domain ontology by another specialised software agent. The result is no way arbitrary and subject-dependent. In reverse, it is purely empiric and as up to date as the resources used, when considering the current state of the domain.

The extracted information is stored in a universal internal format, no matter which extraction technique has been used. The format is extensively expressive and universal with respect to efficient encoding of various relations and uncertainty representation[3]. In order to facilitate the merging process, the current domain ontology is processed in this format as well and the alignment is executed on these representations.

The resulting updated ontology can be flushed by applying respective translation rules. These rules are implemented as an independent plug-in (likewise the extraction algorithms) responsible for producing the output file in a desired format. Currently, the OWL DL format is supported only, but OLITE is able to produce any other format by the same mechanism.

4 Results

The method of pattern-based acquisition of simple relations was tested on general corpus texts containing about 10^8 words. The selected patterns are presented in the intuitive regular expression-like form in the first column of the table below[4].

The H_{abs} column contains numbers of matching sentences. Relative frequency of matches is given in the H_{rel} column[5]. The F_{all} field contains a ratio of successful pattern hits among randomly chosen sample of 50 matching sentences. Eventually, the F_{acq} column offers a ratio of conceptual structures acquirable by the OLITE module from the matching sentences.

Relatively high frequency of currently recognised semantic structures (compared to relations identified by a human) is very promising for further development. However, implementation of another techniques is essential in order to gain more general relations and even properties. Also, uniqueness of gained concepts must be examined properly across particular domains, because when choosing multidisciplinary random matches from the corpus, the measure is not very evidential.

5 Related Work

One of the best-known ontology-acquisition efforts is represented by the OntoLearn project (Gangemi *et al.* 03). The statistical methods based on frequency measures are equipped in terminology extraction from the source data in OntoLearn as well as in OLE. However, the systems considerably differ in their use of the "template" ontology. The WordNet database is queried in several stages of the semantic interpretation and specific relation discovery in OntoLearn. New relation patterns are inferred based on the known WordNet conceptual relations. Therefore, the results of OntoLearn are determined by the coverage of WordNet. On the other hand, the process of ontology acquisition can start from scratch in OLE and the current "template" ontology can be dynamically extended and refined.

The KnowItAll (Etzioni *et al.* 04) system incorporates the same extraction of semantic relations as is implemented in OLE. The uncertainty is introduced in the form of so called web-scale probability assessment in KnowItAll, but not as a part of the ontology structure itself. OLE represents the whole conceptual structure of a given domain in the unified system integrating the uncertain information.

The OLITE and PALEA modules implement just basic methods of pattern learning and their application. Advanced algorithms for pattern-based extraction of semantic relations are described in (Etzioni *et al.* 04), (Hearst 92) and (P. Pantel 04). The concept clustering techniques for terascale knowledge acquisition are introduced in (P. Pantel 04), (T. T. Quan 04) presents the

[3]The research behind proposal and implementation of this format will be presented in another paper.

[4]The patterns are partially adopted from (Etzioni *et al.* 04) and (Hearst 92).

[5]The overlap among the matches was found to be insignificant.

Selected *isa* patterns	H_{abs}	H_{rel}	F_{all}	F_{acq}
NP (and|or) other NP	17384	0.28	94	85
NP including (NPList (and|or))? NP	23985	0.38	92	73
NP (is|was) a NP	140632	2.26	66	30
(NPList)? NP like NP	147872	2.37	16	14
sums (H fields) and averages (F fields)	329873	5.29	52.00	50.50

Table 1: The most productive extraction patterns

fuzzy concept clustering. All these techniques can be adopted by the OLITE module to supplement the dynamic pattern learning and application.

6 Conclusions and Future Directions

OLE is primarily intended for autonomous creation and management of domain specific ontologies. The bottom-up approach to the ontology acquisition is emphasised, as well as the need for uncertainty representation. The preliminary results clearly show that OLE provides the modular and flexible platform for comparing and testing various information extraction techniques. The OLITE component implements the basic knowledge acquisition methods, other modules can be easily added.

Challenging work remains to be done on the PALEA module, especially in the area of dynamic acquisition of new patterns for additional semantic relations. Many advanced techniques for concept mining still wait for their implementation. One of them is FFCA — the Fuzzy Formal Concept Analysis (T. T. Quan 04) which is based on fuzzy concept clustering. The notion of uncertainty is implicitly embraced already at the initial level of information extraction. The ontology merging process in OLE will benefit from this approach. It will be implemented as another information extraction plug-in.

7 Acknowledgements

This work is partially supported by Academy of Sciences of Czech Republic, 'Information Society' program, the research grant T100300419.

References

(Arpirez *et al.* 03) J. C. Arpirez, O. Corcho, M. Fernandez-Lopez, and A. Gomez-Perez. Webode in a nutshell. *AI Magazine*, 24(3):37–47, 2003.

(Brill 94) E. Brill. A report of recent progress in transformation-based error-driven learning. In *Proc. ARPA Human Language Technology Workshop '94*, Princeton, NJ, 1994.

(Bro05) *The Brown Corpus Tag-set*, 2005. Available at: http://www.scs.leeds.ac.uk/amalgam/tagsets/brown.html.

(Doan *et al.* 03) A. Doan, J. Madhavan, R. Dhamankar, P. Domingos, and A. Halevy. Learning to match ontologies on the semantic web. *The VLDB Journal*, 12:303–319, 2003.

(Ehrig & Staab 04) M. Ehrig and S. Staab. Qom — quick ontology mapping. In *Proceedings of the International Semantic Web Conference (ISWC)*, pages 683–697. Springer-Verlag, Berlin, Heidelberg, 2004.

(Etzioni *et al.* 04) O. Etzioni, S. Kok, S. Soderland, M. Cafarella, A.-M. Popescu, D. S. Weld, D. Downey, T. Shaked, and A. Yates. Web-scale information extraction in knowitall (preliminary results). In *Proceedings of the 13th International Conference on World Wide Web*. Springer-Verlag, Berlin, Heidelberg, 2004.

(Gangemi *et al.* 03) A. Gangemi, R. Navigli, and P. Velardi. Corpus driven ontology learning: a method and its application to automated terminology translation. *IEEE Intelligent Systems*, pages 22–31, 2003.

(Hearst 92) M. A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics*, pages 539–545, Morristown, NJ, USA, 1992. Association for Computational Linguistics.

(NLT05) *NLTK: Natural Language Toolkit – Technical Reports*, 2005. Available at: http://nltk.sourceforge.net/tech/index.html.

(P. Pantel 04) E. Hovy P. Pantel, D. Ravichandran. Towards terascale knowledge acquisition. In *Proceedings of Conference on Computational Linguistics (COLING-04)*, pages 771–777, 2004.

(Psy05) *The Ultimate Psyco Guide*, 2005. Available at: http://psyco.sourceforge.net/psycoguide/index.html.

(T. T. Quan 04) T. H. Cao T. T. Quan, S. C. Hui. Automatic generation of ontology for scholarly semantic web. In *ISWC 2004: Third International Semantic Web Conference. Proceedings*, pages 726–740. Springer-Verlag Berlin Heidelberg, 2004.

(Y. Peng 05) R. Pan Y. Peng, Z. Ding. Bayesowl: A probabilistic framework for uncertainty in semantic web. In *Proceedings of Nineteenth International Joint Conference on Artificial Intelligence (IJCAI05)*, 2005.

Mining Personal Data Collections to Discover Categories and Category Labels

Shih-Wen Ke*, Michael P. Oakes, Chris Bowerman
School of Computing and Technology
University of Sunderland
David Goldman Informatics Centre,
St. Peter's Campus, Sunderland, SR6 0DD, England

{*george.ke, michael.oakes, chris.bowerman}@sunderland.ac.uk

Abstract

This paper describes the text mining of personal document collections in order to learn the categories of the documents in the collection, and to assign a suitable text label to each category. In the first experiment we make use of a pre-classified collection of documents from which we extract a text label for each category. In the second experiment we use the *k*-means clustering algorithm to automatically learn which categories are present in a collection, then generate short text labels to describe each category. The technique, which is essentially a form of automatic indexing, can be used whenever class exemplars are generated as part of the document clustering process.

1 Background

Whilst much research has been undertaken in the area of mining large, public data collections organised by trained information scientists, typified by the MUC and TREC conferences, the increasing use of digital technology has led to growing personal data collections which are poorly organised and continually changing in content. Typical collections of this sort are collections of images from digital cameras, videos, voicemail and music files as well as bodies of emails. Electronic text documents, such as emails, project notes, favourite articles, are now widely used by office workers, administrators and everyone. The amount of these kinds of documents used and processed is rapidly increasing, which leads to difficulties in managing them. These personal collections of documents often have categories of various sizes, are likely to be poorly structured and individualised and will therefore need to be categorised in a unique manner for each user in order to be able to be searched meaningfully by them (Whittaker & Hirschberg 01). A number of authors have described methods of automatically labelling document clusters. Such methods are necessary for personal document collections where the clusters must be constantly updated.

The LabelSOM approach (Rauber & Merkl 99) automatically labels every node of a trained Self-Organising Map (SOM). The SOM differs from the clustering structures described in this paper in that the resulting clusters are arranged in a lattice, where neighbouring clusters are similar to each other. For each cluster, LabelSOM finds the square of the difference between the value of every feature (word) of every input document and the weight vector of the cluster. The sums of these squared differences for each word are stored in a "quantisation vector". To create a set of n labels for each cluster, LabelSOM takes the n smallest vector elements from the quantisation vector. Words which appear less than a threshold number of times in the documents making up a cluster are disregarded. Maarek et al. (Maarek et al. 00) investigate ephemeral clustering, the situation present in personal document collections which are constantly being added to so that the output clustering structure is constantly being updated. They annotate the generated clusters using "lexical affinities" as indexing units. A lexical affinity consists of a pair of words which frequently co-occur together, for example within a sliding window of plus or minus 5 words. Lexical affinity can be estimated by a number of measures including mutual information, and a cluster can be characterised by the highest scoring word pairs found within the documents of that cluster. An advantage of this method is that the lexical affinities provide de facto word sense disambiguation. Chen and Liu (Chen & Liu 04) although not working specifically within the field of

Information Retrieval, describe the ClusterMap method of automatically labelling clusters. The cluster descriptors are the parameters used to map the k-dimensional space occupied by the cluster (where k is the number of attributes characterising each input pattern) onto a two-dimensional representation, and the characteristics of the two-dimensional map itself.

Although the literature presents automatic cluster labelling, few authors focus on smaller datasets such as personal document collections, which make the job more difficult as there are much fewer resources from which information can be extracted. Popescul and Ungar (Popescul & Ungar 00) state that the most commonly used method of automatic cluster annotation, labelling with the most frequent words in the clusters, "ends up using many words that are virtually void of descriptive power even after traditional stop words are removed". In this paper, we overcame this problem by creating our own collection specific vocabulary list, from which all cluster descriptors were chosen. Using the tf·idf method described in section 2, we ensured that descriptors could not be chosen if they were common to large numbers of documents. As for the work described in this paper, each cluster can be represented by a class exemplar, and we need to know which features are characteristic for a particular cluster.

The users are not typically information retrieval experts, so we wish to proceed without the need for user interaction with the system. In addition the size of the collection changes over time as new categories will be created or existing ones will be deleted and so an efficient means of automatically organising this data according to its content is needed. In this paper we consider one such technique in which we are able to cluster documents automatically and then assign to each cluster a meaningful text label.

2 Approach

The simplest method of deciding which word or words best characterise a document or category of documents as a whole is based on the principle, widely used in information retrieval, that mid-frequency terms have greatest information content. The highest frequency words are function words, ("the", "and", "or" etc.) which can be filtered out by the use of a list of stop words. A more refined method is the tf·idf (term-frequency, inverse document frequency), which takes into account not only how frequently a word appears in a text, but in how many texts altogether it is found. For example, a word like "the" will occur very frequently in the "earn" texts, but since it also occurs frequently in all the other categories, is not particularly indicative of the "earn" category. However, a word like "bundesbank" occurs quite frequently in the money-fx category, but relatively rarely in the other categories, so it helps form a good descriptor for the "money" category. Although many feature selection methods have been devised, tf·idf has been proven extraordinary robust and difficult to beat (Robertson 04). We used tf·idf to carry out the feature selection process, where the words with the greatest tf·idf scores were retained as attributes to describe the documents. The formula for tf·idf is as follows:

For a term i in document j:

$$W_{ij} = tf_{ij} \times \log\left(\frac{N}{df_i}\right), \text{ where}$$

W_{ij} = weight of i in j
tf_{ij} = number of occurrence of i in j
df_i = number of documents containing i
N = total number of documents

We also use VSM (Vector Space Model) where each document d_j is represented by a set of features (words) $d_j = (f_{j1}, f_{j2}, f_{j3} \ldots f_{jx})$, where x is the number of features. For each category c_k, we sum each feature value of each document and then we will obtain a final set of features V_k.

$$V_k = (\sum_{n=1}^{n} f_{kj1}, \sum_{n=1}^{n} f_{kj2}, \sum_{n=1}^{n} {}_{kj3} \cdots \sum_{n=1}^{n} f_{kjx})$$

where n is the number of document in the category.

For example, there are four documents in a category and each document has eight features. The final set of features is obtained by adding up each column which represents each feature.

Term	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**
Attr.	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
d_1	2	0	1	3	0	0	0	1
d_2	0	2	1	2	0	2	0	0
d_3	1	1	0	3	1	1	3	0
d_4	0	1	0	1	0	0	1	0
V	**3**	**4**	**2**	**9**	**1**	**3**	**4**	**1**

Table 1: Example of obtaining a final set of features

Once we have obtained V, we then look up the original terms corresponding to each attribute in V. Therefore, in this case term 'D' corresponding to the

attribute f_4 has the highest weight and will form a suitable text label for the category as a whole.

3 Use of predefined vs automatically discovered categories

In this section we describe our experience of determining category labels both for a predefined set of text categories and a set learned by the *k*-means clustering algorithm.

3.1 Predefined categories

For the predefined categories we used the ApteMod version of Reuters-21578 as our dataset, which has been a benchmark corpus in text categorisation evaluation. The ApteMod version of Reuters-21578 was obtained by removing unlabelled documents and selecting the categories that have at least one document in the training set and the test set. This process resulted in 90 categories in both training and test set (Yang & Liu 99). In our initial study we selected the five categories with the most documents and we used only the documents in the training set. The statistics of the selected categories are shown in Table 2.

Category	Number of documents
Earn	2877
Acq	1650
money-fx	538
Crude	389
Trade	369
TOTAL	**5823**

acq = Mergers/Acquisitions
earn = Earnings and Earnings Forecasts
crude = Crude Oil
money-fx = Money/Foreign Exchange
trade = Trade

Table 2: Selected categories statistics

After stop word removal, stemming and the tf·idf feature selection process which eliminated those terms with weights less than 1, we were able to represent the documents by feature sets of 2113 unique words. The aim was to find whether the learned category labels corresponded well with the pre-defined labels.

A subjective perusal of the discovered terms representing each of the Reuters predefined categories showed that although only a few single words describe their respective categories completely, using all five most frequent words in each category as a compound descriptor gives a good indication of what the documents in the category are about. The five most weighted features selected to represent each category are listed in Table 3.

In category "earn" (short for Earnings and Earnings Forecasts), less informative terms such as "vs" and "qtr" came to the top of the list. This is because in the "earn" category there are many economy and business reports where "vs" is used to compare figures and numbers and "qtr" is short for quarter.

Earn	
Term	**weight**
vs	910.7
qtr	156.5
quarterli	27.2
Payabl	25.5
acquisit	14.4

Acq	
Term	**weight**
acquisit	61.9
file	22.4
plc	19.1
affili	10.2
store	9.6

money-fx	
Term	**weight**
intervent	17.1
bundesbank	13.1
interven	12.4
Taiwan	7.7
miyazawa	7.5

Crude	
term	**weight**
crude	43.2
refineri	15.2
ecuador	15.1
drill	11
venezuela	8.6

Trade	
Term	**weight**
gatt	16.9
tariff	15.9
reagan	13.5
taiwan	11.8
chip	11.7

**Please note that the terms have been stemmed.*

Table 3: Most weighted terms for each category

The results show that the discovered descriptors for each predefined category have a certain degree of relationship to the original category labels, such as "gatt" (short for General Agreement on Tariff and Trade) for the "trade" category.

3.2 Automatically discovered categories

For our experiment on automatically discovered categories, we used the DUC 2004 (Document Understanding Conference) corpus as our dataset. The DUC corpus was designed for the evaluation of automatic text summarisation systems at the annual DUC conference, which is organised by the US National Institute of Science and Technology (NIST). DUC 2004 contains 500 documents which are organised into 50 folders with each folder containing 10 documents. We used a subset of these,

where only five documents were randomly selected from each folder in order to conduct a small-scale experiment for demonstration purposes. This resulted in 250 documents in total. After stop word removal, stemming and tf·idf feature selection a vocabulary of 5373 unique terms was obtained (terms with $W_{ij} < 10$ were eliminated). DUC 2004 can be considered as raw data as the documents are not labelled and not organised into categories, as would be the case for an evolving personal data collection – such as RSS newswire feed.

We used an unsupervised clustering algorithm, *k*-means, to cluster the total of 250 documents into a number of partitions. Clustering is fundamental in knowledge acquisition and considered as the process of partitioning a set of vectors into homogeneous groups (Tasoulis & Vrahatis 05). In our case we clustered the documents into 10 partitions.

The *k*-means algorithm is a partitional, non-hierarchical and the simplest and most commonly used data clustering method which employs a squared error criterion. The number of clusters *k* needs to be defined prior to the clustering process. This process operates in an iterative manner where an initial partition is found and the process keeps on searching for a partition of documents that has the minimum sum of squared errors within the collection (Jain et al. 99).

The procedures of the *k*-means algorithm we used are described as follows:

1. Initial centroids are found for the whole document collection by randomly choosing five document vectors.
2. Assign each document vectors to their closest centroid.
3. Re-calculate the centroid vector for each cluster using current cluster memberships.
4. Compute the quality function. If a better quality partition is produced, the process is repeated from step 2 until the optimal partitioning is reached.

At the end of this process, 10 vectors representing the centroids of 10 clusters are obtained. Now we need to find the documents within each cluster. In order to do this a similarity threshold needs to be specified beforehand. The similarity threshold determines the size of each cluster. In our case, the larger the threshold the more documents each cluster contains. Normally the similarity threshold value is determined by a cross validation process so that an optimal value will be found. In our experiment, we obtained the similarity threshold value by simply calculating the mean (denoted by ST_m) of the distances between ten centroids using Euclidean distance. We used two similarity threshold values for our automatically discovered categories – ST_m and $2ST_m$. Using similarity threshold allows users to specify how strict the class label is going to be for each cluster. Although many soft clustering methods, which allow a document to appear in multiple clusters, are available, this may cause confusion in extracting class exemplars as these documents contain terms that are not distinctive for any of clusters (Lin & Kondadadi 01). However, similarity threshold method may also have a drawback. When the documents within the specified similarity threshold are very sparse, it may result in producing less representative and informative class exemplars for this particular cluster.

4 Results

When the similarity threshold = $2ST_m$, each cluster contained a relatively large number of documents, which results in many clusters overlapping over a large area in vector space. This leads to most of the clusters being very similar to each other and the class exemplars (centroids) created for each cluster being almost identical, with much less discriminatory power to distinguish the clusters.

Cluster 1	
Term	**Weight**
Govern	3.1
Office	3.1
New	2.9
Presid	2.7
Report	2.6
# of docs.	**22**

Cluster 5	
Term	**Weight**
Umpire	2.0
Fire	1.6
Seri	1.6
Game	1.4
Leagu	1.1
# of docs.	**2**

Cluster 6	
Term	**Weight**
Govern	2.9
Parti	2.7
Ecevit	2.1
Turkey	1.4
Parliament	1.3
# of docs.	**4**

Cluster 7	
Term	**Weight**
offici	7.2
New	7.1
Govern	5.9
People	5.7
Year	5.6
# of docs.	**47**

Table 4: Category exemplars for clusters with ST_m

Better results, shown in Table 4, were obtained with DUC 2004 after the documents were partitioned into different clusters based on the similarity threshold ST_m. As seen in Table 4 only four clusters of documents were obtained and the other six clusters

were empty. This is because clusters with ST_m cover a much smaller area in vector space, so fewer documents are partitioned into clusters. Now each cluster has more distinctive exemplars, but there is a trade-off of having more distinctive exemplars and losing more documents in each cluster, which is quite significant in our case.

The mode of generating category exemplars for each of the predefined Reuters categories and unlabelled DUC 2004 documents is given above. The unsupervised clustering method, *k*-means, that we have discussed also involves the creation of category exemplars for each of the learnt categories. Once again each attribute of the exemplars corresponds to a non-stoplisted word in the overall vocabulary of the documents, so the highest weights in each cluster centroid or class exemplar can correspond to words which act as useful descriptors of the categories.

The next step is to conduct experiments on the Enron email dataset, which is a real personal documents collection distributed by William Cohen at Carnegie Mellon University. The Enron email dataset consists of 517,431 email messages that belong to 150 users of Enron Corp. These emails have many attributes (e.g. sender's information, recipient's information, subject, etc.) and are heavily time-dependent (Bekkerman et al. 04) . Business emails usually contain more formal vocabulary, whereas personal emails contain many informal conversations and slang. Before creating class exemplars for emails, we need to determine what attributes are to be considered and what feature selection metric will yield the most informative and distinctive terms. Further user studies and evaluations, such as questionnaires, interviews and observations, are needed in order to discover the variables of personal document collections and what users perceive to be of benefit in managing such collections.

5 Conclusion

Although the technique described in this paper is relatively simple, it enables meaningful category labels to be generated from the category centroids or class exemplars which are generated during the *k*-means and other clustering processing. This is particularly useful for personal data collections, to which new documents are continually being added and existing ones being removed. This results in a continual need to re-categorise. New category labels can be automatically added whenever new categories are created. Other feature selection techniques, such as information gain or the χ^2 statistic can be employed to produce more meaningful category labels (Yang & Pederson 97). The feature selection process can be carried out for each cluster after documents are clustered so that the category exemplars of each cluster are selected more independently of other clusters.

The trade-off between having more distinctive exemplars and losing more documents in each cluster must be optimised. Furthermore, there is a need for user evaluation to find out which technique produces the most subjectively meaningful category labels.

User studies are also needed to explore who keeps a personal document collection, how they behave in organising and managing the collection and what factors for such relatively small collections might help us to help users personalise their ways of managing documents.

References

(Bekkerman et al. 04) Bekkerman, R., McCallum, A. and Huang, G., *Automatic Categorization of Email into Folders: Benchmark Experiments on Enron and SRI Corpora*. CIIR Technical Report IR-418 2004. Available at: http://www.cs.umass.edu/~ronb/papers/email.pdf

(Chen & Liu 04) Chen, K. & Liu, L., *ClusterMap: Labeling Clusters in Large Datasets via Visualization*. In ACM 13th Conference on Information and Knowledge Management, 2004.

(Jain et al. 99) Jain, A. K., Murty, M. N., Flynn, P.J., *Data Clustering: A Review*, ACM Computing Surveys, vol. 31(3), pp.264-323.

(Lin & Kondadadi 01) Lin, K. & Kondadadi, R., *A Similarity-Based Soft Clustering Algorithm for Documents*. In International Conference on Database Systems for Advanced Applications, 2001.

(Maarek et al. 00) Maarek, Y.S., Fagin, R., Ben-Shaul, I.Z., and Pelleg, D., *Ephemeral Document Clustering for Web Applications*. Research Report RJ10186, IBM Almaden Res. Ctr., San Jose, CA, 2000. Available at http://www.almaden.ibm.com/cs/people/fagin/cluster.ps

(Popescul & Ungar 00) Popescul, A. & Ungar, L., *Automatic Labeling of Document Clusters*, unpublished paper, University of Pennsylvania. Available at: http://www.cis.upenn.edu/~popescul/Publications/popescul00labeling.pdf

(Rauber & Merkl 99) Rauber, A., & Merkl, D., *Automatic Labeling of Self-Organizing Maps: Making a Treasure-Map Reveal Its Secrets*. In Pacific-Asia Conference on Knowledge Discovery and Data Mining, 1999, pp. 228-237

(Robertson 04) Robertson, S., *Understanding Inverse Document Frequency: On theoretical Arguments for IDF*. Journal of Documentation, vol. 60(5), pp.503-520.

(Tasoulis & Vrahatis 05) Tasoulis, D. K. & Vrahatis, M. N., *Unsupervised Clustering on Dynamic Databases*. Pattern Recognition Letters, article in press.

(Whittaker & Hirschberg 01) Whittaker, S. & Hirschberg, J., *The Character, Value and Management of Personal Paper Archives*. ACM TOCHI, vol.8(2), pp.150-170.

(Yang & Liu 99) Yang, Y. & Liu, X., *A Re-examination of text categorization methods*. In SIGIR 1999.

(Yang & Pederson 97) Yang, Y. & Pederson, J., *A Comparative Study on Feature Selection in Text Categorization*. In International Conference on Machine Learning, 1997.

Information Management: The Parmenides Approach

Alexander Mikroyannidis[1], Apostolos Mantes[2], Christos Tsalidis[3]
[1]School of Informatics, University of Manchester,
Sackville Street, Manchester M60 1QD, United Kingdom
A.Mikroyannidis@manchester.ac.uk
[2, 3]Neurosoft S.A., 24 Kofidou Str., 142 31 N. Ionia, Athens, Greece
{mantesap, tsalidis}@neurosoft.gr

ABSTRACT
The amounts of heterogeneous data that an organization is provided nowadays, has made information management a seriously complicated task. In the context of the present work, a platform for information management through document analysis and warehousing is introduced. Workflows employing Natural Language Processing (NLP) and Information Extraction (IE) components are used in order to extract semantic information from documents. This information is warehoused in a way to enable its efficient retrieval as well as further discovery of knowledge with the use of data mining methodologies.

Keywords
Information Management, Information Extraction, Document Analysis, Document Warehousing, Semantic Annotation, Workflow.

1. INTRODUCTION

The rapid rate of growth in the amount of information available these days has led information overload to hit tragic proportions. The situation has grown worse with the propagation of the World Wide Web, which contains vast amounts of unstructured and diverse information. Consuming and exploiting all this knowledge is extremely difficult, yet it can provide tremendous benefits.

In the context of the IST project Parmenides (IST-2001-39023), a framework for information management was developed. The Parmenides framework covers all phases of the document lifecycle, from collection to storage and retrieval. Information extraction (IE) and Natural Language Processing (NLP) techniques are utilized in order to achieve semi-automatic semantic annotation. A specialized XML schema accommodates the document metadata and facilitates indexing for efficient knowledge retrieval. All individual components are integrated under a software platform.

The remainder of this paper is organized as follows. Section 2 presents some related work that has been carried out in the area of information extraction and management. The document lifecycle as this is defined in the Parmenides framework is introduced in section 3. Section 4 describes in detail the information management tasks that the user can execute in the Parmenides integrated software platform. Finally, the paper is concluded in section 5.

2. RELATED WORK

Numerous systems with varying capabilities on information extraction and management have been developed. GATE [5] is an architecture for language engineering developed at the University of Sheffield, containing a suite of tools for language processing and information extraction. GATE's IE system, ANNIE, is rule-based, which means that unlike machine-learning based approaches, it requires no training data [11]. On the other hand, it requires a developer to create rules manually, so it is not totally dynamic. The architecture consists of a pipeline of processing resources which run in series. Many of these processing resources are language and domain-independent, so that they do not need to be adapted to new applications [10]. Pre-processing

stages include word tokenization, sentence splitting, and part-of-speech tagging, while the main processing is carried out by a gazetteer and a set of grammar rules. These generally need to be modified for each domain and application, though the extent to which this is necessary depends on the complexity and generality of the domain. Due to its open and extensible architecture, GATE has been used in various domains [6, 9, 12].

IBM's Unstructured Information Management Architecture (UIMA) [1] is a software architecture for developing and deploying unstructured information management (UIM) applications. A UIM application can be generally characterized as a software system that analyzes large volumes of unstructured information in order to discover, organize, and deliver relevant knowledge to the end user [8]. In analyzing unstructured content, UIM applications make use of a variety of technologies including statistical and rule-based natural language processing, information retrieval, machine learning, ontologies, and automated reasoning. UIM applications may consult structured sources to help resolve the semantics of the unstructured content [7].

Ellogon [2] is a text engineering platform developed by the Software and Knowledge Engineering Laboratory of N.C.S.R. "Demokritos", Greece. Ellogon is a multi-lingual, cross-platform, general-purpose text engineering environment, aiming to aid both researchers in the natural language field or computational linguistics, as well as companies that produce and deliver language engineering systems [13]. An infrastructure is provided for managing, storing and exchanging textual data and the associated linguistic information. Ellogon is based on a modular architecture that allows the reuse of Ellogon sub-systems in order to ease the creation of applications targeting specific linguistic needs.

3. THE PARMENIDES ARCHITECTURE

Figure 1 illustrates the architecture adopted in the Parmenides framework. It mainly consists of components for the construction and maintenance of the *Document Warehouse* (DW). The document lifecycle begins with its collection and conversion to an XML-based annotation scheme. The documents can be of various types: web pages, e-mails, word processor documents, etc. An agent-based document collector gathers these from various sources and converts them into XML for further processing. The utilized XML scheme is called *Common Annotation Scheme* (CAS) [14]. CAS allows a wide range of annotations: structural, lexical and semantic or conceptual.

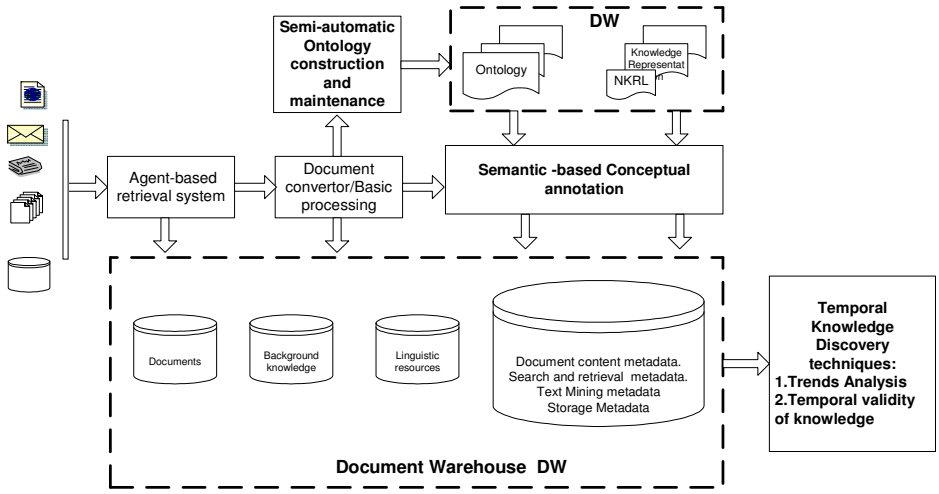


Figure 1. The Parmenides architecture

Next, a conceptual annotator extracts temporal semantic information from the documents, by applying information extraction and lexical chaining techniques, including tokenization, part of speech tagging [15], sentence splitting, categorization [4, 17], ontological lookup and rule-based IE [3, 16]. During this process, the Domain Knowledge is provided by Domain Ontologies, through their classification and organization of the concepts of the domain. The semantic annotations are stored inside the document with the use of the proper CAS structures, thus enriching the document with metadata.

The analysis of documents is organized in *workflows* (user-defined pipelines of analysis tasks), which can be fully customized by the user. New analysis components can be added and others can be removed, according to the information that the user seeks to retrieve. At the end of each workflow, the documents are stored in the DW. The DW contains both documents and metadata based upon the extracted semantic contents of the documents.

The DW is further exploited in a number of ways. Temporal knowledge discovery is achieved through the incorporation of temporal data mining methodologies. Trends analysis and pattern evolution is performed in order to trace significant changes in patterns over time. By providing the means for several ways of efficient exploitation, the Parmenides framework holds the potential for serving as a source of Business Intelligence.

4. THE RESOURCE MANAGER PLATFORM

The Resource Manager is a software platform that integrates the resources used in the Parmenides architecture for the Document Warehouse construction and maintenance. Through the Resource Manager, the user can manage the components used for the collection, conversion and analysis of documents. The ability to organize these components in workflows is also provided. Finally, the user can inspect and perform queries on the documents and their metadata.

4.1 Components

Every component is characterized by the type (format) of the documents it can process and the type of the documents it produces. These document types are identified with URLs pointing to the schema through which the document's contents are validated. The *Input URL* identifies the format of the documents that the component can understand and process while the *Output URL* identifies the format of the produced document contents. These *Input & Output URLs* restrict also the way the components can be put in a pipeline. In order to put component B after component A in a pipeline, the *Output URL* of A must be the same with *Input URL* of the component B. The Workflow editor uses these URLs when the user creates a workflow. In the toolbar of the existing components that can be put in a workflow it activates only the compatible components with the active (selected) component in the pane. So if the user wants to extent the workflow by putting another component in the pane, he can do by choosing a compatible component.

Depending on the way that a component produce output documents, components are divided into two categories: *Document* and *Corpus*. The *Document* components process a document at a time and produce one or more documents as the result of their process. On the other side, *Corpus* components process a collection of documents (a corpus) and after finishing the process of the whole collection, they produce one or more result documents.

The way an analysis component can be added in the Resource Manager is shown in Figure 2. The user needs to provide the following information for each component he adds:

- *Component name*: The name with which the component will be referred to.

- *Component Class*: this must be filled with the full Java Class name of the component.
- *Component Config*: a URL designating a file that contains the component's default configuration file. The file must contain valid XML.
- *Component Icon*: a URL designating a file that points to a small image file. This will be used when creating a visual representation for the component within the application.
- *Component Schema*: a URL designating a XML Schema. This is the XML Schema to which all the component's configurations must conform.

4.2 Queues

Queues are used as a buffering and synchronization mechanism of the workflow execution. Components can process their documents in different speeds so we need a buffer mechanism in order to be able to efficiently connect a slow and quick component. We can also have components that communicate with foreign systems and must operate in a real time fashion. They can not be blocked by the next component in the workflow and they want to deliver their produced documents as quick as possible. Most importantly, queues are necessary in parallel execution of the components in a workflow. This can be achieved by permitting components to run in different threads. In this situation slow components can clone its instances in order to speedup their process time and the queues are a safe, efficient and widely used mechanism to synchronize different threads.

We distinguish three types of queues:

1. *Simple* queues that have one input and one output end. Documents are inserted in queue from the previous (in line) component and delivered to the next component in the order of insertion.
2. *Broadcaster* queues have one input and multiple output queues. Every document inserted in the queue is cloned in as many copies as the outputs of the queue. This queue is used if we want to create parallel lines of execution in the workflow.
3. *Concentrator* queues reverse the operation of the broadcasters. The have many inputs and one output. The concentrate documents from multiple parallel execution lines (pipelines) and drive them to a common output.

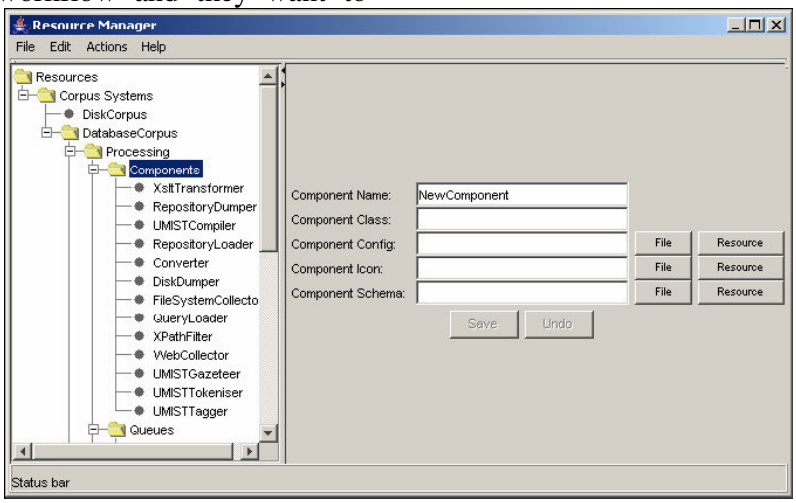


Figure 2. Adding a component in the Resource Manager

4.3 Workflows

Execution of a workflow is totally differentiated from the structure of the workflow. The *Workflow Executor* is the module that takes care of the way the workflow will be executed. Two intuitive execution mechanisms have been implemented leaving place for other more complex architectures:

1. The *Serial Executor* synchronizes the execution of all components in the same thread. This resembles the old single tasking operating systems where all programs share the same processing in a single thread of execution and the scheduler controls the schedule, i.e. the order and the time the programs will acquire the processor.
2. The *Parallel Executor* starts all the components simultaneously, letting them to synchronize themselves through the intermediary queues. Every component runs in its own thread achieving real parallelism in multiprocessor machines.

There are different views of what a workflow is:

- *Workflows as mathematical formulas*. Considering the different type of components as different operators and the different type (format) of documents as operands of different types, then the workflow is the mathematic expression representing the result of the processing.
- *Workflows as parallel flows of execution*. Documents are the processed units flowed from a processing element (component) to the next.
- *Workflows expressing the agent metaphor*. Components do not need to be installed and executed locally in the same machine. Implementing proxy components of remote processors or message queues connecting distributed components, we can achieve distributed workflow execution.

Figure 3 illustrates workflow management in the Resource Manager. By clicking at a component inside the workflow visualization area, the user is able to inspect the component's XML configuration in a tree-like manner. The editor which is being used is aware of the Components XML Schema, thus enforcing that the configuration's integrity will be kept, as far as the syntax is concerned.

The blue arrows mark the route that documents will follow during their processing. The toolbar buttons allow the insertion of new components into the workflow.

The rules under any workflow is constructed are the following:

- The workflow graph must be connected. No components or queues are permitted to be lying around without having at least an incoming or outgoing connection. Editing the workflow through the visualization panel ensures that this requirement is met.
- The workflow graph must not contain any circles. Editing the workflow through the visualization panel ensures that this requirement is met.
- All edges in the workflow are directed, either from a component to one ore more queues, or from a queue to one or more components. The valid graphs for Workflows are Directed Acyclic Graphs.

An example of a slightly more complicated workflow is shown in Figure 4. This workflow has the following effect: Supposing that the RepositoryLoader loads documents A, B and C from the document repository, the RepositoryDumper marked with X will store these documents in some location (e.g. /Documents/X). The same is true for the RepositoryDumper marked with Y. However, both X and Y will also forward the documents to the next Queue. It is clear now that the RepositoryDumper (Z) will create two versions in the Repository for each of the documents A, B and C, since the first NormalQueue doubles the number of incoming document instances.

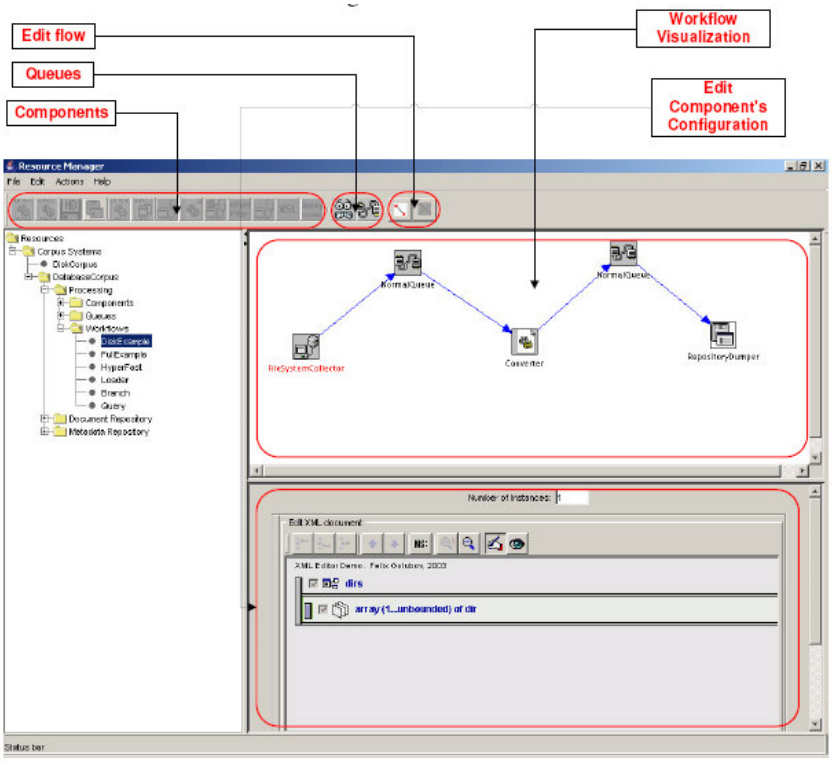


Figure 3. Workflow management in the Resource Manager

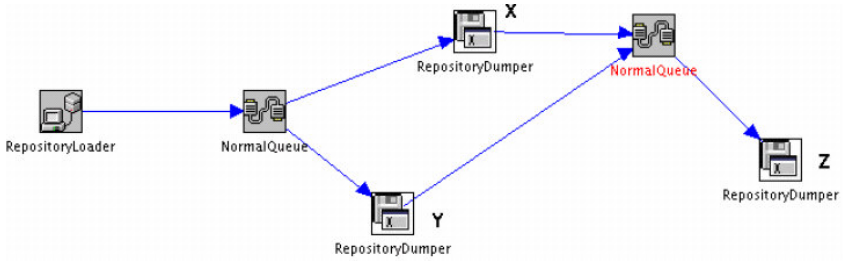


Figure 4. Example of a Resource Manager workflow

4.4 Document Repository

It is possible to create queries in order to search for documents that satisfy certain criteria. For this reason, the Resource Manager has the ability to create and execute queries on the Document Repository.

In the example of Figure 5, there are two panels that appear next to the tree view. The one shows the actual query, whereas the other one shows the results that the query produces.

Document Repository Queries may vary from a simple expression, to a complex logical expression using operators AND, OR and NOT on simple expressions. As far as a simple query is concerned, we can request that results must satisfy a query by filename, a query for a specific word, a query for a specific XPath[1] rule for XML files and finally, a query for one or more words under a specific XPath.

[1] http://www.w3.org/TR/xpath

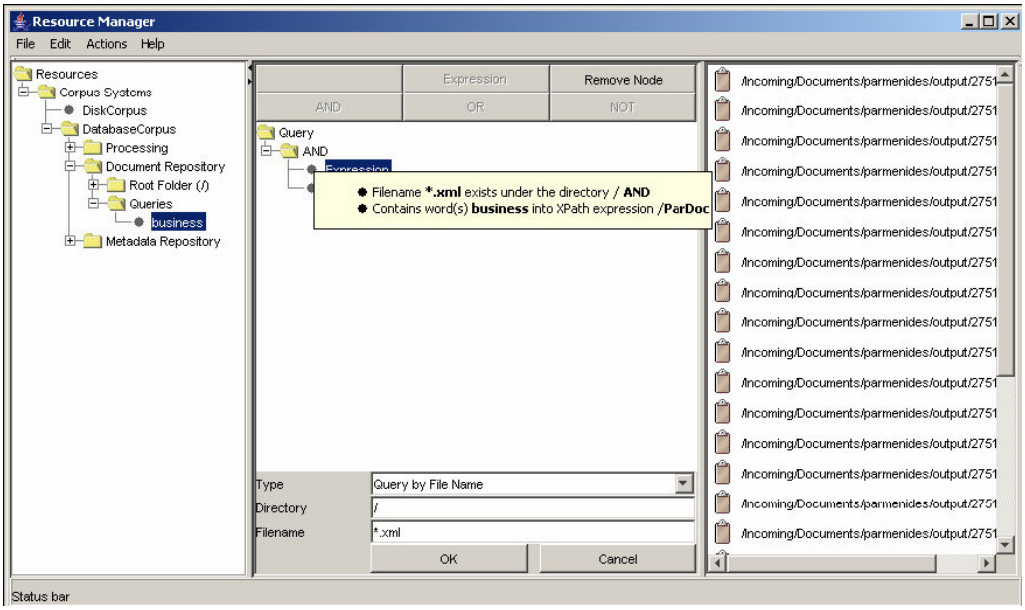


Figure 5. Query example in the Document Repository

4.5 Metadata Repository

The semantic information gathered during the analysis of the documents is accommodated in the Metadata Repository. The Metadata Repository allows for warehousing and retrieval of the documents' semantics.

An interface for visually building queries in the Metadata Repository is provided by the Resource Manager. The user can use the tables of the repository to form queries in order to retrieve specific semantics. For example, a query could be targeted to the products that have been developed in the last 6 months and the developing company has participated in a market alliance. Each query is translated into XML Query (XQuery)[2] and is applied on the CAS documents of the metadata repository. Figure 6 shows an overview of views and query capabilities of the Resource Manager.

Views of all types of metadata are available in the Resource Manager. For example, the user can inspect the named entities, time expressions, events and relations that have been discovered in the analyzed documents.

Queries on the Metadata Repository retrieve tuples instead of documents from the existing views on the metadata table. The Query Editor is similar to the Document Repository query editor. As seen in Figure 6, there are three sections below the "Query" node. The first section is the Tables section. The user chooses which views should be used in the query. Since the editor produces an SQL statement to be executed, a fair question is how these tables or views are joined, and with what criteria. The answer is that all views used are joined so that their tuples refer to the same document. All views contain the document id for the document that they refer to. As a result, when we are looking for documents with entity type "heart_disease", the two views are joined by the document id they refer to.

The second section is the "Columns" section. The user must add all the columns of existing views he wants to view.

The Rules section allows the user, through a mixture of simple expressions, to limit the number of tuples to those that satisfy various criteria. The example of Figure 6 utilizes a simple expression defining that only PEntities with type equal to "heart_disease" should be returned.

[2] http://www.w3.org/XML/Query

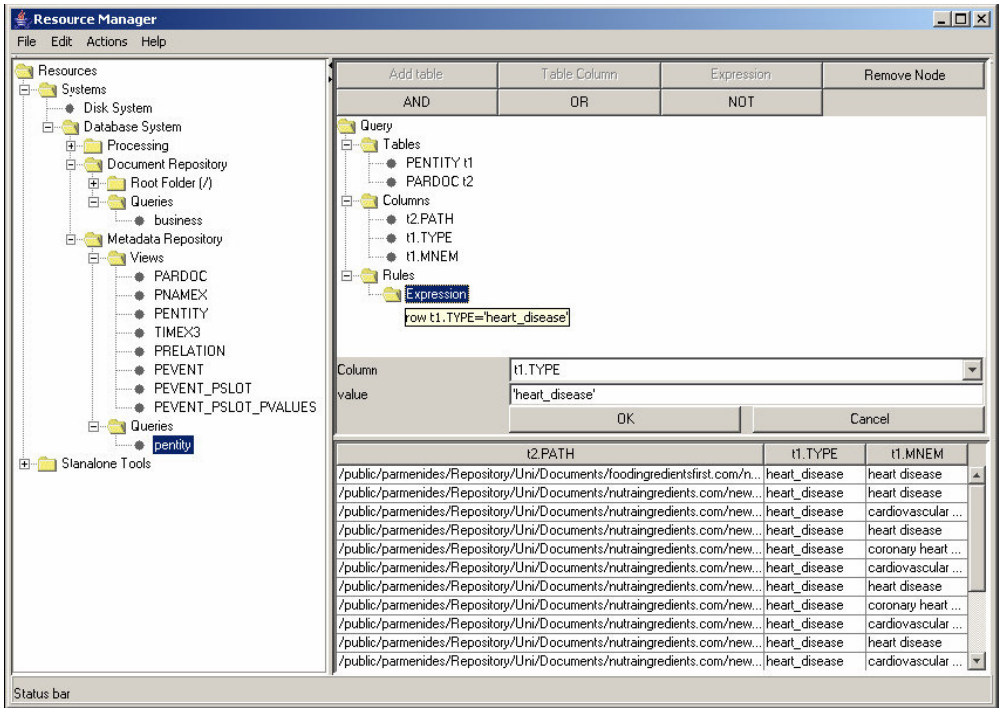


Figure 6. Views and queries on the Metadata Repository

5. CONCLUSIONS

The present work investigated the area of information management. A platform was presented, which allows for document analysis, warehousing and retrieval. Through the use of customized workflows, documents are gathered, converted to a specialized XML schema, analyzed with NLP components and stored in a warehouse. The warehouse provides the ability of querying on metadata, as well as on the documents themselves.

The proposed platform provides the ability to efficiently manage large amounts and different types of organizational data. Nevertheless, its ultimate purpose is the extraction of Business Intelligence. The metadata warehouse holds the resources needed for data mining methodologies that will lead to the discovery potentially significant knowledge.

6. ACKNOWLEDGMENTS

The Parmenides project is co-funded by the European Commission (contract No. IST-2001-39023) and the project partners, as well as the Swiss Federal Office for Education and Science (BBW/OFES). Please see http://www.crim.co.umist.ac.uk/parmenides for a detailed description of the project.

7. REFERENCES

[1] The Unstructured Information Management Architecture Project. http://www.research.ibm.com/UIMA/, 2004.

[2] The Ellogon Language Engineering Platform. http://www.ellogon.org/, 2005.

[3] Black, W., McNaught, J., Vasilakopoulos, A., Zervanou, K., and Rinaldi, F. CAFETIERE: Conceptual Annotations for Facts, Events, Terms, Individual Entities and RElations. UMIST 2003.

[4] Cortes, C. and Vapnik, V. Support Vector Networks. *Machine Learning*, 20, 3 (1995), 273-297.

[5] Cunningham, H., Maynard, D., Bontcheva, K., and Tablan, V. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In *Proceedings of the 40th Anniversary Meeting*

of the Association for Computational Linguistics (ACL'02) (2002).

[6] Cunningham, H., Tablan, V., Bontcheva, K., and Dimitrov, M. Language engineering tools for collaborative corpus annotation. In *Proceedings of the Corpus Linguistics 2003* (Lancaster, UK, 2003).

[7] Ferrucci, D. and Lally, A. Building an example application with the unstructured information management architecture. *IBM Systems Journal*, 43, 3 (2004), 455-475.

[8] Ferrucci, D. and Lally, A. UIMA: An Architectural Approach to Unstructured Information Processing in the Corporate Research Environment. *Natural Language Engineering*, 10, 3 (2004), 327–348.

[9] Manov, D., Kiryakov, A., Popov, B., Bontcheva, K., Maynard, D., and Cunningham, H. Experiments with geographic knowledge for information extraction. In *Proceedings of the Workshop on Analysis of Geographic References* (Edmonton, Canada, 2003).

[10] Maynard, D. and Cunningham, H. Multilingual Adaptations of a Reusable Information Extraction Tool. In *Proceedings of the Demo Sessions of EACL'03* (Budapest, Hungary, 2003).

[11] Maynard, D., Tablan, V., and Cunningham, H. NE recognition without training data on a language you don't speak. In *Proceedings of the ACL Workshop on Multilingual and Mixed-language Named Entity Recognition: Combining Statistical and Symbolic Models* (Sapporo, Japan, 2003).

[12] Maynard, D., Yankova, M., Kourakis, A., and Kokossis, A. Ontology-based information extraction for market monitoring and technology watch. In *Proceedings of the End User Aspects of the Semantic Web Workshop, 2nd European Semantic Web Conference (ESWC 2005)* (Heraklion, Greece, May 29-June 1, 2005), 33-42.

[13] Petasis, G., Karkaletsis, V., Paliouras, G., Androutsopoulos, I., and Spyropoulos, C. D. Ellogon: A New Text Engineering Platform. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC-2002)* (Las Palmas, Canary Islands, Spain, 2002), 72-78.

[14] Rinaldi, F., Dowdall, J., Hess, M., Ellman, J., Zarri, G. P., Persidis, A., Bernard, L., and Karanikas, H. Multilayer annotations in Parmenides. In *Proceedings of the K-CAP2003 Workshop on Knowledge Markup and Semantic Annotation* (Sanibel, Florida, USA, 2003).

[15] Vasilakopoulos, A. Improved Unknown Word Guessing by Decision Tree Induction for POS Tagging with TBL. In *Proceedings of the 6th Annual CLUK Research Colloquium* (Edinburgh, UK, 2003).

[16] Vasilakopoulos, A., Bersani, M., and Black, W. A Suite of Tools for Marking Up Temporal Text Mining Scenarios. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC 2004)* (Lisbon, Portugal, 2004), 807-810.

[17] Yu, H., Zhai, C., and Han, J. Text classification from positive and unlabeled documents. In *Proceedings of the 12th international conference on Information and knowledge management* (New Orleans, LA, USA, 2003), 232-239.

Semantic Indexing: Cognitive Maps Based Approach

Vladimir F. Khoroshevsky

Computer Center RAS,
40 Vavilov str., Moscow,Russia, GSP-1
Khor@ccas.ru

Victor P. Klintsov

AviComp AG,
84/2, Vernadskogo Av., Moscow, Russia, 119606
Victor.Klintsov@avicomp.ru

Abstract

An approach to the semantic indexing, based on the usage of cognitive maps, which are the result of the text mining with OntosMiner family systems, is presented in the paper. Special metrics to determine meaningful fragments of the internal representation of the processed texts are proposed. The usage of the scoring functions for clustering/categorization and summarization is the practical result of the presented approach.

1. Introduction

The aim of this research is to present an approach to the semantic indexing of the cognitive maps generated by the text mining systems. This approach is based on the usage of the content extraction (CE) technologies (Maynard, et all. 02), as well as special metrics to determine meaningful fragments of the internal representation of the processed texts. In contrast to standard approaches to the semantic indexing problem solving, we propose the following technology chain: using of the text mining to extract the meaning of processed text in context of appropriate domain ontology; presentation of extracted text meaning in a special semantic network (cognitive map) form; using special scoring functions based on cognitive maps. It is shown that the proposed metrics and scoring functions can be used for clustering/categorization of the documents from collections, as well as for summaries generation.

The presentation is organized as follow. Brief discussion of text mining systems (OntosMiner family) is given in the next section. Within this discussion we'll show that these systems can be used for the content extraction and that the results may be used for semantic indexing. Next section of the paper is connected to the presentation of the special method for the cognitive map indexing. Then we discuss the proposed scoring functions usage for the clustering/categorization and summarization problems resolving and present an example of document indexing. Last section of the paper outlines the results and future works.

2. OntosMiner family systems

General goal of text mining systems is to get an adequate representation of the text meaning. There are many projects and practically useful systems in this domain (LREC 04). OntosMiner family systems which are oriented to the content extraction from multilingual documents, are in this scope. Their development and implementation is carried out by Russian IT-company Avicomp AG[1] together with Ontos AG[2] (Switzerland).

The main requirements for OntosMiner family systems are the following:

- Multilingual documents collection processing (English, French, German and Russian texts).
- Monothematic documents collection processing ("Business Duties" domain - information about IT-companies, their founding, investing, selling, buying, merging, top-management CVs, etc.).
- An adequate processing of relevant objects and relations, according to the concrete ontology.
- Representation of processing results in a form of a cognitive map, which is a kind of semantic network.
- Multi-platform implementation of all systems of the family.

OntosMiner project is based on the powerful multi-platform environment GATE that was created in Sheffield University, Great Britain (Cunningham, et all. 02) with appropriate extensions and additional components developed and implemented within our project. Main characteristics of OntosMiner family systems functionality are summarized in Tables 1, 2.

[1] *http://www.avicomp.ru*
[2] *http://www.ontos.ch*

Processing results are represented in the form of XML documents that specify cognitive maps. Such maps can be integrated into knowledge bases and visualized according to user demands and preferences.

Table 1.

Named Entities	Languages			
	En.	Fr.	Ge.	Ru.
Person	+	+	+	+
JobTitle/Title	+	+	+	+
Organization	+	+	+	+
Location	+	+	+	+
Date/Period	+	+	+	+
Address	+			+
Money	+	+	+	+
Percent	+	+	+	+
Degree	+	+	+	
Car				+

Table 2.

Semantic Relations	Languages			
	En.	Fr.	Ge.	Ru.
Affiliate	+			
Buy-Sell	+	+	+	
Employ	+	+	+	+
Found	+	+	+	
Graduate	+	+	+	
EarnDegree	+	+	+	
Invest	+	+	+	
Joint Venture	+	+		
Own	+	+	+	+
Rival	+	+	+	
LocatedIn	+	+	+	+
Reside				+
Belong				+
Petition				+
Investigate				+
Hijack				+
ConnectedWith	+	+	+	+
The Same?	+	+	+	+

Information about the companies from KM Top100 list (every year such a list is outlined by international journal "Knowledge Management") was used as evaluation corpus for OntosMiner family systems. This corpus consists of 150 texts, from 4 to 10 Kb each. In addition to such documents, at about 30 Russian texts from this domain were added into the corpus. Linguists from Moscow State University manually marked all texts of the corpus and so-call "Gold Standard" was received as a result of this activity.

Processing these texts with CE-systems from OntosMiner family give us the possibility to automatically compare the received results on the NEs (Named Entities) level and to compute such characteristics as precision, recall and F-measure. As it was shown in experiments the results are correspondent to the ones presented at TREC/MUC conferences. So, OntosMiner family systems can be used as front-end components for semantic indexing of the documents under processing.

3. Cognitive maps based indexing

3.1. General remarks

The traditional and advanced approaches to the indexing of text mining results are based on the using of the scoring functions. Such functions kinds depend of the goal of indexing. For example, if such a goal is the clustering/categorization of the documents scoring functions reflect the significance of presented within the documents key words and/or terms. In the case of the summarization task scoring functions are oriented at the estimation of sentence significance (as a rule, sentence position and its length, tf*idf for the sentences and, more rarely, headlines and CE-patterns) and calculate the total score of each sentence as the weighted sum of the partial scores (Sekine & Nobata 03).

Experimental results presented in (Nobata et al. 02) show that the biggest contribution is the sentence position and tf*idf. On the other hand, it seems that the scoring results are strongly depended on the texts category, their style, etc. For example, in (McKeown et al. 01) 4 categories of document sets based on the main topic taxonomy were outlined. Advanced taxonomy of the documents topics was proposed in (Sekine & Nobata 03). And it seems that each category has its own distribution of the importance of the scoring factors.

Our approach to semantic indexing of text mining results is connected with the searching of the stable factors for the extraction of the important information from the documents (not from the sentences) and the development of appropriate scoring functions suitable for the clustering/categorization and summarization as well.

3.2. Cognitive maps scoring: informal view

As it was mentioned above the meaning of the texts processed by CE-systems of OntosMiner family is presented by cognitive maps. And our research is in the development of balanced scoring functions based on these cognitive maps.

There are several points in the solving this task within the presented approach. The first one is connected with the using of domain ontology and the cognitive maps. Really, domain ontology is the important semantic filter for the text under processing course only those NE(s) types and generic relationships between them that represented

in the model of domain are involved into consideration and can receive their examples from the text. So, the first scoring function should reflect the quality of domain ontology and CE-systems. The first component can be estimate by human experts only. The second can be based on extended F-measures: for NE(s) and for relationships as well.

Next scoring function can be based on the using of the cognitive map's structural characteristics. Really, if the cognitive map that represents the meaning of the texts involved into indexation has one (several) NE(s) closely connected by semantically important relationships such a "clouds" should be estimate within the semantic index.

To illustrate above presented heuristics lets overview the cognitive map extracted by OntosMiner/English text mining system from the text about Google management team (GOOGLE homepage) presented in Figure 1.

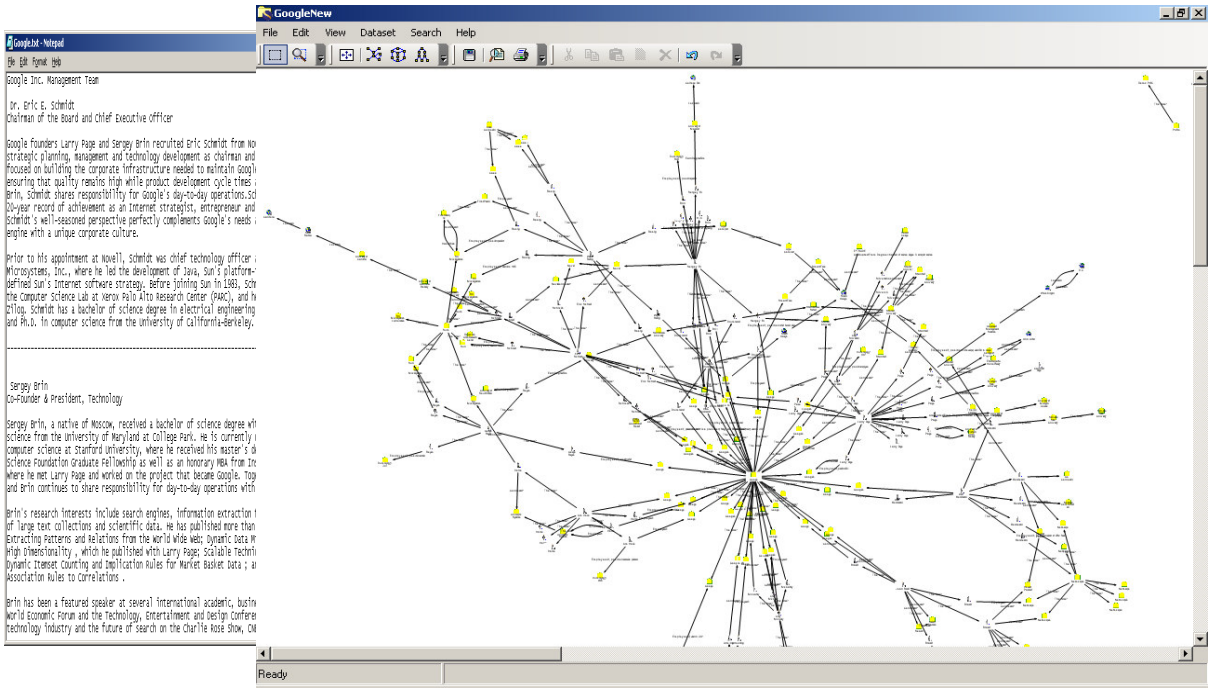


Figure 1: Cognitive map extracted from English text about Google

There are several focuses can be viewed at presented above cognitive map. The first one is Google titled node that can be outlined as the focus of the whole document (see Figure 2).

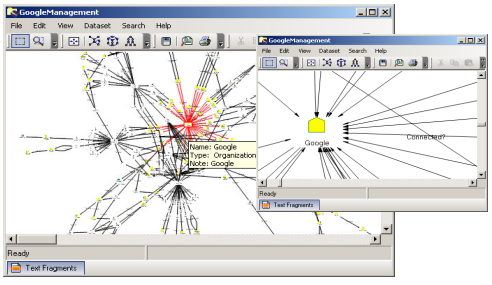


Figure 2: Main focus of cognitive map about Google

Common number of all links connected with Google node equals to 41. The number of links after merging the objects connected by "The Same?" links equals to 23.

The other focuses are connected with the main Persons of Google management team. Some of them (for example, focuses related to Sergey Brin and Larry Page presented in Figure 3) have a little distance from focus Google. And the number of links after merging the objects related to Person Sergey Brin connected by "The Same?" links equals to 10. The same characteristic for Larry Page is 16.

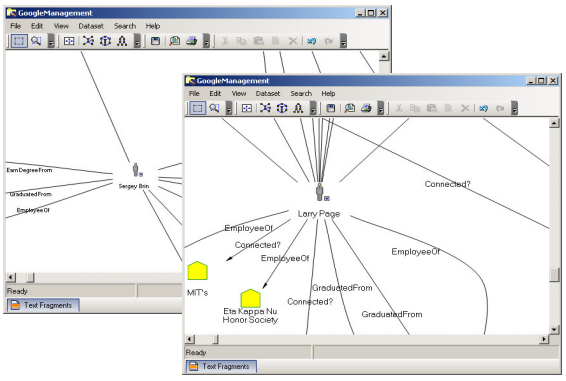


Figure 3: Focuses related to Persons Sergey Brin and Larry Page

Others (for example, focus related to Shona Brown presented in Figure 4) are at the periphery relatively to Google focus. And the number of links after merging the objects related to Person Shona Brown connected by "The Same?" links equals to 7.

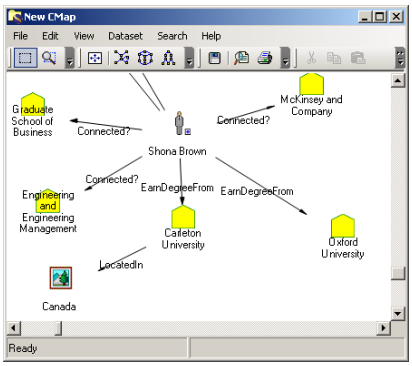


Figure 4: Focuses related to Person Shona Brown

It is interesting to outline that the length of "The Same?" relation chains has significant variations from one object to another. For example, this parameter for Organization Google equals to 41, for Person Sergey Brin – 23, for Person Larry Page – 25, and for Person Shona Brown – 10 only.

All presented above give us possibilities to state that our informal heuristics can be transform into special scoring functions based on cognitive maps metrics.

3.3. Cognitive maps metrics

To formalize above discussed heuristics lets use standard graph theory notions: nodes (objects), links (relations), node arity (the number of the links connected this node with another nodes of the graph), distance between two nodes of the graph (the length of the path between these nodes).

According to above, let's define the following notions:

Cognitive map (*CM*) – special kind of semantic network with nodes x_i that present extracted from the text NEs and links R_j that present extracted from the text semantic relations between nodes.

Distance between nodes *D(x, y)*

unweighted

$$D(x,\ y) = \begin{cases} \sum_{i=1}^{n} d_i,\ if\ \exists\ path(x,y) \\ 0,\ otherwise \end{cases}, \text{ where}$$

$d_i = 1,\ if\ \exists\ x_{i-1} R_j x_i$

weighted

$$D_w(x,\ y) = \sum_{i=1}^{n} w_i * d_i,\ where\ w_i = weight(R_i)$$

Above depicted formulas can be used *iff* two nodes connected by the single link. Otherwise the distance between such a nodes is reduced relatively to the number of the links connected these nodes and appropriate formulas for d_i should be transformed into the following:

$d_i = \frac{1}{N}$, where *N* is the number of the links corresponding to d_i (for unweighted case), and

$d_i = \sum_{k=1}^{M} \frac{\sum_{i=1}^{N_k} w_i}{N_k}$, where M is the number of links' groups (for weighted case).

Normalized node arity $\hat{K}(x)$

$\hat{K}(x) = \frac{K(x)}{\max(K(x_i))}$, where *K(x)* –arity of node *x*, and $x \in CM$

Let's also use the notations

$\hat{W}(x) = \hat{K}(x)$ for the absolute normalized weight of the node *x*, and

$\hat{W}(x|y) = \hat{K}(x) * e^{-D_w(x,y)}$ for the relative to the node *y* normalized weight of the node *x*.

Cognitive map "cloud" (it's useful to define this notion relatively to some node, for example, relatively to the focuses of the cognitive map)

$Cloud(f) = \{x_i \,|\, x_i \in node(CM)\ \&\ D(x_i, f) \le \sigma\}$, where σ is the threshold defined by the expert.

Focus *f(CM)* (**set of focuses** *fSet(CM)*) of the cognitive map

$f(CM) = x \,|\, x \in node(CM)\ \&\ \hat{K} = 1;$

$fSet(CM) = \{x_i \,|\, x_i \in node(CM)\ \&\ \delta \le \hat{K}(x_i) \le 1\}$, where δ is the threshold defined by the expert.

Weight of the "cloud" (absolute and relative to the focus)

$\hat{W}(Cloud) = \sum_{x_i \in Cloud} \hat{W}(x_i)$,

$\hat{W}(Cloud \mid f_{ext}) = \hat{W}(Cloud) * e^{-D_w(f(Cloud),\ f_{ext})}$.

According to above defined metrics we can estimate the structural importance of each focus of the cognitive map and structural importance of the "clouds" for each focus.

It is clear that the structural importance can be extended by semantic importance if we add to our formulas special coefficients reflected the importance of the nodes and relationships involved into consideration. One manner to determine these coefficients is the usage of F-measures for nodes and relations types. Another way to this problem solving is the usage of human expert estimations.

The next scoring function can be based on the estimation of coreference chains' length. Really, if within the cognitive map coreference chain length for some NE(s) is relatively long in comparison with the average value of this parameter it can be used for characterization of the document's category. So, each node can be accompanied by the "The Same?"-chain length (*CL*) and normalized "The Same?"-chain length (*NCL*) estimations:

$CL(x)$ = number*("The Same?" relations for node x)*,

and

$$NCL(x) = \frac{CL(x)}{\max(CL(x_i))}.$$

So, we discussed cognitive maps metrics and now can define scoring functions for the cognitive maps by the following way:

$$Score(f) = \hat{W}(f(CM)) + NCL(f(CM));$$

$$Score(fSet \mid y) = \sum_{x \in fSet(CM)} (\hat{W}(x \mid y) + NCL(x));$$

$$TotalScore(CM) = Score(f) + Score(fset \mid f).$$

It is necessary to outline that presented above metrics and scoring functions are connected with the cognitive map as a whole instead of the sentences scoring approach presented in (Nobata et al. 02).

4. Usage of cognitive maps indexing

4.1. Preliminary remarks

Our experiments with KM Top100 corpus show that the focuses and their types analysis together with the analysis of the coreference chains length can be used for the categorization/clustering of the documents and choosing of appropriate discourse scheme for the summarization.

4.2. An example of cognitive map indexing

To understand the advantages of cognitive maps indexing let's discuss the scoring of the text related to Google management team received from this company site (GOOGLE homepage). Extracted from this text cognitive map and its fragments were presented above in Figures 1 - 4.

The calculation of above defined metrics for this cognitive map brings us the possibility to outline 10 focuses (δ=0.4) in *fSet*:

fSet(Google_CM) = {Google, Omid Kordestani, Larry Page, Dr. Eric E. Schmidt, Sergey Brin, George Reyes, Shona Brown, Michael Moritz, Stanford University, Intel}.

Appropriate clouds for these focuses and their normalized weights are summarized below in Tables 3 (for the simplicity σ=1).

Table 3.

fSet	$\hat{W}(Cloud)$
Google	6,391304
Stanford University	5,521740
Intel	4,391306
Dr. Eric E. Schmidt	3,347824
Larry Page	3,260869
Sergey Brin	3,086956
Michael Moritz	2,913043
Omid Kordestani	2,565214
Shona Brown	1,826086
George Reyes	1,521737

It is easy to observe that only 6 focuses of *Google_CM* possess with the clouds weights up to 3.0. So, let's concentrate below at these focuses and appropriate clouds.

Focus "Google" has maximal weight. The members of its cloud with their normalized weights (δ=0.4; σ=1) are presented in Table 4.

Table 4.

Clouds' NE	$\hat{W}(NE)$
Wayne Rosing	0,347826
George Reyes	0,434783
Dr. Eric E. Schmidt	0,521739
Cindy McCaffrey	0,130435
Larry Page	0,695652
Sergey Brin	0,434783
Stanford(location)	0,086956
Toronto	0,086956
Los Angeles	0,086956
Shona Brown	0,695652
John Doerr	0,260870
John Hennessy	0,260870
Stanford University	0,565217
Arthur Levinson	0,260870
Omid Kordestani	0,739130
Ram Shriram	0,260870
Jonathan Rosenberg	0,173913
David C. Drummond	0,347826

The last of the focuses involved into consideration is the focus "Sergey Brin". Short fragment of the text related to this Person and Person "Larry Page" presented below:

Sergey Brin
Co-Founder & President, Technology
Sergey Brin, a native of Moscow, received a bachelor of science degree with honors in mathematics and computer science from the University of Maryland at College Park. He is currently on leave from the Ph.D. program in computer science at Stanford

University, where he received his master's degree. Brin is a recipient of a National Science Foundation Graduate Fellowship as well as an honorary MBA from Instituto de Empresa . It was at Stanford where he met Larry Page and worked on the project that became Google. Together they founded Google Inc. in 1998, and Brin continues to share responsibility for day-to-day operations with Larry Page and Eric Schmidt.

……...

Larry Page

Co-Founder & President, Products

Larry Page was Google's founding CEO and grew the company to more than 200 employees and profitability before moving into his role as president, Products in April 2001. He continues to share responsibility for Google's day-to-day operations with Eric Schmidt and Sergey Brin.

The son of Michigan State University computer science professor Dr. Carl Victor Page, Page's love of computers began at age six. While following in his father's footsteps in academics, Page became an honors graduate from the University of Michigan, where he earned a bachelor of science degree in engineering, with a concentration on computer engineering. During his time in Ann Arbor, Page served as president of the University's Eta Kappa Nu Honor Society and built a programmable plotter and inkjet printer out of Lego™.

While in the Ph.D program in computer science at Stanford University, Page met Sergey Brin and together they developed and ran Google, which began operating in 1998. Page went on leave from Stanford after earning his master's degree. He was granted an honorary MBA by Instituto de Empresa and was the first recipient of the University of Michigan Alumni Society Recent Engineering Graduate Award.

Appropriate fragment of Google_CM presented in Figure 5.

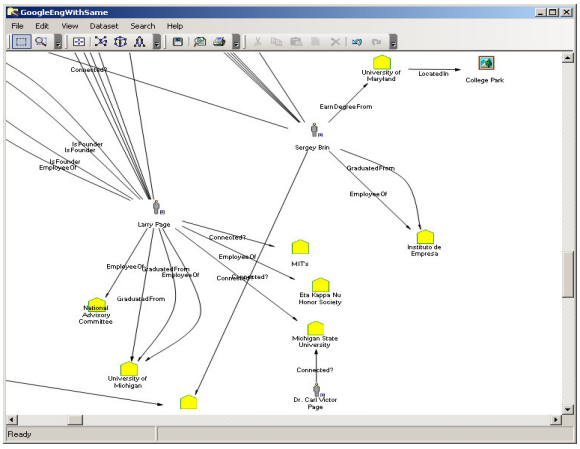


Figure 5: Fragment of Google_CM

The clouds associated with the focuses “Larry Page” and “Sergey Brin” with their normalized weights (δ=0.4; σ=1) are summarized below in Tables 5, 6 respectively.

Table 5.

Clouds' NE	$\hat{W}(NE)$
Google	1,000000
Sherpalo	0,391304
Kleiner Perkins Caufild Byers	0,130435
University of Michigan	0,130435
Michigan State University	0,086957
Eta Kappa Nu Honor Society	0,043478
National Advisory Committee	0,043478
MIT	0,043478
Stanford University	0,565217
Intel	0,434783
Sequoia Capital	0,391304

Table 6.

Clouds' NE	$\hat{W}(NE)$
Google	1,000000
Novell	0,086956
Kleiner Perkins Caufild Byers	0,130435
Sherpalo	0,391304
Instituto de Empresa	0,086957
Stanford University	0,565217
Intel	0,434783
Sequoia Capital	0,391304

At the basis of above presented data it is possible to outline main focus of this cognitive map and to calculate all clouds weights relatively to this main focus. The results of appropriate calculations are summarized in Table 7.

Table 7.

f	$\hat{W}(Cloud)$	$D(NE, f)$	$\hat{W}(Cloud \setminus f_{ext})$	%
Google	6,391304	0,00	6,391304	100
Intel	4,391306	0,20	3,595297	56,2
Stanford University	5,521740	1,00	2,031335	31,8
Larry Page	3,260869	0,25	2,539567	39,7
Dr. Eric E. Schmidt	3,347824	1,00	1,231596	19,3
Sergey Brin	3,086956	1,00	1,135628	17,8
Michael Moritz	2,913043	1,00	1,071649	16,8
Omid Kordestani	2,565214	1,00	0,943689	14,8
George Reyes	1,521737	0,50	0,922980	14,4
Shona Brown	1,826086	1,00	0,671779	10,5

According to presented results we can see that main focus connected with object “Google” and its cloud has outlined weight in comparison with nearest objects “Intel” and “Stanford University”. The situation is the same relatively to objects of Person type.

So, let’s try to understand the situation with the coreference chains’ length. Appropriate data with nonzero values are presented in Table 8 (for the objects of Person type) and in Table 9 (for the objects of Organization type) that were extracted by OntosMiner/English system from text about Google management team.

Table 8.

NE	*CL(NE)*	*NCL(NE)*
Larry Page	25	1,00
Sergey Brin	23	0,92
Dr. Eric E. Schmidt	16	0,64
Wayne Rosing	12	0,48
David C. Drummond	11	0,44
George Reyes	10	0,40
Shona Brown	10	0,40
Omid Kordestani	9	0,36
Jonathan Rosenberg	8	0,32
Cindy McCaffrey	8	0,32

Table 9.

NE	*CL(NE)*	*NCL(NE)*
Google	41	1,00
Stanford University	9	0,22
Sun	8	0,19
University of Michigan	3	0,07
McKinsey and Company	2	0,05
Intel	1	0,02
Instituto de Empressa	1	0,02

And now we can calculate the scoring functions values for our cognitive map. The results of appropriate calculations are summarized in Table 10.

Table 10.

f	$\hat{W}(Cloud \mid f_{ext})$	*NCL(NE)*	*Score(f)*
Google	6,391304	1,00	7,391304
Intel	3,595297	0,02	3,615297
Stanford University	2,031335	0,22	2,251335
Larry Page	2,539567	1,00	3,539567
Dr. Eric E. Schmidt	1,231596	0,64	1,871596
Sergey Brin	1,135628	0,92	2,055628
Michael Moritz	1,071649	0,00	1,071649
Omid Kordestani	0,943689	0,36	1,303689
George Reyes	0,922980	0,40	1,32298
Shona Brown	0,671779	0,40	1,071779

It is clear from data presented in Table 10 that the set of objects involved into consideration is not changed. The only difference connected with the coreference chains' length is in the changing of importance two objects – "Dr. Eric E. Schmidt" and "Sergey Brin". And it is not wonder course the text related to Google company, and Larry Page and Sergey Brin are its founders, and Dr. Eric E. Schmidt is the Chairman of the Executive Committee and Chief Executive Officer of this company.

Finally, scoring functions for above discussed text are the following:

Score(Google) = 7,391304,
Score(fSet | Google) = 18,10352, and
TotalScore(Google_CM) = 25,49482.

4.3. Semantic index usage

Presented above analysis shows that text about Google management team may be categorized to Single-organization and Multi-person types in terms of (Sekine & Nobata 03) categories.

Such a results can be used in information retrieval task (to estimate the relevance of processed documents to users' needs according their scoring), in categorization/clustering of the documents (documents that have similar scoring results may be gathered in cluster), and in summarization (to choose appropriate discourse scheme for summary generation).

5. Conclusion

An approach to the semantic indexing based on the using cognitive maps that are the result of text mining with OntosMiner family systems were presented in the paper. Special metrics to determine meaningful fragments of the internal representation of the processed texts were proposed and discussed. The usage of scoring functions for categorization/ clustering documents collections and documents summarization is the practical result of presented approach.

References:

(Cunningham, et all. 02) Cunningham, H., Maynard D., Bontcheva K., Tablan V., *GATE: an Architecture for Development of Robust HLT Applications*, Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, July 2002.

(GOOGLE homepage) http://www.google.com

(LREC 04) *Proceedings of the 4th International Conference On* Language Resources And Evaluation (LREC 2004), Lisbon, Portugal, 26-28 May 2004.

(LREC 04) *Proceedings of the 4th International Conference On* Language Resources And Evaluation (LREC 2004), Lisbon, Portugal, 26-28 May 2004.

(Maynard, et all. 02) Maynard D., Bontcheva K., Cunningham. H. *Toward a semantic extraction of named entities*, Proceedings of the AIMSA 2002, Varna, Bulgaria, 2002.

(McKeown et al. 01) K.R. McKeown, R. Barzilay, D. Evans, V. Hatzivassiloglou, M.Yen Kan, B. Schiffman, S. Teufel, Colmbia Multi-Document Summarization: Approach and Evaluation, In Proc. of the Document Understanding Conference (DUC-2001), 2001.

(Nobata et al. 02) Chikashi Nobata, Satoshi Sekine, Hitoshi Isahara, Ralph Grishman, Summarization System Integrated with Named Entity Tagging and IE patterns Discovery, Proc. of the LREC-2002 Conference, May 2002.

(Sekine & Nobata 03) Satoshi Sekine, Chikashi Nobata, A Survey for Multi-Document Summarization, In Proc. of HLT-NAACL 2003 Workshop: Text Summarization, DUC03, Edmonton, Canada, May-June 2003.

Text Mining Software Survey

Haralampos Karanikas[1] **and Thomas Mavroudakis**[2]

[1] University of Manchester, UK
karanik@co.umist.ac.uk
[2] National & Kapodistrian Univ. of Athens,
Knowledge Management Lab., Hellas

Abstract

The area of Knowledge Discovery in Text (KDT) and Text Mining (TM) is growing rapidly mainly because of the strong need for analysing the vast amount of textual data that reside on internal file systems and the Web. In this paper we provide an overview of the area and main approaches. We propose a feature classification scheme that can be used to compare and study text mining software. This scheme is based on the software's general characteristics and text mining features. We then apply our feature classification scheme to investigate the most popular software tools, which are commercially available. Finally we discuss the main issues that arise from the comparison of the tools and offer possible future directions. Our intention is not to conduct an extensive presentation of every tool in the area; only representative tools for each of the main approaches identified are considered.

1 Introduction

We are living in the "Information Age". The main characteristic is the amazing growth of data that are being generated and stored making it difficult for humans to understand. In most companies and organisations employee time and effort is wasted in ineffective searches through multiple information sources including web sites and other conventional sources. This problem of information overload is further exacerbated due to the unstructured format of the majority of the data. The vast amount of data found in an organisation, some estimates run as high as 80%, are textual such as reports, emails, etc. [Mer00] This type of unstructured data usually lacks metadata (data about data) and as a consequence there is no standard means to facilitate search, query and analysis. On the other hand Web is the biggest document collection and most of the current Web's content is designed for humans to read and not for computer programs. To date, the Web has developed a medium of documents for people rather than for data and information that can be processed automatically [Bern01].

While the amount of textual data available to us is constantly increasing, our ability to understand and process this information remains constant. A human editor can only recognise that a new event has occurred by carefully following all the web pages or other textual sources. This is clearly inadequate for the volume and complexity of the information involved. The need for automated extraction of useful knowledge from huge amounts of textual data in order to assist human analysis is apparent. The rapid adoption of the e-commerce business model is driving the demand for software that will help companies to analyse, manage, manipulate and effectively control how business disseminate and leverage information to competitive advantage [Mer00].Knowledge discovery and Text Mining are mostly automated techniques that aim to discover high level information in huge amount of textual data and present it to the potential user (analyst, decision-maker, etc).

The aim of this study is fourfold:

- Provide an overview of existing techniques that can be used to extract information from textual sources.
- Provide a feature classification scheme that identifies important characteristics according to which knowledge discovery and text mining software tools can be studied.
- Apply our feature classification scheme to investigate existing software tools.
- Discuss the main issues that arise from the comparison of the tools and the possible future directions.

In this paper we have included tools with focus on Text Mining, especially these that combine natural language processing with data mining and machine learning techniques.

The rest of the paper is organised as follows. In section 2 we discuss the process of analysing vast amount of textual data. We provide the definitions that will be used throughout the paper and describe the various analysis tasks as well as the techniques employed to complete these tasks. In section 3 we present the proposed feature classification scheme, which will be used to review the text mining software. Finally, in section 4 we discuss the survey findings; the main techniques adopted and offer possible future directions. In particular we focus on two important issues, namely the *framework architecture* and the *methodologies*. At the end the resulting comparison tables are provided.

2 Knowledge Discovery in Text and Text Mining

This section provides an introduction to the area of knowledge discovery in text, serving as background explanation for our feature classification scheme and the software feature tables 1, 2 and 3. After we briefly explain our use of the terms

knowledge discovery in text (KDT) and text mining (TM) in section 2.1, we describe the text mining operations.

2.1 Overview

KDT and TM is a new research area that tries to resolve the problem of information overload by using techniques from data mining, machine learning, natural language processing (NLP), information retrieval (IR), information extraction (IE) and knowledge management. As with any emerging research area, there is still no established vocabulary for KDT and TM, a fact which can lead to confusion when attempting to compare results and techniques. Often the two terms are used to denote the same thing. Knowledge discovery in text [Fel95], [Kod] (or in textual databases), Text Data Mining [Hea99] and Text Mining [Raj97], [Nah00], [Tan99], [Heh], [Sul01] are some of the terms that can be found in the literature.

We use the term KDT to indicate the overall process of turning unstructured textual data into high level information and knowledge, while the term Text Mining is used for the step of the KDT process that deals with the extraction of patterns from textual data. By extending the definition, for Knowledge discovery in databases (KDD) by Fayyad and Piatetsky-Shapiro [Fay], we give the following simple definition:

Knowledge Discovery in Text (KDT) is the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in unstructured textual data.

- *Unstructured textual data* is a set (collection) of documents. We use the term *document* to refer to a logical unit of text. This could be Web page, a status memo, an invoice, an email etc. It can be complex and long [Sul01], and is often more than text and can include graphics and multimedia content. In this paper we are only concerned with the textual elements of documents. Documents that use extensible markup language (XML), standard generalised markup language (SGML) and similar conventions are called semi-structured textual data. For our purposes, we will consider both, unstructured and semi-structured textual data.

- *Pattern*. If we consider our data as a set of facts *F* (e.g. cases in a database) a *pattern* is a rule expression E that describes facts in a subset *FE* of *F* [Fay]. Generally speaking, there are two kinds of patterns the *predictive* and the *informative***.** We use predictive patterns to predict one or more attributes in a database from the rest. This kind of patterns make an educated guess about the value of an unknown attribute given the values of other known attributes. On the other hand, informative patterns do not solve a specific problem, but they present interesting patterns that the user might not know.

KDT is a multi-step process, which includes all the tasks from the gathering of documents to the visualisation of the extracted information. The process is assumed to be non-trivial; that is the result has to be qualified as discovery. The discovered patterns should be *valid* on new textual data with some degree of certainty. The patterns are novel at least to the system and should potentially lead to some useful actions, as measured by some utility function. A main goal of KDT is to make patterns *understandable* to humans in order to facilitate a better understanding of the underlying data [Fay].

Text Mining (TM) is a step in the KDT process consisting of particular data mining and NLP algorithms that under some acceptable computational efficiency limitations produces a particular enumeration of patterns over a set of unstructured textual data.

Text Mining uses unstructured textual information and examines it in attempt to discover structure and implicit meanings "hidden" within the text [Kar00]. Text mining's main objective is the support of the knowledge discovery process in large document collections (Web or conventional, storage). Text Mining utilises specialised data mining and NLP techniques specifically operating on textual data in order to extract useful information. Text mining applications imposes strong constrains on the usual NLP techniques [Raj97]. For example as they involve large volumes of textual data, they do not allow to integrate complex treatments. It is this close connection that makes Text Mining a new research area derived from data mining and NLP.

It is best understood by comparing it with a related technology: Data mining. Data mining is the application of algorithms to a set of data to uncover previously unidentified connections and correlations. Data mining works with structured data, often numerical in nature. Text mining is analogous to data mining in that it uncovers relationships in information. Unlike data mining, it works with information stored in an unstructured collection of text documents.

Text Mining techniques are not the only tools to solve the information overload problem. Other techniques from different research areas, such as information retrieval (IR), natural language processing (NLP) etc. could also be used. Text Mining could be used directly or indirectly (similar to Web mining [Kos00]). By the direct approach we mean that the application of TM techniques directly address the problems that derive from the information overload; for example, finding relevant information on a huge collection of documents. By the indirect approach we mean that the TM techniques are used as a part of a bigger application that addresses the problems of information overload. For example, TM techniques can be used to disambiguate word sense [Ped97] for a search or information retrieval application.

The process of KDT includes three major stages [Sul01], [Kos00] (Figure 1):

1. Collect relevant documents: The first step is to identify what documents are to be retrieved. Once we have identified the source of our documents we need to retrieve the documents (from Web or form internal file systems).
2. Pre-processing documents: This step includes any kind of transformation processes of the original documents

retrieved. These transformations could aim at obtaining the desired representation of documents such as XML, SGML. The resulting documents are then processed to provide basic linguistic information about the content of each document.
With an understanding of how words, phrases, and sentences are structured, we can control text and extract information. From Text Mining prospective, our task in this pre-processing stage is to use language rules and word meanings to produce reusable by the other stages, representations of text. Main areas of linguistics (study of language) include the Morphology (structure and form of words), Syntax (How words and phrases form sentences) and Semantic (the meaning of words and statements).

3. Text mining operations: High-level information is extracted (metadata creation). Patterns and relationships are discovered within the extracted information.

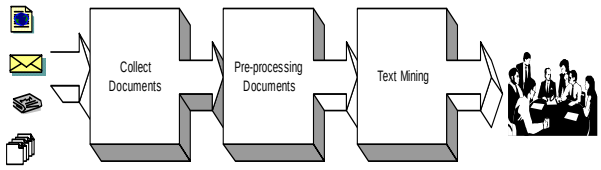


Figure 1: Major KDT stages of processing.

2.2 Main Text Mining operations

Main goal of Text mining is to unable users extract information from large textual resources. Natural language processing data mining and machine learning techniques work together to automatically discover patterns at the extracted information and the metadata the have been derived from the documents.

Most Text Mining objectives fall under the following categories of operations:

- Feature Extraction.
- Text-base navigation.
- Search and Retrieval
- Categorisation(Supervised classification)
- Clustering(Unsupervised Classification)
- Summarisation
- Temporal Analysis of Documents (i.e. Trends Analysis).
- Associations
- Visualisation

2.2.1 Feature extraction

Primary objective of the feature extraction operation is to identify facts and relations in text. Most of the times this includes distinguishing which noun phrase is a person, place, organisation or other distinct object.

Feature extraction algorithms may use dictionaries to identify some terms and linguistic patterns to detect others. For example, the name of an organisation, such as "*UMIST*", may not be in a dictionary but a feature extraction algorithm could identify it is a noun and probably significant term. Pattern recognition algorithms (like Hidden Markov Models HMM) are trained to detect patterns i.e. a noun phrase followed by a verb phrase is often followed by another noun phrase as in "*UMIST hired lecturers*". Of course always a pre- or post- processing is required to manually specify significant terms or remove unnecessary automatically identified terms.

The terms extracted usually have to be in canonical or standard form. This makes indexing retrieval and the other operations which follow more accurate. For example "*studying*" and "*study*" should be identifying as the same word. Additionally this operation should indicate the number of times each term appears in a document (word frequency). This mainly supports the classification of documents.

2.2.2 Text-base navigation.

The text-base navigation enables users to move about in a document collection by relating topics and significant terms. It helps to identify key concepts and additionally presents some of the relationships between key concepts. For example, when searching for "*UMIST*", we should be able to quickly move to "*UMIST courses*", "*UMIST degrees*" and other terms related to the "*UMIST*". The important features about this operation are two. The first is the ability to see related terms in context. For example when we notice that two terms seem to keep occurring together probably there must be an important relationship between them. The second important feature is the ability to move from one pair of terms to another. For example if the pair "*UMIST*" and "*courses*" does not satisfy us it should be possible to move to another pair such as "*UMIST*" and "*degrees*". [Wiz01b] [Tex01b]

2.2.3 Search and Retrieval

It is used for searching internal documents collections or for the Web. Its main characteristic is the various text search options. After indexing which is the first step, a wide range of text search options maybe utilised. These include basic search options such as Boolean (and/or/not), proximity, wildcard, segment, numeric range, etc. as well as some advance search capabilities, such as relevancy-ranked natural language searching, fuzzy search, concept search etc.[dtS01] [Ora01a] [Tex01]

2.2.4 Categorisation

Categorisation is the operation that we use when we want to classify documents into predefined categories. Due to this, we are able to identify the main topics of a document collection.

The categories are either pre-configured (by the programmer) or left for the user to specify. There are two ways to create the classifications. In the first approach a thesaurus can be created that defines a set of domain-specific terms and the relationships between them (the most common relationships are broader term, narrower term, synonym, and related term). The categoriser can then determine the subject of the text according to the frequency of the domain specific terms which has been identified within the text. In the second approach the categoriser is trained with sample documents.

The categoriser statistically analyse linguistic patterns (such as lexical affinities, word frequencies) from sample documents (pre-categorised) that belong in each category in order to create a statistical signature for each category. Then it uses the statistical signature to classify new documents. The advantage of the second approach is that a thesaurus, which is hard to build for large domains, is not needed.

In order to avoid wrong classification of a document it is preferable to provide multiple categories per document. Additionally it is very helpful when the tool supports both inter- and intra-document ranking. [IBM01] [Aut 01]

2.2.5 Clustering

A cluster is a group of related documents, and clustering is the operation of grouping documents on the basis of some similarity measure, automatically without having to pre-specify categories.

The most common clustering algorithms that are used are hierarchical, binary relational, and fuzzy. Hierarchical clustering creates a tree with all documents in the root node and a single document in each leaf node. The intervening nodes have several documents and become more and more specialised as they get closer to the leaf nodes. It is very useful when we are exploring a new document collection and want to get an overview of the collection. Binary relational clustering partitions the documents into a flat structure; each cluster is placed in only one set. With Fuzzy clustering all documents are included in all clusters, but with a different degree.

The most important factor in a clustering algorithm is the similarity measure. All clustering algorithms are based on similarity measures, and there are various types. One type uses words that frequently appear together (lexical affinities i.e. Department of Computation) as common features in order to group documents. Another type uses extracted features such as person name e.g. Mr. Mitsos Karaiskos. [Bern01] [Tex01b] [IBMa]

2.2.6 Summarisation

Summarisation is the operation that reduces the amount of text in a document while still keeping its key meaning.

With this operation the user usually is allowed to define a number of parameters, including the number of sentences to extract or a percentage of the total text to extract. The result includes the most significant sentences within the document. [Tex01b] [IBMa] [Aut 01]

2.2.7 Temporal Analysis of Documents

One example of such analysis is trends analysis. This operation is used for identifying trends in documents collected over a period of time. Trends can be used, for example to discover that a company is shifting interests from one domain to another. [Len]

2.2.8 Associations Analysis

Given a set of documents, identify relationships between attributes (features that have been extracted from the documents) such as the presence of one pattern implies the presence of another pattern. For example, *Neurosoft SA, Intrasoft → Take over* could be a rule that been discovered. An application based on this operation is presented by Feldman. [Fel98] [Cle01]

2.2.9 Visualisation

Visualisation utilises feature extraction and key term indexing in order to build a graphical representation of the document collection. This approach supports the user in identifying quickly the main topics or concepts by their importance on the representation. Additionally, it is easy to discover the location of specific documents in a graphical document representation. [Cle01] [Sem01]

3 Methodology of Survey

In this section, we present the feature classification scheme that we have proposed to study knowledge discovery in Text (KDT) and Text Mining tools. Then we apply this scheme to review existing tools. Although not exhaustive, we believe that the reviewed tools are representative for the current status of technology.

Features relevant to our study are grouped into 3 groups as follows: general features of tools, text mining approaches and operations supported.

3.1 General features (Table 1)

The following features are considered as general characteristics of the systems:

1. Product name and vendor, home page location on the Web.

2. Purpose and functionality.

3. Demo: Specifies if there is a demo version available. (D) Demo version available for download on the internet, (R) Demo version available on request, and (U) Unknown.

5 Architecture: The computer architecture on which the software runs. (S) Standalone, (C/S) Client/Server and (P) Parallel.

3.2 Text mining approaches (Table 2)

1. Framework Architecture: (F) File oriented, (C) Component Oriented, (D) Database oriented.

2. Text mining Methodology: (HEUR) Heuristic approach (lookup techniques)/ (STAT) Statistical approach, (NN) Neural Network, (KNOW) Knowledge-based approach.

3.3 Text mining steps and operations (Table 3)

1. (DR) Include Document Retrieval Module. Some of the products include components that support the gathering (Retrieval) of relevant documents in ordered to be processed.

2. Pre-processing: (Morph) Morphology, (Synt) Syntax, (Sem) Semantic.

3. Text mining operations: (FE) Feature Extraction, (TBN) Text-base navigation, (SR) Search and Retrieval, (CAT) Categorisation (Supervised classification), (CLU) Clustering (Unsupervised classification) (SUM) Summarisation, (TA) Trends Analysis, (ASS) Associations, (VIS) Visualisation.

Detailed description for each software product is provided in [Kar02].

4 Critical Discussion and Future Work

In this paper we provide an overview of the area of Knowledge Discovery in Text and Text Mining. We indicate some confusion regarding the use of the term text mining. Additionally we present the text mining operations and examine the most popular commercial products. In this section we will discuss our survey findings and focus on two main issues, the methodologies adopted and the framework architectures.

4.1 Text Mining Methodologies

The objective of Text Mining is to exploit information contained in textual documents in various ways, including the type of analyses that are typically performed in Data Mining: discovery of patterns and trends in data, associations among entities, predictive rules, etc. [Gro00]. Most approaches to text mining perform discovery operations either on external tags associated with each document, or on a set of words (single and multi terms) within each document. Some others apply discovery operations on phrases and facts, as extracted from the documents (IE). [Fel]

Text Mining can be performed by a collection of methods from various technological areas, not just a single one. However, all of these methods can be roughly grouped under two main headings.

The two broad groups of approaches for the development of systems that aim to extract information and knowledge from text are performance-based and knowledge-based [Sul01] (or linguistic-powered [Lex02]). In the former case designers are concerned with the effective behaviour of the system and not necessarily with the means used to obtain that behaviour. The most common performance-based algorithms are statistical methods and neural networks.

Statistical Methods: These approaches usually rely on an explicit underlying probability model. They are not context sensitive.

Neural Networks: Neural Networks are a class of systems modelled after the human brain. As the human brain consists of millions of neurons that are interconnected by synapses, neural networks are formed from large numbers of simulated neurons, connected to each other in a manner similar to brain neurons. Like in the human brain, the strength of neuron interconnections may change (or be changed by the learning algorithm) in response to a presented stimulus or an obtained output, which enables the network to learn [Goe99]. Neural networks are highly suited to textual input, being capable of identifying structure of high dimensions within a body of natural language text. Neural networks work best with data that contain noise, has a poorly understood structure and changing characteristics (all are present in textual data). [Mer96]

Knowledge-based systems or linguistic-powered systems on the other hand use explicit representations of knowledge, such as the meaning of words, relationships between facts, and rules for drawing conclusions in particular domains. Common knowledge representation schemes include inference rules, logical propositions and semantic networks such as taxonomies and ontologies. The ability to understand human language is provided by linguistics, commonly referred as Natural Language Processing (NLP). Morphology, syntactic and semantic analyses are some of the techniques being used. [Lex02]

Oracle Text is knowledge-based tool. The thesaurus provided in the tool drives thematic classification of documents. With a predefined set of terms and well defined relationships between terms, such as “stock is a narrower term for financial investment”; there is no need for the statistical analysis of a corpus (as with TextAnalyst and Intelligent Miner for Text etc.). With a taxonomy dependent analysis tool we have to create a domain specific taxonomy to get the level of document discrimination needed for the application, which is a difficult process. Taxonomies however are not the only kind of knowledge representation. Pattern matching rules are also used extensively. Heuristics are often represented as pattern matching rules. For example, “If the first two words of the sentence are in conclusion then add the sentence to the summary”. These types of rules can be represented using IF-THEN conditions and regular expressions pattern matching. Like taxonomies, which need to be domain specific, text analysis heuristic tends to be language specific. Adopting a text mining tool to additional languages, then, requires the creation of another language specific rule base.

Most of the tools combine performance-based with knowledge-based techniques in order to balance the flexibility and adaptability of statistical techniques with the domain and language specific knowledge provided by thesauri and heuristic. Choosing between statistical oriented and knowledge oriented tools depends on the domain. For rapidly changing areas such as genetics research would be facilitated by a performance based tool. On the other hand, for relatively stable areas such as finance and politics a knowledge-based tool can operate better. The ideal though would be to work with taxonomies that can automatically adjust to accommodate new terms.

4.2 Framework architecture

In this subsection we will compare how the text mining tools are integrated with each other and how well they work with third-party tools. Three kinds of architectures considered [Sul01]:

- Component-oriented architecture. For example ActiveX components.
- File-oriented architecture. Suite of independent applications. Input is provided to each tool as a stream of files or as a file containing fully qualified pathnames. The result of the analysis tools is output as streams of semi-structured text that can be easily manipulated.
- Database-oriented architecture. Processing operations are integrated into the relational database.

At one end of the component range we have TextAnalyst (Megaputer) with its ActiveX suite for dealing with text and semantic analysis. At the other end reside Oracle Text and Thunderstone's Texis, which are fully embedded into a relational database engine. In the middle and closer to Oracle Text are InFact and LexiQuest. In the middle, and closer to TextAnalyst, is Intelligent Miner for Text, ClearForest suite, etc. (Figure 2)

Tools with component-oriented architecture like TextAnalyst are appropriate for applications requiring embedded text analysis. For example in TextAnalyst the simplicity with which the ActiveX components can be integrated using common programming tools, such as Visual Basic, C++, and Python make this an appropriate choice for client and server based applications.

Intelligent Miner for Text provides a suite of independent applications. The key to integration is a standard output format that is compatible with all the programs in the suite. By this approach is easy for example to use the feature extraction tool with the clustering tool. Additionally with this semi-structured output format is easy the integration with third-party tools, and custom applications. Finally the results can easily be stored in relation database and processed by other applications.

When using Oracle Text or Texis it is already assumed that we are working with the relational database. Although we can store the documents outside the database (by keeping only the pathname or URL in the database), all processing is controlled from within the database.

When we have to analyse small amount of text then the tools with component-oriented architecture are more appropriate. For large-scale document collections, the benefits of using tools with database-oriented architecture are obvious. The documents and the related metadata can be stored and queried together. Secondly text indexing is integrated into the database, so the query optimiser can use statistics about both structured and long objects columns when formulating a query execution plan. Thirdly, SQL has been extended to support a range of text-specific selection criteria such as proximity searches, multilingual searches, thematic searches, and word variation searches.

The key criteria from the integration perspective are to determine how much of the control and the results produced from text analysis need to be integrated with other applications.

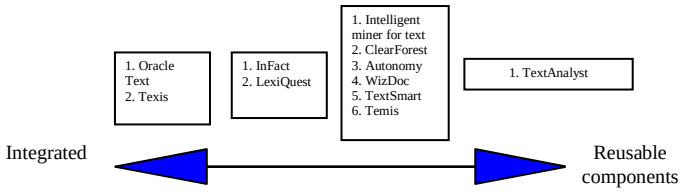


Figure 2: Framework Architectures

Oracle Text and Texis from Thunderstone provide the most extensive support for repository operations, including a wide-ranging query capability. InFact and LexiQuest stores all the information extracted in a database for use by the other subsystems. This allows a number of mining operations to be performed and great flexibility.

Intelligent miner for Text does not include a relational database engine but the output from the various text analysis tools is semi-structured that makes it easy to load into relational tables.

TextAnalyst is not geared for tight integration with databases, but it does provide its own storage mechanism. The ActiveX architecture, however, provides flexibility for programmatic control so that the lack of direct support for relational database does not prevent using the tool with a database.

4.3 Future work

KDT and TM is an emerging research area, where the need for new successful applications is of paramount importance. A promising new direction is the *integration of different techniques* in order to accomplish the text mining tasks. Currently available tools adopt a single technique or a limited set of techniques to carry out text analysis. From our research we have concluded that there is no best technique, therefore the issue is not which technique is better than the other, but which technique is suitable for the problem to be solved. In the future, though tools will have to provide a wide range of different techniques. Users should be able to choose and easily apply the techniques that are suitable for their analysis scenario.

There is a strong commercial desire to utilise data mining techniques on the masses of textual data. [Dix97] A problem with many knowledge workers is that may not know exactly what they are looking for, from the textual database; and mining is incredibly effective at retrieving interesting facts from a database. A closer and easier tie of data mining techniques with linguistic and information extraction techniques are what need to be done in the near future. Currently tools that work closer with the database (InFact, LexiQuest etc.) will have the advantage, as they facilitate efficient applications of various data mining techniques.

For a demanding, domain specific knowledge discovery task, an additional linguistic processing requirement appears;

and is necessary to perform semantic analysis to derive a sufficiently rich representation of relationships between fact and concepts described within the documents. [Sul01] Semantic analysis is computationally expensive and according to our survey few tools currently utilise it. It is a challenge for text mining tools to include an efficient and scalable semantic analysis.

An issue that may restrict the use of some of the tools we have included in our survey is multilingual text mining. Tools that do not "understand" the text (non linguistic tools) will not be able to multilingual access document collections. This is an issue to be resolved as the information that the user is looking may rely on document in various languages.

Domain knowledge and the way that can support the linguistic pre-processing steps plus the discovery tasks is an interesting topic to explore. Most of the current tools make use of taxonomies that mainly are building manually or semi-automatically. Richer taxonomies such as Ontologies to be utilised and the automatic construction seems to be the future direction.

Another identified drawback of current tools is the inability to capture the *temporal issues*. The majority of the tools do not address temporal aspects at all, while the rest do so very poorly. In the near future, tools are expected to deal with issues such as information temporal validity and trends.

There are many challenges for text mining technology but the major one, as emerged from the current investigation, will be the development of tools which will support *highly automated procedures* in order to *extract accurate and useful high level information (knowledge)* from textual resources, either the web or conventional, such as internal file systems.

References

[Aut01] Autonomy, http://www.autonomy.com/autonomy v3/ , October 2001.

[Ber01] Berners-Lee Tim, Hendler James and Lassila Ora, "*The Semantic Web*", Scientific American May 2001.

[Cle01] ClearForest, http://www.clearforest.com/index.asp, November 2001.

[Dix97] Dixon M. (1997), "*An Overview of Document Mining Technology*".

[dtS01] dtsearch , http://www.dtsearch.com/dtsoftware.html#anchor412454, November 2001.

[Fay] Fayyad U. and Piatetsky-Shapiro G., "*From Data Mining to Knowledge Discovery: An Overview*", Advances in knowledge Discovery and Data Mining, Fayyad U., Piatetsky-Shapiro G.

[Fel] Feldman R., et al., "*A domain Independent Environment for Creating Information Extraction Modules*"

[Fel95] Feldman R. & Dagan I. (1995), "*Knowledge discovery in textual databases (KDT)*". In proceedings of the first International Conference on knowledge Discovery and Data Mining (KDD-95), Montreal, Canada, August 20-21, AAAI Press, 112-117.

[Fel98] Feldman R., Fresko M., Hirsh H, et al., "*Knowledge Management: A Text Mining Approach*", in proc. of the 2nd Int. Conf. on practical Aspects of Knowledge Management (PAKM98).

[Gai97] Gaizauskas R. and Humphreys K. (1997), "*Concepticons vs. Lexicons: An Architecture for Multilingual Information Extraction*", International Summer School, SCIE-97.

[Goe99] Goebel M & Gruenwald L., "*A Survey of Data Mining and Knowledge Discovery Software tools*", ACM SIGKDD, June 1999

[Gri97] Grishman R. (1997), "*Information Extraction: Techniques and Challenges*", International Summer School, SCIE-97.

[Gro00] Grobelnik M., Mladenic D., Milic-Frayling N., "*Text Mining as Integration of Several Related Research Areas: Report on KDD'2000 Workshop on Text Mining*", Web site for KDD-2000 Workshop on Text Mining: http://www.cs.cmu.edu/~dunja/WshKDD2000.html

[Sto01] Hans-Georg Stork and Franco Mastroddi, Summary report "*Semantic Web Technologies-a New Action Line in the European Commission*", http://www.cordis.lu/ist/ka3/iaf/swt_presentations/semwebarticle.htm, Oct 2001

[Hea99] Hearst Marti A., "*Untangling Text Data Mining*", Proceedings of ACL '99: the 37th Annual Meeting of the Association for computational Linguistics, University of Maryland, June 20-26, 1999.

[Heh] Hehenberger M., Coupet P., "*Text Mining applied to patent Analysis*".

[IBMa] IBM, Text Mining Technology, "*Turning Information into Knowledge*", a White Paper from IBM.

[IBM01] IBM Intelligent Miner for Text, http://www-4.ibm.com/software/data/iminer/fortext/, September 2001

[InF02] InFact, Insightful, http://www.insightful.com/products/infact/, September 2002.

[Kar00] Karanikas H., Tjortjis C. and Theodoulidis B., "*An approach to Text Mining using Information Extraction*", PKDD 2000 Knowledge Management: Theory and Applications.

[Kar02] Karanikas H. and Theodoulidis B., "*Knowledge Discovery in Text and Text Mining Software*", Technical Report, UMIST Department of Computation, January 2002.

[Kod] Kodratoff Y., "*About Knowledge Discovery in Texts: A Definition and an Example*" Unpublished Paper.

[Kos00] Kosala R. and Blockeel H., "*Web Mining Research: A Survey*", ACM SIGKDD, July 2000.

[Len] Lent B. Agrawal R. and Srikant R., "*Discovering Trends in Text Databases*", IBM Almaden Research center.

[Lex02] LexiQuest, SPSS, http://www.spss.com/spssbi/lexiquest/, September 2002.

[Mei01] Meidan A., "*WizDoc: A concept-based search engine that retrieves relevant sections in text*". http://www.wizsoft.com/, October 2001.

[Mer00] Merrill Lynch, *e-Business Analytics*, In-depth Report, 20 November 2000

[Mer96] Merkl D. and A Min Tjoa, "*Data Mining in large free text document archives*", In Proceedings of the International Symposium on Cooperative Database Systems for Advanced Applications (CODAS'96), Kyoto, Japan, December 5-7 1996.

[Nah00] Nahm U.Y., Mooney R. J., "*Using Information Extraction to Aid the Discovery of prediction Rules from Text*".

[Ora01a] Oracle Text, "*Application Developer's Guide*", Release 9.0.1, June 2001, Part No. A90122-01

[Ora01b] Oracle Text, http://technet.oracle.com/products/text/content.html, December 2001

[Ped97] Pedersen T. and Bruce R., "*Unsupervised Text Mining*", Technical Report 97-CSE-9, Department of Computer Science and Engineering, Southern Methodist University, June 1997

[Raj97] Rajman M., Besançon R., "*Text Mining: Natural Language techniques and Text Mining applications*", Artificial Intelligence Laboratory, Computer Science Department, Swiss Federal Institute of Technology, IFIP 1997. Published by Chapman & Hall.

[Rij79] Rijsbergen C.J. van, "*Information Retrieval*", Butterworths, 1979

[Sem01] SemioMap, http://www.semio.com/, October 2001.

[Ser] Sergei Ananyan, "*Text Mining-Applications and Technologies*", Megaputer Intelligence, www.megaputer.com

[Sul01] Sullivan D., "*Document Warehousing and Text Mining*", Wiley Computer Publishing 2001.

[Tan99] Tan Ah-H., "*Text Mining: The state of the art and the challenges*", in proceedings of the Pacific Asia Conf on Knowledge Discovery and Data Mining PAKDD'99 workshop on Knowledge Discovery from Advanced Databases.

[Tex01] Texis, Thunderstone http://www.thunderstone.com/texis/site/pages, December 2001

[Tex01b] TextAnalyst, http://www.megaputer.com/products/ta/index.php3, December 2001

[Tex01c] TextSmart, http://www.spss.com/textsmart/, October 2001.

[Wac97] Wacholder N. and Ravin Y., "*Disambiguation of Proper Names in Texts*", Applied Natural Language Conference (April 3, 1997).

[Wil97] Wilks Yorick (1997), "*Information Extraction as a Core Language Technology*", International Summer School, SCIE-97

[Wiz01a] WizDoc, "*WizDoc for Web Sites*", User's Manual.

[Wiz01b] WizDoc, http://www.wizsoft.com/, October 2001.

[Wor00] Workshop - Semantic Web Technologies 22-23 November 2000, Luxembourg http://www.cordis.lu/ist/ka3/iaf/swt_presentations/swwstoc.htm , December 2001

	Product name, vendor and homepage	Purpose and functionality	Demo	Arch.
1	WizDoc, WizSoft www.wizsoft.com/	WizDoc is a search engine that enables users to search by two methods, 'string search' and 'concept-based search'.	D	S
2	TextSmart 1.0, SPSS http://www.spss.com/textsmart/	TextSmart is a tool for quantitative analysis of text-based survey responses. It basically makes cleaning, filtering and categorisation of open-ended responses.	U	S
3	TextAnalyst 2.0, Megaputer http://www.megaputer.com/products/ta/index.php3	Is a software tool for semantic analysis, navigation, and search of unstructured texts	D	S, C/S
4	Intelligent Miner for Text, IBM www-4.ibm.com/software/data/iminer/fortext/	It uses a statistical and heuristic approach to text analysis. It provides a set of text analysis tools and a search engine enhanced with mining functionality and visualisation of results.	D	S, C/S
5	ClearForest suite, ClearForest Corporation http://www.ClearForest.com	The tools structure the text data, extract important information, assign them to taxonomy and define their inter-relationships.	U	S, C/S
6	Oracle Text (interMedia Text), Oracle //technet.oracle.com/products/text/content.html	It extends the standard relational database to include operators for textual processing. The key operations of Oracle's tool are Loading, Indexing and Searching.	D	C/S
7	Autonomy, Autonomy Incorporated http://www.autonomy.com	Manages unstructured digital information, including word processing and HTML-based files, email messages and electronic news feeds. By applying concept matching techniques supports information retrieval and dynamic personalisation of digital content.	U	S, C/S
8	Temis, Text Intelligence http://www.temis-group.com/	TEMIS optimize information processing by transforming free text into data that can be analyzed for information retrieval or automatic classification of documents.	U	S, C/S
9	dtSearch, dtSearch Corp. http://www.dtSearch.com	dtSearch line of products targets on text search and retrieval. It is used for searching internal documents collections or for the Web. Its main characteristic is the different text search options.	D	S
10	Texis, Thunderstone http://www.thunderstone.com	TEXIS is a fully integrated SQL RDBMS that queries and manages databases containing natural language text, standard data types, geographic information, images, video, audio, and other payload data.	U	S, C/S
11	InFact, Insightful http://www.insightful.com/products/infact/	Insightful is merging the mining and search of structured and unstructured data, creating a search and categorisation system.	U	S, C/S
12	LexiQuest, SPSS http://www.spss.com/spssbi/lexiquest/	LexiQuest products combine natural language technology with data mining in order to facilitate main text mining operations such as categorisation, search and clustering of concepts.	U	S, C/S

Table 1: General feature of tools

1. Demo: Specifies if there is a demo version available. (D) Demo version available for download on the internet, (R) Demo version available on request, and (U) Unknown.

2. Arch.: The computer architecture on which the software runs. (S) Standalone, (C/S) Client/Server and (P) Parallel.

Product		Framework Architecture			Methodology		
		F	C	D	HEUR/STAT	NN	KNOW
1	WizDoc		X		X		X
2	TextSmart 1.0				X		
3	TextAnalyst 2.0		X			X	
4	Intelligent Miner for Text	X			X		X
5	ClearForest		X		X		X
6	Oracle Text			X			
7	Autonomy		X			X	
8	Temis	X	X		X		X
9	dtSearch		X		X		X
10	Texis			X			X
11	InFact			X	X		
12	LexiQuest			X			X

Table 2: Text Mining approaches

1. Framework Architecture: (F) File oriented, (C) Component Oriented, (D) Database oriented.
2. Text mining Methodology (Linguistic processing): (HEUR) Heuristic approach (lookup techniques)/ (STAT) Statistical approach, (NN) Neural Network, (KNOW) Knowledge-based approach.

Product			Pre-processing			Text Mining Operations								
		DR	Morph	Synt	Sem	FE	TBN	SR	CAT	CLU	SUM	TA	ASS	VIS
1	WizDoc		X	X		X	X	X						
2	TextSmart 1.0		X			X				X				
3	TextAnalyst 2.0		X				X	X		X	X			
4	Intelligent Miner for Text	X	X	X		X			X	X	X			
5	ClearForest	X	X	X		X	X	X					X	X
6	Oracle Text		X			X	X	X						
7	Autonomy	X					X		X	X	X			
8	Temis		X	X	X	X	X	X	X	X				X
9	dtsearch		X				X	X						
10	Texis		X			X	X	X						
11	InFact		X	X	X			X	X					
12	LexiQuest		X	X	X	X		X	X	X				X

Table 3: Text Mining steps and operations

(DR) Include Document Retrieval Module **- Pre-processing:** (Morph) Morphology, (Synt) Syntax, (Sem) Semantic. **- Text mining operations:** (FE) Feature Extraction, (TBN) Text-base navigation, (SR) Search and Retrieval, (CAT) Categorisation (Supervised classification), (CLU) Clustering (Unsupervised classification) (SUM) Summarisation, (TA) Temporal Analysis, (ASS) Associations, (DA) Distribution analysis, (VIS) Visualisation.

From likelihoodness between words to the finding of functional profile for ortholog genes

Natalia Grabar, Magali Sillam, Marie-Christine Jaulent
U729 Inserm/SPIM, 15, rue de l'École de médecine
75006 Paris, France
Céline Lefebvre, Édouard Henrion, Christian Néri
Inserm, Avenir Group, Laboratory of Genomic Biology, Centre Paul Broca
75014 Paris, France

Abstract

One of the main objectives of biology and genetics consists in the detection of genes responsible for diseases. This task can be reached with the revealing of ortholog genes, which are responsible for similar functions in different organisms. Since few decades, research in biology offers a huge amount of information available in scientific literature and, recenlty, in databases. In this wotk, we aim at the detection of ortholog genes by mining automatically available scientific literature and consulting databases. In this paper, we apply likelihood ratio algorithm to detect associations between gene names and terms describing their functional profile. This method allows to extract new functions for already described genes from textual corpora and to describe genes not yet encoded in databases. Such work could complete existing databases and provide additional knowledge for biologists.

Keywords : Bioinformatics, domain knowledge, corpus linguistics, text mining, information extraction

1 Introduction

One of the main objectives of biology and genetics consists in the detection of genes responsible for diseases. Research in this area, although few decades old, has generated large amount of scientific literature. Gene databases on different species (mouse, fly, worm, yeast, human, rice, etc.) are aimed to merge information into a single repository. Sequencing results and scientific literature are reviewed by human expert and recorded in these databases. But human reviewing remains difficult especially as the pace of biological research accelerates through the world. Computers and bioinformatics appear as necessary means to support data collection and to perform its analysis : gene names recognition, detection of relations between genes and proteins, merging existing databases, etc. Bioinformatics becomes even more necessary when the comparison of different species and detection of their genetic similarities are involved in the analysis. This comparison can be reached by the mean of detecting the ortholog genes (i.e. genes which present similar functions, which are responsible for similar processes, etc.). Our main contention is that the detection of such ortholog, and also of homologue, genes could benefit from automatically mining available scientific literature and databases. Different methods can be implemented to extract information about the similarities between genes.

Comparison of sequences of genes. The similarity can be detected by the comparison of sequences of genes (Pearson 98; Itoh *et al.* 05). That supposes that if genes have similar sequences, their functions and processes are potentially similar. This kind of comparison relies on formal similarity between gene sequences, as in the following case :

```
ATTGCGTTATGGAAATTCGAAACTGCCAAATACTATGTCACCA...
GATATTGCTTTATGGAAATTCGAAACTGCCAAATACTATGTCA...
```

However, on one hand such comparison gives no information about functional type of genes and on other hand similar sequences can correspond to non similar genes. To obtain more reliable results, this method can be combined with functional annotation of genes and proteins (Lord *et al.* 03; Lefebvre *et al.* 05), as those provided by `GeneOntology` annotation files. Such approach confirms that semantic similarity is correlated with sequence similarity, although disagreement in sequences and annotations appears with complex biological processes, sub-definition of processes and with mis-annotations of genes.

Interactions between genes and/or proteins. Detection of interactions between genes and/or proteins in texts (Sekimizu *et al.* 98; Yakushiji *et al.* 01; Stephens *et al.* 01; Alphonse *et al.* 04) can also point out similarity between genes. This supposes that if genes are involved in the same processes and functions or when they are at a close location in a cell they are potentially similar. When this hypothesis is applyed to textual corpora, the interactions can be detected in more (or less) constrained contexts : genes are supposed to appear in the same sentence (or document), and what is more their interaction can be outlined by function words (*interacts with*, *activates*, *are involved in*, *binds*, etc.) :
... we have demonstrated that both IMA-2 and IMB-1, the homologues of vertebrate importin alpha and beta, are essential for both spindle assembly and nuclear formation in early embryos.
For this kind of analysis, mainly linear and syntactic methods are used, but also learning algorithms (Alphonse *et al.* 04) or information retrieval approaches (Stephens *et al.* 01).

Functional profile of genes. Detection of similar genes can also be done through the construction of individual functional profile of genes and then comparing the profiles of different genes. The functional profile is related to functions of genes, processes in which they are involved, etc. One of advantages of this approach is that profile of each gene can be induced independently of other genes. Functional profile can be built from existing databases, but they describe little number of genes. To obtain more complete results, we aim at the analysis of scientific literature which we suppose to be more complete. But, this literature exists as raw texts and requires application of specific methods to be analyzed. In this paper, we present the experience of construction of functional profiles from texts using a symbolic likelihood ratio method. We then compare induced results with existing clusters obtained for the same genes (Lefebvre *et al.* 05) and with recordings proposed in existing databases.

These three approaches for detection of similarities between genes are not exclusive and can be considered as complementary.

In our work, for the detection of similarity between genes we rely on textual corpora which we suppose to contain recent research data and we apply symbolic methods for extraction of information needed for the construction of functional genes profiles. In the following sections, we first describe the material (sec. 2) and methods (sec. 3). We then present results, discuss them (sec. 4), and outline some perspectives (sec. 5).

2 Material

For the detection of similarity between genes from textual corpora, we use following types of material.

Gene names. Preferred gene names are provided by biologists : 48 genes of worm *C. elegans* and 48 of fly *Drosophila m.* (Lefebvre *et al.* 05). To make processing more complete, we have extracted aliases for these gene names from databases designed for these species : `WormBase`[1] and `FlyBase`[2]. Using the gene names and their registered aliases, we obtain 181 gene names. Among 96 studied genes, only 8 *Drosophila m.* genes are described in `FlyBase`.

Description of functional profile. To encode the functional profile of genes we use a repository of what we suppose to be the normalized description of such profile : `GeneOntology`[3] terms. `GeneOntology` provides 18,696 terms distributed into three separate hierarchies : molecular functions, biological processes and cellular components.

Textual corpora. We use textual corpora built with `PubMed`[4] abstracts. In this experience, we analyze corpus of abstracts collected with keywords relative to gene names and their aliases : *clusters-5966* corpus composed of 5,966 abstracts. Such scientific literature is supposed to

[1] `www.wormbase.org`, database on microscopic worm *C. elegans*.

[2] `flybase.bio.indiana.edu`, database on common fly *Drosphila m.*

[3] `www.geneontology.org`

[4] `www.ncbi.nlm.nih.gov/entrez/query.fcgi`, source of biomedical literature.

contain results from latest research work in this area.

Gene databases on species. Databases `WormBase` and `FlyBase` related to analyzed species, *C. elegans* and *Drosophila m.*, provide us with gene names aliases and for some genes with encoded description of their functional profile. Databases contain reliable information that has been reviewed by experts. But since this is long and complex process, databases cannot be updated as quickly as the state of the art evolves. When the information on genes is avalable, we compare acquired functional profile with the one encoded in these databases.

3 Methods

We present here experiment involving the application of likelihood ratio algorithm for the description of functional profile of genes. This algorithm indicates if the likelihood of a given result would be expected for an event compared to the likelihood that same result would be not expected for the same event. It was originally applyed to reckon associations in databases, among different fields, like in (Bodenreider *et al.* 05). Applyed to textual data, this kind of methods is usually implemented for the detection of associations between words or phrases (Manning & Schütze 99) : neighbourings, cooccurrences, collocations, etc. We assume that this algorithm will detect strong semantic relations between genes and their functions described within biological texts. But the nature of these relations is not known in advance. It can vary with the size of windows into which the associations are computed, with the nature of texts, and with post-processing and filtering of results.

In our experiment, we tune this method in the following way to adapt it for bioinformatic texts and for the detection of functional profiles of genes :

- Suitable pre-processing of corpora
- Definition of the size of windows
- Filtering raw likelihood data
- Definition of the type of relations between gene names and their profile

During the initial step of processing, we tokenize corpora in words. This has been done with respect to the conventions of gene denomination, where the use of punctuations (comma, dot, parenthesis, slash, etc.) is meaningful.

The size of windows in which cooccurrences of tokens will be computed has to fit to the type of relations we are looking for. As we want to discover semantic relations between tokens, we suppose that tokens can occur in the same sentence or in adjacent sentences. We set the size of windows to 2*20 tokens.

For the filtering of likelihood ratio output, we look for pairs :

gene name / *functional profile*

where *gene name* is one of 96 genes we analyze or one of its aliases, and *functional profile* is recognized as `GeneOntology` term. Recognition of `GeneOntology` terms can be done in two ways : exact or approximate matching. With the exact matching, `GeneOntology` term and word in text are identical, and likelihood ratio score informs us about the reliability of this association. Associated `GeneOntology` term is then an one-word term. For instance, gene *unc-54* is related to `GeneOntology` one-word term *myosin* (GO :0016459) with the score of 501.8252. As genes from *unc-54* cluster are known to be responsible for muscular development, this confirm this induced association.

Since we work currently with one-word tokens, we have not direct access to terms of `GeneOntology` composed with more that one word. Whereas the majority of terms in `GeneOntology` are composed of more that one word. The approximate matching we apply allows to bypass partly this limitation : if our method associates gene name to (all) words of a given complex `GeneOntology` term, we consider that the gene name is associated to this complex term. We can then compute the recovery, or intersection, of `GeneOntology` term and words from corpora, as well as the average association score of this recomposed term and gene name. For instance *Drosophila m.* gene *E2f* is associated with words *activity*, *repressor* and *transcriptional*. Since these words are included in the

GeneOntology term *transcriptional repressor activity* (GO :0016564) we can consider that this term is indirectly associated with gene *icd-1* with the average score of 291.17 points. What is interesting is that this term is already used to annotate *icd-1* gene in FlyBase.

And finally, for the detection of type of relations existing between gene and its associated functional profile, we use the semantics formalized within the GeneOntology hierarchies. These relations can correspond to biological processes, molecular functions or cellular components. For instance, *transcriptional repressor activity* (GO :0016564) is a molecular function of *E2f* gene, and *myosin* (GO :0016459) is cellular component of *unc-54* gene. In our approach, these three types of profiles are not exclusive. In this way, each gene can be described transversally with terms of more that one GeneOntology hierarchy.

If genes have already been described in databases, we compare their known profile with the functional profile we induce from scientific texts. Anyway, we compare induced profiles of different genes within the same cluster and contrast them with expected profiles.

4 Results and discussion

In this section, we present results of induced associations between gene names and their functional profile in the *cluster-5966* corpus. Table 1 contains results for direct and approximate matching with GeneOntology terms. Horizontal lines indicate separation of known clusters of genes. Column content is described in the following of this section.

Completeness of information in databases. First column (*Gene ID*) indicates identifiers of genes : when starting with *FBgn* string genes are those of *Drosophila m.*, remaining genes are from *C. elegans*. Second column (*DB*) indicates if gene is known as preferred gene name in databases. We can see that, in our set of 96 genes, four *Drosophila m.* genes (*ct*, *da*, *E2f* and *bic*) are described in FlyBase. For remaining genes databases contain no information. Notice that this result should be different if we consider our gene names both as preferred names and as aliases. But this position should increase the ambiguity of information : same name can correspond to different genes.

Gene name ambiguity. Third column (*Gene*) indicates preferred names for genes. When we compare 96 gene names with Specialist Lexicon from UMLS (NLM01) we obtain following list of ambiguous names (and aliases) : *bic* (*bic*, *bicaudal*), *ct*, *da* (*da*, *daughterless*), *ct* (*kf*), *sop*, *sta*, *up*, *wee*. These words can be identical with common english words (*up*, *wee*), medical words (*bicaudal*) or medical abbreviations (*ct* for *computed tomography*, *kf* for *Kenner-fecal medium*, *kidney function*, *Klippel-Feil anomaly*, *Kayser-Fleischer*, *Klenow fragment*, *filtration coefficient*, etc.). Notice that biomedical scientific literature is extremely innovative in creation of new abbreviations. The ambiguity of gene names, especially those of *Drosophila m.*, has been researched and noticed in previous work (Hirschman *et al.* 02; Tuason *et al.* 04).

Association scores. Association scores are indicated in the column *Score*. Each association pair *gene name* (column *Gene*) / *functional profile* (column *GO term*) shows score higher than 5 points. Such pairs always belong to the top 50% of associations and can be considered as reliable. We show in the table four best associations for genes. For instance, one of highest scores, 544.71 points, is read for the gene *unc-54* and cellular component *myosin*. This association is meaningful as genes from this cluster (*up*, *unc-54*, *myo-3* and *mup-2*) are known to be related to the muscular development. To facilitate the rating of scores we can apply normalization factor which will fit them into a range of 0 and 1.

Matching of GeneOntology terms. The last three columns (*GO term*, *GO ID* and *Type*) provide functional profiles of genes obtained with direct and approximate matching with GeneOntology terms. Table 1 describes 19 genes, between which four genes are already recorded in databases (*ct*, *bic*, *da* and *E2f*). In-

duced description of genes can belong to one or more GeneOntology hierarchies : biological processes (*bp*), cellular components (*cc*), and molecular function (*mf*). But this table contains only terms which are completely matched, like *transcriptional repressor activity* (GO :0016564) whose words are all associated to the gene *E2f*. If we consider also partially matched terms, with at least 50% of matched words, 2 more genes are described (*rpl-4* and *rps-2*). The total number of associated terms is 25,017, and each gene receives about 1,191 terms (from 4,853 to 6). It's obvious that ambiguous gene names obtain more important number of terms.

Correctness of matching with GeneOntology terms. Four *Drosophila m.* genes' induced profiles can be compared with information already recorded in FlyBase. We use for this *FBgn.acode* and *FBgn.summary.acode* files available on line, especially as they contain references to GeneOntology terms. In these files, *ct* is described with three terms : *specific RNA polymerase II transcription factor activity* (GO :0003704), *oogenesis (sensu Insecta)* (GO :0009993) and *calmodulin inhibitor activity* (GO :0005517). Our method associates only *specific RNA polymerase II transcription factor activity* to this gene with the score rate of 33.06 and the intersection of 0.71 (5 associated words out of 7). As for gene *E2f*, it is described with 21 GeneOntology terms. Our method induces 13 of them, for instance *cell cycle control* (GO :0000074) or *transcription factor activity* (GO :0003700), with scores of 412.83 and 487.65 respectively and complete matching. And finally, FlyBase associates *da* and *bic* genes with two and three GeneOntology terms respectively. None of them have been found with our method. This evaluation gives some indications about the database-based recall of our method. Otherwise, our method provides 4,853 terms for *ct*, 4,462 for *E2f*, 3,439 for *da*, 352 for *bic*, and more terms for other genes.

To evaluate more associations, we have analyzed homogeneity of terms within the same cluster and their conformity to the expected profiles. The first cluster is then supposed to be responsible for the *gametogenesis*, the second for *ribosome*, the third for *RNA polymerase II* and the last for *muscular development*. We will analyze first and last clusters. In the first cluster, terms associated with *ima-2*, *yps* and *wee-1.3* genes match with their expected profiles (*spindle*, *embryogenesis*, *oocyte maturation*, *male meiosis*, etc.) which obtain high scores. While relevant terms for ambiguous gene names *ct* and *wee* receive low scores. We can explain this by the fact that likelihood algorithm favors rare associations. Since the analyzed corpus is composed mainly with abstracts on biological research but also with abstracts on other fields of medicine, we suppose that *ct* and *wee* have shown unusual and solid cooccurrences in non genetic contexts. The situation is similar for the last cluster : genes *unc-54*, *myo-3* and *mup-2*, but not the ambiguous gene name *up*, are associated with relevant terms like *myosin*, *muscle myosin*, *muscle contraction*, *muscle thick filament assembly*.

Improvements to the use of GeneOntology. GeneOntology terms can be considered as normalized description for biological processes, molecular functions and cellular components of genes. But these terms don't fit a lot to the terms used in scientific literature. The approximate matching we perform allows to partially bypass this limitation. We can improve matching in following way : (1) use of the UMLS resources (NLM01), which have included and linked recently GeneOntology to terms from other biomedical terminologies ; (2) use of lexical ressources, as those provided by UMLS Specialist lexicon. Such links and resources enhance lexical anchoring of GeneOntology terms and allow to improve their recognition ; (3) use of hierarchical links to help matching of induced functional profiles and those proposed in databases.

5 Conclusion and Perspectives

The work we have presented in this paper adresses the mining of textual corpora in order to detect functional description of genes and to confirm ortholog and homologue links bet-

ween these genes. Mining scientific abstracts from PubMed shows that, in comparison with genetic databases, abstracts contain more updated and complete information about genes.

We have applyed likelihood algorithm to detect links between genes and GeneOntology terms. Direct and approximate matching allows to describe genes uknown in existing genetic databases. This description seems to satisfy expected functional profile of gene clusters and information already recorded in databases. But it has to be analyzed in a more detailed way. We obtain an interesting finding about compositionality of complex biological terms : their meaning seems to be preserved when they are recomposed.

To obtain more complete results, it would be necessary to build and analyze other corpora collected in other bibliographical sources or even with generic search engines on Internet. Full papers available on PubMed can also be used. These corpora should be processed with POS-tagger and lemmatizer like GENIA (Tsuruoka *et al.* 05), specifically trained for biological texts.

It could be also interesting to complete the list of gene aliases. We plan to do it through exploring other existing gene databases. Analysis of textual corpora can also help in this task (Séguéla & Aussenac-Gilles 99; Weissenbacher 04).

Finally we plan to combine method presented in this paper with other methods for text mining. We suppose this will allow to contrast results from different methods and to weight them. This is especially relevant for ambiguous gene names.

References

(Alphonse *et al.* 04) Erick Alphonse, Sophie Aubin, Philippe Bessières, Gilles Bisson, Thierry Hamon, Sandrine Lagarrigue, Adeline Nazarenko, Alain-Pierre Manine, Claire Nédellec, Mohamed Ould Asdel Vetah, thierry Poibeau, and Davy Weissenbacher. Extraction d'information appliquée au domaine biomédical. In *Proc of CIFT*, pages 7–20, 2004.

(Bodenreider *et al.* 05) Olivier Bodenreider, Marc Aubry, and Anita Burgun. Non-lexical approaches to identifying associative relations in the gene ontology. In *Pacific Symposium of Biocomputing*, pages 91–102, 2005.

(Hirschman *et al.* 02) L. Hirschman, AA. Morgan, and AS. Yeh. Rutabaga by any other name : extracting biological names. *Biomed*, 25 (4) :247–259, 2002.

(Itoh *et al.* 05) M Itoh, S Goto, T Akutsu, and M Kanehisa. Fast and accurate database homology search using upper bounds of local alignment scores. *Bioinformatics*, 21(7) :912–21, 2005.

(Lefebvre *et al.* 05) Céline Lefebvre, Jean-Christophe Aude, Éric Glément, and Christian Néri. Balancing protein similarity and gene co-expression reveals new links between genetic conservation and developmental diversity in invertebrates. *Bioinformatics*, 21 (8) :1550–1558, 2005.

(Lord *et al.* 03) P.W. Lord, R.D. Stevens, A. Brass, and C.A. Goble. Investigating semantic similarity measures across the Gene Ontology : the relationship between sequence and annotation. *Bioinformatics*, 19(10) :1275–1283, 2003.

(Manning & Schütze 99) C. D. Manning and H. Schütze. *Foundations of statistical natural language processing*. MIT Press, Cambridge, MA, 1999.

(NLM01) NLM (National Library of Medicine), Bethesda, Maryland. *UMLS Knowledge Sources Manual*, 2001. www.nlm.nih.gov/research/umls/.

(Pearson 98) WR Pearson. Empirical statistical estimates for sequence similarity searches. *Journal for Molecular Biology*, 276(1) :71–84, 1998.

(Séguéla & Aussenac-Gilles 99) Patrick Séguéla and Nathalie Aussenac-Gilles. Extraction de relations sémantiques entre termes et enrichissement de modèles du domaine. In *Actes d'Ingénierie des Connaissances (IC)*, pages 79–88, Palaiseau, France, juin 1999.

(Sekimizu *et al.* 98) Takeshi Sekimizu, Hyun S. Park, and Juniichi Tsujii. Identifying the intercation between genes and gene products based on frequently seen verbs in Medline abstracts. In *Proc of the Ninth WS on Genome Informatics*, 1998.

(Stephens *et al.* 01) M. Stephens, M. Palakal, S. Mukhopadhyay, R. Raje, and J. Mostafa. Detecting gene relations from medline abstracts. In *Pacific Symposium of Biocomputing*, pages 483–495, 2001.

(Tsuruoka *et al.* 05) Y Tsuruoka, Y Tateishi, J-D Kim, T Ohta, J McNaught, S Ananiadou, and J Tsujii. Developing a robust part-of-speech tagger for biomedical text. *LNCS*, 2005.

(Tuason *et al.* 04) O. Tuason, L. Chen, H. Liu, J.A. Blake, and Carol friedman. Biological comenclatures : a source of lexical knowledge and ambiguity. In *Pac Symp Biocomput*, pages 238–249, 2004.

(Weissenbacher 04) Davy Weissenbacher. La relation de synonymie en génomique. In *RECITAL*, Fès, Maroc, 2004.

(Yakushiji *et al.* 01) Akane Yakushiji, Yuka Tateisi, Yusuke Miyao, and Jun-Ichi Tsujii. Event extraction from biomedical papers using a full parser. In *Proc of Pacific Symposium of Biocomputing*, 2001.

Gene ID	DB	Gene	Score	GO term	GO ID	Type
F26B1.3	-	ima-2	47.15	localization	GO :0051179	bp
F26B1.3	-	ima-2	42.27	spindle	GO :0005819	cc
F26B1.3	-	ima-2	41.29	embryonic development	GO :0009790	bp
F26B1.3	-	ima-2	35.26	nuclear envelope	GO :0005635	cc
FBgn0004198	+	ct	79.21	protein complex	GO :0043234	cc
FBgn0004198	+	ct	75.24	protein binding	GO :0005515	mf
FBgn0004198	+	ct	71.64	protein uptake	GO :0017038	bp
FBgn0004198	+	ct	69.96	protein body	GO :0042735	cc
FBgn0011737	-	wee	146.76	antibody	GO :0003823	mf
FBgn0011737	-	wee	126.56	viral infection	GO :0016032	bp
FBgn0011737	-	wee	76.94	viral transmission	GO :0019089	bp
FBgn0011737	-	wee	56.10	viral transcription	GO :0019083	bp
FBgn0022959	-	yps	56.39	embryogenesis	GO :0009790	bp
FBgn0022959	-	yps	49.52	chromosome	GO :0005694	cc
FBgn0022959	-	yps	38.65	oocyte maturation	GO :0001556	bp
FBgn0022959	-	yps	33.05	chromosome localization	GO :0050000	bp
Y53C12A.1	-	wee-1.3	52.30	meiosis	GO :0007126	bp
Y53C12A.1	-	wee-1.3	42.77	spermatogenesis	GO :0007283	bp
Y53C12A.1	-	wee-1.3	31.31	male meiosis	GO :0007140	bp
Y53C12A.1	-	wee-1.3	20.28	embryonic development	GO :0009790	bp
B0250.1	-	rpl-2	12.34	hormone activity	GO :0005179	mf
B0250.1	-	rpl-2	6.84	growth	GO :0040007	bp
B0250.1	-	rpl-2	6.72	kinase activity	GO :0016301	mf
C56C10.8	-	icd-1	48.69	chromosome	GO :0005694	cc
C56C10.8	-	icd-1	42.67	apoptosis	GO :0006915	bp
C56C10.8	-	icd-1	28.36	chromosome segregation	GO :0007059	bp
C56C10.8	-	icd-1	18.92	malic dehydrogenase	GO :0030060	mf
FBgn0000181	+	bic	21.36	glycine receptor	GO :0016934	mf
FBgn0000181	+	bic	12.05	protein modification	GO :0006464	bp
FBgn0000181	+	bic	10.33	collagen	GO :0005581	cc
FBgn0000181	+	bic	10.61	learning	GO :0007612	bp
FBgn0003517	-	sta	230.84	guanylin	GO :0030250	mf
FBgn0003517	-	sta	228.60	secretion	GO :0046903	bp
FBgn0003517	-	sta	186.25	fluid secretion	GO :0007589	bp
FBgn0003517	-	sta	135.60	intestinal absorption	GO :0050892	bp
FBgn0004867	-	sop	113.88	sensory organ development	GO :0007423	bp
FBgn0004867	-	sop	91.26	plasmid partitioning	GO :0030541	bp
FBgn0004867	-	sop	91.75	sensory processing	GO :0050893	bp
FBgn0004867	-	sop	88.26	plasmid maintenance	GO :0006276	bp
FBgn0000413	+	da	426.41	dopamine metabolism	GO :0042417	bp
FBgn0000413	+	da	226.59	extracellular	GO :0005576	cc
FBgn0000413	+	da	223.04	dopamine receptor signaling pathway	GO :0007212	bp
FBgn0000413	+	da	137.12	neurotransmitter release	GO :0007269	bp
FBgn0011766	+	E2f	942.63	transcription	GO :0006350	bp
FBgn0011766	+	E2f	592.02	cell cycle	GO :0007049	bp
FBgn0011766	+	E2f	562.50	cyclin	GO :0016538	mf
FBgn0011766	+	E2f	557.51	cell	GO :0005623	cc
M05B5.5	-	hlh-2	57.78	chromosome	GO :0005694	cc
M05B5.5	-	hlh-2	35.86	cell fate specification	GO :0001708	bp
M05B5.5	-	hlh-2	32.22	development	GO :0007275	bp
M05B5.5	-	hlh-2	31.69	fusion cell fate specification	GO :0035156	bp
R119.6	-	taf-4	34.12	transcription	GO :0006350	bp
Y102A5C.18	-	efl-1	13.61	cell	GO :0005623	cc
Y102A5C.18	-	efl-1	13.29	localization	GO :0051179	bp
Y102A5C.18	-	efl-1	10.58	cell development	GO :0048468	bp
Y102A5C.18	-	efl-1	8.69	embryonic development	GO :0009790	bp
F11C3.3	-	unc-54	544.71	myosin	GO :0016459	cc
F11C3.3	-	unc-54	334.10	muscle myosin	GO :0005859	cc
F11C3.3	-	unc-54	213.95	myosin filament assembly	GO :0031034	bp
F11C3.3	-	unc-54	77.31	muscle thick filament assembly	GO :0030241	bp
FBgn0004169	-	up	32.31	cell	GO :0005623	cc
FBgn0004169	-	up	21.43	cell proliferation	GO :0008283	bp
FBgn0004169	-	up	20.29	biotin binding	GO :0009374	mf
FBgn0004169	-	up	20.67	cell activation	GO :0001775	bp
K12F2.1	-	myo-3	78.64	myosin	GO :0016459	cc
K12F2.1	-	myo-3	48.11	muscle myosin	GO :0005859	cc
K12F2.1	-	myo-3	13.08	actin filament	GO :0005884	cc
T22E5.5	-	mup-2	23.96	muscle contraction	GO :0006936	bp
T22E5.5	-	mup-2	21.23	morphogenesis	GO :0009653	bp
T22E5.5	-	mup-2	20.45	body morphogenesis	GO :0010171	bp
T22E5.5	-	mup-2	20.12	tail morphogenesis	GO :0035121	bp