MIMA Search: A Structuring Knowledge System towards Innovation for Engineering Education

Hideki Mima
School of Engineering
University of Tokyo
Hongo 7-3-1, Bunkyo-ku, Tokyo 113-0033, Japan
mima@t-adm.t.u-tokyo.ac.jp

Abstract

The main aim of the MIMA (Mining Information for Management and Acquisition) Search System is to achieve 'structuring knowledge' to accelerate knowledge exploitation in the domains of science and technology. This system integrates natural language processing including ontology development, information retrieval, visualization, and database technology. The 'structuring knowledge' that we define indicates 1) knowledge storage, 2) (hierarchical) classification of knowledge, 3) analysis of knowledge, 4) visualization of knowledge. We aim at integrating different types of databases (papers and patents, technologies and innovations) and knowledge domains, and simultaneously retrieving different types of knowledge. Applications for the several targets such as syllabus structuring will also be mentioned.

1 Introduction

The growing number of electronically available knowledge sources (KSs) emphasizes the importance of developing flexible and efficient tools for automatic knowledge acquisition and structuring in terms of knowledge integration. Different text and literature mining techniques have been developed recently in order to facilitate efficient discovery of knowledge contained in large textual collections. The main goal of literature mining is to retrieve knowledge that is "buried" in a text and to present the distilled knowledge to users in a concise form. Its advantage, compared to "manual" knowledge discovery, is based on the assumption that automatic methods are able to process an enormous amount of text. It is doubtful that any researcher could process such a huge amount of information, especially if the knowledge spans across domains. For these reasons, literature mining aims at helping scientists in collecting, maintaining, interpreting and curating information.

In this paper, we introduce a knowledge structuring system (KSS) we designed, in which terminology-driven knowledge acquisition (KA), knowledge retrieval (KR) and knowledge visualization (KV) are combined using automatic term recognition, automatic term clustering and terminology-based similarity calculation is explained. The system incorporates our proposed automatic term recognition / clustering and a visualization of retrieved knowledge based on the terminology, which allow users to access KSs visually though sophisticated GUIs.

2 Overview of the system

The main purpose of the knowledge structuring system is 1) accumulating knowledge in order to develop huge knowledge bases, 2) exploiting the accumulated knowledge efficiently. Our approach to structuring knowledge is based on:

- automatic term recognition (ATR)
- automatic term clustering (ATC) as an ontology[1] development
- ontology-based similarity calculation
- visualization of relationships among documents (KSs)

One of our definitions to structuring knowledge is discovery of relevance between documents (KSs) and its visualization. In order to achieve real time processing for structuring knowledge, we adopt terminology / ontology-based similarity calculation, because knowledge can also be represented as textual documents or passages (e.g. sentences, subsections) which are efficiently characterized by sets of specialized (technical) terms. Further details of our visualization scheme will be mentioned in Section 4.

[1] Although, definition of ontology is domain-specific, our definition of ontology is the collection and classification of (technical) terms to recognize their semantic relevance.

Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions, pages 21–24,
Sydney, July 2006. ©2006 Association for Computational Linguistics

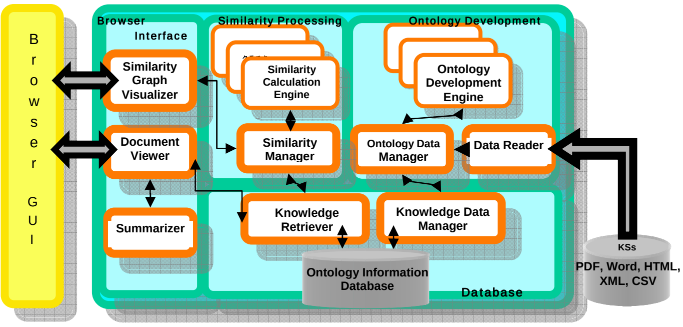

Figure 1: The system architecture

The system architecture is modular, and it integrates the following components (Figure 1):

- *Ontology Development Engine(s) (ODE)* – components that carry out the automatic ontology development which includes recognition and structuring of domain terminology;
- *Knowledge Data Manager (KDM)* – stores index of KSs and ontology in a ontology information database (OID) and provides the corresponding interface;
- *Knowledge Retriever (KR)* – retrieves KSs from TID and calculates similarities between keywords and KSs. Currently, we adopt tf*idf based similarity calculation;
- *Similarity Calculation Engine(s) (SCE)* – calculate similarities between KSs provided from KR component using ontology developed by ODE in order to show semantic similarities between each KSs. We adopt Vector Space Model (VSM) based similarity calculation and use terms as features of VSM. Semantic clusters of KSs are also provided.
- *Graph Visualizer* – visualizes knowledge structures based on graph expression in which relevance links between provided keywords and KSs, and relevance links between the KSs themselves can be shown.

3 Terminological processing as an ontology development

The lack of clear naming standards in a domain (e.g. biomedicine) makes ATR a non-trivial problem (Fukuda et al., 1998). Also, it typically gives rise to many-to-many relationships between terms and concepts. In practice, two problems stem from this fact: 1) there are terms that have multiple meanings (*term ambiguity*), and, conversely, 2) there are terms that refer to the same concept (*term variation*). Generally, term ambiguity has negative effects on IE precision, while term variation decreases IE recall. These problems show the difficulty of using simple keyword-based IE techniques. Obviously, more sophisticated techniques, identifying groups of different terms referring to the same (or similar) concept(s), and, therefore, could benefit from relying on efficient and consistent ATR/ATC and term variation management methods are required. These methods are also important for organising domain specific knowledge, as terms should not be treated isolated from other terms. They should rather be related to one another so that the relations existing between the corresponding concepts are at least partly reflected in a terminology.

3.1 Term recognition

The ATR method used in the system is based on the *C / NC-value* methods (Mima et al., 2001; Mima and Ananiadou, 2001). The *C-value* method recognizes terms by combining linguistic knowledge and statistical analysis. The method extracts multi-word terms[2] and is not limited to a specific class of concepts. It is implemented as a two-step procedure. In the first step, term candidates are extracted by using a set of linguistic filters which describe general term formation patterns. In the second step, the term candidates are assigned termhood scores (referred to as *C-values*) according to a statistical measure. The measure amalgamates four numerical corpus-based characteristics of a candidate term, namely the frequency of occurrence, the frequency of occurrence as a substring of other candidate terms, the number of candidate terms containing the given candidate term as a substring, and the number of words contained in the candidate term.

The *NC-value method* further improves the *C-value* results by taking into account the context of candidate terms. The relevant context words are extracted and assigned weights based on how frequently they appear with top-ranked term candidates extracted by the *C-value* method. Subsequently, context factors are assigned to candidate terms according to their co-occurrence with top-ranked context words. Finally, new termhood estimations, referred to as *NC-values*, are calculated as a linear combination of the *C-values* and context factors for the respective terms. Evaluation of the *C/NC-methods* (Mima and Ananiadou, 2001) has shown that contextual information improves term distribution in the extracted list by placing real terms closer to the top of the list.

[2] More than 85% of domain-specific terms are multi-word terms (Mima and Ananiadou, 2001).

3.2 Term variation management

Term variation and ambiguity are causing problems not only for ATR but for human experts as well. Several methods for term variation management have been developed. For example, the BLAST system Krauthammer et al., 2000) used approximate text string matching techniques and dictionaries to recognize spelling variations in gene and protein names. FASTR (Jacquemin, 2001) handles morphological and syntactic variations by means of meta-rules used to describe term normalization, while semantic variants are handled via WordNet.

The basic *C-value* method has been enhanced by term variation management (Mima and Ananiadou, 2001). We consider a variety of sources from which term variation problems originate. In particular, we deal with orthographical, morphological, syntactic, lexico-semantic and pragmatic phenomena. Our approach to term variation management is based on term normalization as an integral part of the ATR process. Term variants (i.e. synonymous terms) are dealt with in the initial phase of ATR when term candidates are singled out, as opposed to other approaches (e.g. FASTR handles variants subsequently by applying transformation rules to extracted terms). Each term variant is normalized (see table 1 as an example) and term variants having the same normalized form are then grouped into classes in order to link each term candidate to all of its variants. This way, a list of normalized term candidate classes, rather than a list of single terms is statistically processed. The termhood is then calculated for a whole class of term variants, not for each term variant separately.

Table 1: Automatic term normalization

Term variants		Normalised term
human cancers	}	→ human cancer
cancer in humans		
human's cancer		
human carcinoma		

3.3 Term clustering

Beside term recognition, term clustering is an indispensable component of the literature mining process. Since terminological opacity and polysemy are very common in molecular biology and biomedicine, term clustering is essential for the semantic integration of terms, the construction of domain ontologies and semantic tagging.

ATC in our system is performed using a hierarchical clustering method in which clusters are merged based on average mutual information measuring how strongly terms are related to one another (Ushioda, 1996). Terms automatically recognized by the NC-value method and their co-occurrences are used as input, and a dendrogram of terms is produced as output. Parallel symmetric processing is used for high-speed clustering. The calculated term cluster information is encoded and used for calculating semantic similarities in SCE component. More precisely, the similarity between two individual terms is determined according to their position in a dendrogram. Also a commonality measure is defined as the number of shared ancestors between two terms in the dendrogram, and a positional measure as a sum of their distances from the root. Similarity between two terms corresponds to a ratio between commonality and positional measure.

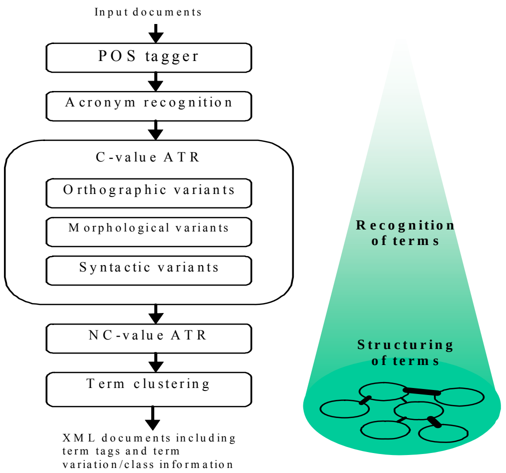

Figure 2: Ontology development

Further details of the methods and their evaluations can be referred in (Mima et al., 2001; Mima and Ananiadou, 2001).

4 Structuring knowledge

Structuring knowledge can be regarded as a broader approach to IE/KA. IE and KA in our system are implemented through the integration of ATR, ATC, and ontology-based semantic similarity calculation. Graph-based visualization for globally structuring knowledge is also provided to facilitate KR and KA from documents. Additionally, the system supports combining different databases (papers and patents, technologies and innovations) and retrieves different types of knowledge simultaneously and crossly. This feature can accelerate knowledge discovery by combining existing knowledge. For example, discovering new knowledge on industrial innovation by structuring knowledge of trendy scientific paper database and past industrial innovation report database can be expected. Figure 3 shows an example of visualization of knowledge structures in the domain of engineering. In order to structure knowledge, the system draws a graph in which nodes indicate relevant KSs to keywords given and each links between KSs indicates semantic similarities dynamically calculated using ontology information developed by our ATR / ATC components.

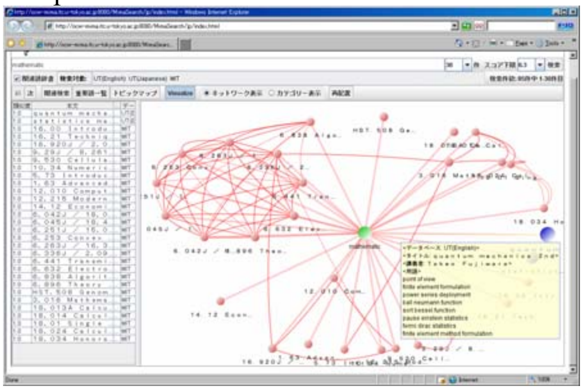

Figure 3: Visualization

5 Conclusion

In this paper, we presented a system for structuring knowledge over large KSs. The system is a terminology-based integrated KA system, in which we have integrated ATR, ATC, IR, similarity calculation, and visualization for structuring knowledge. It allows users to search and combine information from various sources. KA within the system is terminology-driven, with terminology information provided automatically. Similarity based knowledge retrieval is implemented through various semantic similarity calculations, which, in combination with hierarchical, ontology- based matching, offers powerful means for KA through visualization-based literature mining.

We have applied the system to syllabus retrieval for The University of Tokyo`s Open Course Ware (UT-OCW)[3] site and syllabus structuring (SS) site[4] for school / department of engineering at University of Tokyo, and they are both available in public over the Internet. The UT-OCW's MIMA Search system is designed to search the syllabuses of courses posted on the UT-OCW site and the Massachusetts Institute of Technology's OCW site (MIT-OCW). Also, the SS site's MIMA Search is designed to search the syllabuses of lectures from more than 1,600 lectures in school / department of engineering at University of Tokyo. Both systems show search results in terms of relations among the syllabuses as a structural graphic (figure 3). Based on the automatically extracted terms from the syllabuses and similarities calculated using those terms, MIMA Search displays the search results in a network format, using dots and lines. Namely, MIMA Search extracts the contents from the listed syllabuses, rearrange these syllabuses according to semantic relations of the contents and display the results graphically, whereas conventional search engines simply list the syllabuses that are related to the keywords. Thanks to this process, we believe users are able to search for key information and obtain results in minimal time. In graphic displays, as already mentioned, the searched syllabuses are shown in a structural graphic with dots and lines. The stronger the semantic relations of the syllabuses, the closer they are placed on the graphic. This structure will help users find a group of courses / lectures that are closely related in contents, or take courses / lectures in a logical order, for example, beginning with fundamental mathematics and going on to applied mathematics. Furthermore, because of the structural graphic display, users will be able to instinctively find the relations among syllabuses of other universities.

Currently, we obtain more than 2,000 hits per day in average from all over the world, and have provided more then 50,000 page views during last three months. On the other hand, we are in a process of system evaluation using more than 40 students to evaluate usability as a next generation information retrieval.

The other experiments we conducted also show that the system's knowledge structuring scheme is an efficient methodology to facilitate KA and new knowledge discovery in the field of genome and nano-technology (Mima et al., 2001).

References

K. Fukuda, T. Tsunoda, A. Tamura, T. Takagi, 1998. Toward information extraction: identifying protein names from biological papers, Proc. of PSB-98, Hawaii, pp. 3:705-716.

H. Mima, S. Ananiadou, G. Nenadic, 2001. ATRACT workbench: an automatic term recognition and clustering of terms, in: V. Matoušek, P. Mautner, R. Moučėk, K. Taušer (Eds.) Text, Speech and Dialogue, LNAI 2166, Springer Verlag, pp. 126-133.

H. Mima, S. Ananiadou, 2001. An application and evaluation of the C/NC-value approach for the automatic term recognition of multi-word units in Japanese, Int. J. on Terminology 6/2, pp. 175-194.

M. Krauthammer, A. Rzhetsky, P. Morozov, C. Friedman, 2000. Using BLAST for identifying gene and protein names in journal articles, in: Gene 259, pp. 245-252.

C. Jacquemin, 2001. Spotting and discovering terms through NLP, MIT Press, Cambridge MA, p. 378.

A. Ushioda, 1996. Hierarchical clustering of words, Proc. of COLING '96, Copenhagen, Denmark, pp. 1159-1162.

[3] http://ocw.u-tokyo.ac.jp/.

[4] http://ciee.t.u-tokyo.ac.jp/.