Domain-specific Concept-based Information Retrieval System

L. Shen[1], Y. K. Lim[1], H. T. Loh [2]
[1]Design Technology Institute Ltd, National University of Singapore, Singapore
[2]Department of Mechanical Engineering, National University of Singapore, Singapore

Abstract—**A domain-specific concept-based information retrieval system is introduced for the ease of retrieving organizational internal documents, i.e., technical reports, professional references and even emails. The system is composed of separate modules for the purpose of easy customization. Domain knowledge is organized using taxonomy for ease of knowledge storage and sharing. Finally the comparison between concept-based information retrieval and keywords-based retrieval shows the advantages of the system.**

Keywords—**Domain-specific, Concept-based, information retrieval**

I. Introduction and Motivation

We live in an information intensive society. The problem faced by the organizations today is not lack of information but rather, too much of it. We are caught in the midst of Information Explosion. Everyday, organizations generate huge amount of data such as e-mails, reports, journals, news, memos, etc, to support their business operations. A large portion of these data contains valuable information which is useful for making informed business decisions. However, with the vast amount of information generated at such an alarming rate, manual processing has become increasingly difficult, making it not only inefficient, but also ineffective in capturing and organizing the information. As a result, most of the data are not utilized for gaining insights to obtain competitive advantages for the company.

In this paper, we introduce a concept-based information retrieval system. It incorporates the element of concepts modeled after the application domain to enable more effective and accurate searching. Most conventional search mechanisms are primarily based on key-word search and this poses several limitations. One difficulty springs from the fact that a single word may have different meanings depending on its usage. For example, the word "lead" has different meanings in such phrases as "lead astray", "take the lead", "heavy as lead", etc. In contrast, different words or phrases may refer to the same thing. For example, the phrases - soda, pop, soft drink, cola, carbonated beverage, can all refer to the same thing. To overcome this ambiguity, we implement a concept-based framework using text mining techniques to to allow searching based on concept.

In a nutshell, text mining is the art and science of extracting information and knowledge from text. Currently, most text mining systems employ some forms of statistical methods without using any context information in the application domain. In fact, the role of domain knowledge has been given little attention. Hence, we are developing a domain specific, context based information retrieval system to provide an intelligent solution for precise information delivery. This Information Retrieval System is very similar to performing a keyword search on a search engine. However, domain knowledge is used to resolve ambiguity resulted from keyword comparison, thereby enhancing the quality of the final search result.

The outline of this paper is organized in following way. Section 2 overviews the system structure. Section 3 describes knowledge presentation in the format of ontology. The text analysis module which enhances the retrieval accuracy is introduced in section 4. Section 5 presents some snapshots of our system and the performance between keywords-based search and concept-based search is compared and discussed. Finally we present some conclusions and future work.

II. Overview of System

Fig 1 illustrates the information retrieval system. The domain knowledge here is Data Mining. The separate modules are introduced as follows.

1) *Text*: The documents are collected from various sources. Some are technical reports from different departments. Some are professional references used for research and applications, most of which are in pdf format. Useful webpages and corresponding emails are also used as inputs.

2) *Preprocessing*: At this step, the original texts are first transformed into text format. A NLP (Natural Language Processing) software – GATE [1] was applied to parse the documents, including tokenization and part-of-speech tagging. Stemming and stop-word removal were also applied to remove noise from text. Finally, the distinguished keywords/phrases are extracted to represent documents.

3) *Keywords/phrase extraction*: At this step, some heuristics are applied, including the frequency of each keyword/phrase, the different combination of part-of-speech and so on. The extracted keywords/phrases are used to present the documents. Inverted document

0-7803-8519-5/04/$20.00

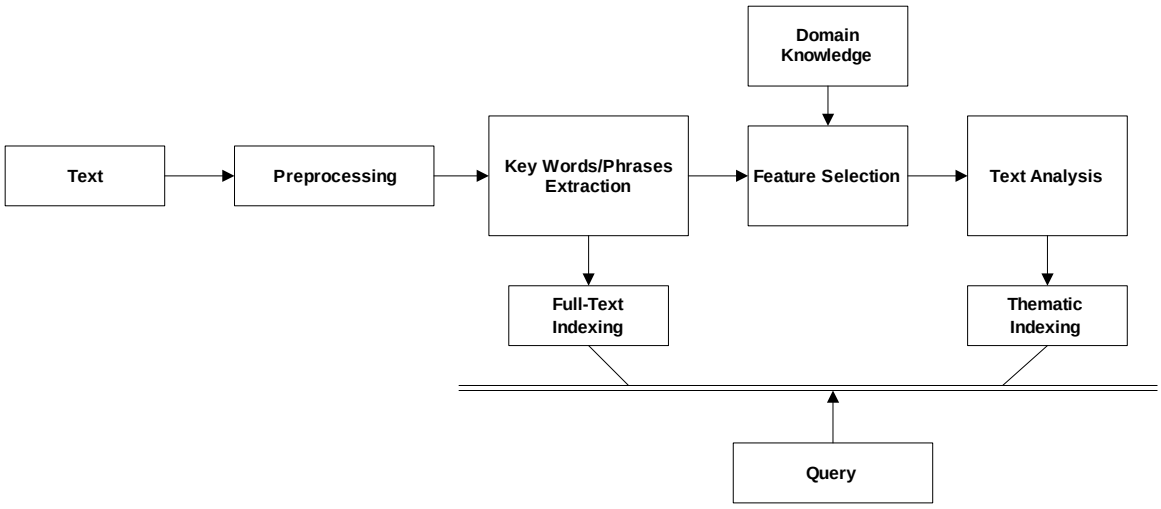

Fig. 1. Structure of concept-based information retrieval system

indexing was used to generate full-text indexing which will be used for keyword-based retrieval

4) *Feature selection:* since the system is used for concept-based retrieval, it cannot use all the words in full-text indexing. Only the representative words within concept-hierarchy are important. How to define which are differentiating and non-differentiating words are decided by domain expert. The details of domain knowledge presentation will be introduced in section 3.

5) *Text analysis*: Different text mining techniques are used to map documents into different concepts. Currently, document classification gives good performance in our system.

6) *Thematic indexing*: The documents are indexed according to concept-hierarchy.

7) *Query*: Our system can handle two kinds of queries. They are keyword-based search and concept-based search. Keyword search uses double quote (“”) to distinguish from concept search and the retrieval was processed using full-text indexing. When the users issue a concept-based query, first it will be mapped to a concept node within taxonomy. The documents associated with this concept node will be returned.

III. Domain Knowledge Representation

One of the advantages of our framework over others is the ability to incorporate the domain knowledge specific to the application environment. The main function of the domain knowledge is to resolve semantic ambiguities such as those generated by the presence of synonyms.

In a nutshell, concepts exist in the application environment are organized into ontology to provide the means for encoding background knowledge of inter-concept relationships and connections to a shared vocabulary.

An ontology is a formal explicit description of concepts, known as classes, in a domain of discourse [2] The properties of each class, which is called slots, describe its various features and attributes. The restrictions on these slots can be represented by the facets. An ontology together with a set of individual instances of classes constitutes a knowledge base.

In practical terms, developing an ontology includes:

- Defining classes in the ontology,
- Arranging the classes in a taxonomic (subclass-superclass) hierarchy,
- Defining slots and describing allowed values for these slots,
- Filling in the values for slots for instances.

We can then create a knowledge base by defining individual instances of these classes, filling in specific slot value with information as well as additional slot restriction.

We will use the task of modeling the knowledge about a Data Mining domain as an example. This model included such slots as keywords, features, kernels, parameters, and algorithms, describing the properties of an instance of the class DM_Technique. We do not attempt to build a comprehensive model of this domain but rather we use it solely as an illustration of the concepts we discuss.

The Protégé [3], an open source software, was used to build taxonomy.

Fig 2 illustrates a simple Data Mining taxonomy. The “Data Mining” is the root node. It includes two subclasses, i.e. “Data Mining Techniques (DM Techniques)” and “Data Mining Attributes (DM Attributes)”. “DM attributes” is associated with “DM Techniques”, which includes “algorithm”, “software” and “keywords”. Actually “keywords” is used to describe the highest level of one certain “DM technique”. They are differentiating keywords which are used to distinguish this technique from others. Table I gives an example of “Keywords” of

"DM technique"- rough sets. All the words are presented in their stemmed format. The words

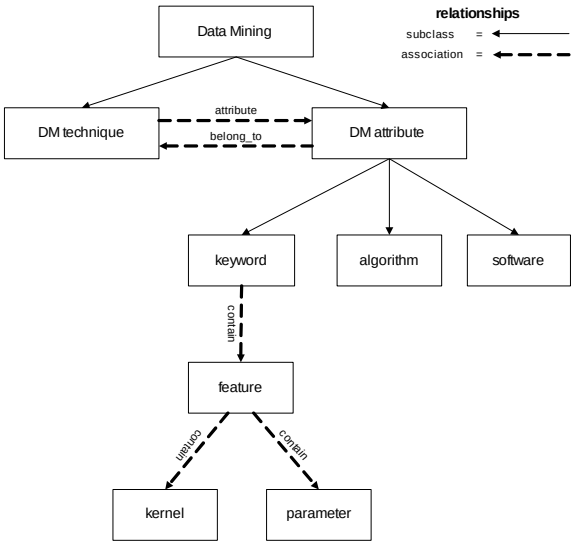

Fig. 2. Data Mining Taxonomy

within square bracket are concept-synonyms, which means they related to each other at the concept level. The documents which contain these words will be retrieved as long as one of them appeared in user's query. The words within curly bracket are keywords-synonyms. They carry the same meaning. These two kinds of synonyms are used in our prototype.

All these "keywords" cannot be assigned into its any sub-concept node. For example, for "DM technique" – "rough sets", all words of "keywords" cannot be repeated under its sub-concept, e.g. "algorithm" or "software".

TABLE I
"KEYWORDS" OF "ROUGH SETS"

Differentiating KWs	**Non-differentiating KWs**
{rough set, RST, RS} [indiscern, equival relat], [decis tabl, inform system], [accuraci of approxim, qualiti of approxim, lower approxim, upper approxim], core, [posit region, neg region]	[condit attribut, decis attribute] redund attribut, [conflict, inconsist], [decis rule, {rule gener, rule extract}]

Each concept has its own set of "keywords". These "keywords" are used to classify the documents into different concept node.

IV. TEXT ANALYSIS

Here text analysis module was used to classify the documents to a specific concept node as depicted in Fig 2. It can be seen as a hierarchical classification problem. At leaf level, for example, "rough sets algorithm", the documents are classified into concept node using the its associated "keywords". Each word of "keywords" is presented as one feature. Its frequency across the documents is used to represent document, like vector space model [4], except that vector space model uses more words than our "keywords". The Naïve Bayes Classifier [5] is used to classify the documents against the concept-node. The reason why we choose Naïve Bayes Classifier is that the output for each object is presented by its probability belonging to a certain category. It solves the document ranking problem in retrieval system.

For the middle level concept node, for example, "rough sets feature", all the "keywords" of its subconcepts – "rough sets kernel" and "rough set parameter" will be escalated and combined with the "keywords" associated with "rough sets feature". The Naïve Bayes classifier is applied to classify the updated documents representation. In this way, the whole document sets is classified according to concept hierarchy.

The classified documents are then used to build thematic indexing for convenience of retrieval.

V. COMPARISON BETWEEN KEYWORD-BASED SEARCH VS CONCEPT-BASED SEARCH

Fig3. gives a screenshot of our information retrieval system. The instruction on how to key in query is illustrated under the query bar.

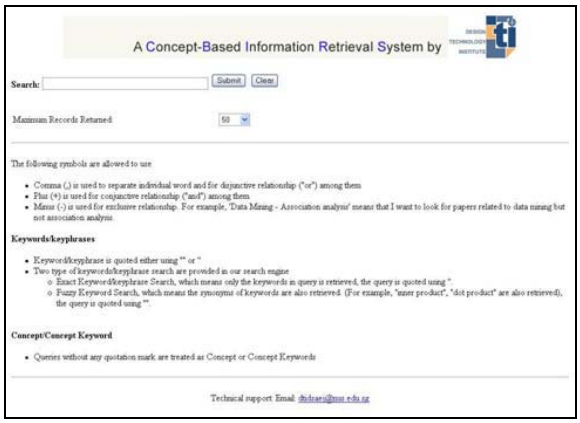

Fig. 3. User Interface of Information Retrieval System

Fig. 4 presents a screenshot for results display. If one query, e.g., "algorithm", belongs to more than one concept, a directory list will be returned. In this example, the "rough set algorithm" and "SVM algorithm" were returned. The user can click on the one in which he is interested and the relevant documents are returned as

shown in Fig. 5. The score associated with each document is displayed in descending order.

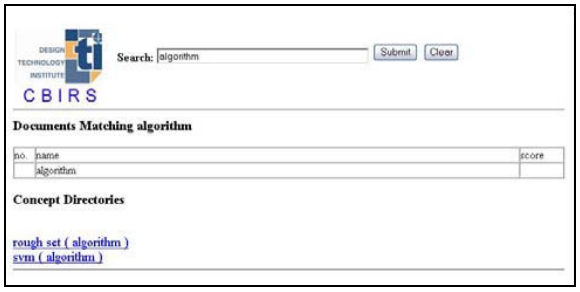

Fig. 4. User Interface for Concept Directory

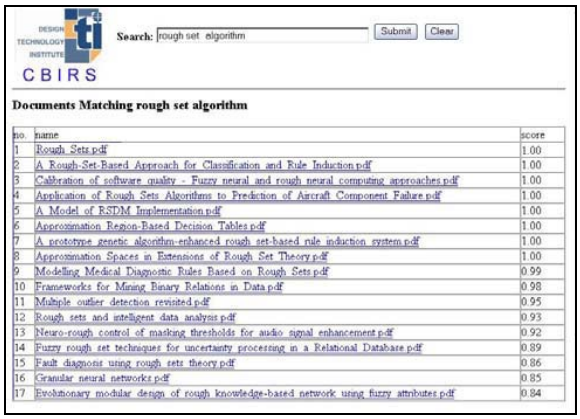

Fig. 5. User Interface for Displaying Results

To demonstrate the effectiveness of our system compared to the conventional keyword-based search, we did the following experiments. The same words are chosen and searched either as a concept or a pure keyword. The precision and recall [6] were used to evaluate the retrieval effectiveness and accuracy. Table II shows the results.

TABLE II
COMPARISON BETWEEN KEYWORDS-BASED SEARCH AND CONCEPT-BASED SEARCH

Query	Concept		Keyword	
	Precision	**Recall**	**Precision**	**Recall**
algorithm	14/26	14/25	20/109	20/25
rough set algorithm	8/16	8/15	1/1	1/15
SVM + kernel	13/15	13/15	13/18	13/15
rough set + software	9/14	9/9	0/5	0/9

By comparing the precision and recall, it can be seen that the precision of concept-based search is much better than keyword-based search, which means we can produce more relevant documents among a smaller returned document set. The recall of two kinds of search also proves that the accuracy of concept-based search is better than that of keyword-based search.

VI. CONCLUSION AND FUTURE WORKS

With the increase of e-mails, reports, journals, news, memos within organizations to support their business operations, there is the need to efficiently and effectively capture the valuable information to improve the competitiveness. Current available search engines are based on keyword search with a lot of false hits and web-related heuristics to rank the relevance. These disadvantages make them not applicable to the internal information retrieval.

Based on the analysis of requirements for current organization, we developed a domain-specific concept-based information retrieval system. The features of system include:

- Able to perform both context-based search and traditional keyword search;
- Intelligent, in that it makes use of knowledge within certain application domain;
- Web-based GUI to facilitate internal search
- Able to migrated into other domains provided that the domain knowledge is available
- Easily customized based on different requirements

By comparison between the concept-based search and keyword-based search, it can be seen that our system produce more relevant documents.

Currently, we are developing other text mining modules which are built on top of the current system. They are the document summarization module, which produces the brief description of retrieved documents, and the document clustering module, which is to be an alternative to the current text analysis module. We are also seeking the integration of web-crawler into our system so that in the future, the documents can be automatically collected and classified into the repository. It will help organizations expand and update their knowledge base easily.

REFERENCES

[1] H. Cunningham, D. Maynard, K. Bontcheva, V. Tablan, "GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications". In *Proc 40th Anniversary Meeting Association for Computational Linguistics, ACL'02*, Philadelphia, July 2002.

[2] N. F. Noy and D. L. McGuinness, "Ontology Development 101: A Guide to Creating Your First Ontology", Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, March 2001.

[3] Protégé-2000, Available : http://protege.stanford.edu/

[4] G. Salton, "Dynamic document processing", *Communications of ACM*, vol. 17, no. 7, pp. 658-668, 1972.

[5] I. Rish, "An empirical study of the Naïve Bayes Classifier", in *Proc IJCAI-01 workshop on Empirical Methods in AI*, pp. 41-46, 2001.

[6] M. Steinbach, G. Karypis and V. Kumar, "A Comparison of Document Clustering Techniques", Technical Report #00-034, Department of Computer Science and Engineering, University of Minnesota, 2000.