# Incorporating Document Keyphrases in Search Results

**Quanzhi Li, Yi-Fang Brook Wu, Razvan Stefan Bot, Xin Chen**
Information Systems Department
New Jersey Institute of Technology
{ql23, wu, rsb2, xc7}@njit.edu

**ABSTRACT**

Effectiveness and efficiency of searching and returned results presentation is the key to a search engine. Before downloading and examining the document text, users usually first judge the relevance of a return hit to the query by looking at document metadata presented in the return result. However, the metadata coming with the return hit is usually not rich enough for users to predict the content of the document. Keyphrases provide a concise summary of a document's content, offering subject metadata characterizing and summarizing document. In this paper, we propose a mechanism of enriching the metadata of the return results by incorporating automatically extracted document keyphrases in each return hit. By looking at the keyphrases in each return hit, the user can predict the content of the document more easily, quickly, and accurately. The experimental results show that our solution may save users time up to 32% and users would like to use our proposed search interface with document keyphrases as part of the metadata of a return hit.

**Keywords**

Search Interface, Search Engine, Document Keyphrase, Metadata, Document Surrogate, Search Result, Relevant Document.

## INTRODUCTION

A search engine's goal is for users to find what they need with a minimum effort. However, search engines often return a very high number of hits, and users are overwhelmed with task of finding useful information. Many researchers (Anick and Vaithyanathan, 1997; Gutwin, Paynter, Witten, Nevill-Manning and Frank, 1999; Croft, Turtle and Lewis, 1991; Smeaton and Kelledy, 1998) focus on two directions to solve this problem: either finding ways to make a query more effective by automatically or semi-automatically refining the query, or designing algorithms to effectively rank the return results to make the relevant ones appear first.

In order to reduce user's time spending on looking for the relevant documents , most search engines will display some document metadata with the return hit, such as document title, authors, snippets (short text description containing query terms) and topic category the document belongs to. These metadata will help the user predict the content of the document. If the user thinks one document may be relevant by looking at its metadata, he/she will open that document. However, document metadata are usually not rich enough for users to predict the content of the document. The title may not properly reflect the document's content; the snippet containing the query terms may not correctly represent the content of the document, either, since it is chosen to be displayed just because it contains the query terms and it can not tell whether or not the document addresses the query in a substantial way. Usually the category descriptor coming with the return hit is too broad. For example, The Association for Information System Electronic Library (AISeL) is an electronic repository for the ICIS and AMCIS conference proceedings. If a "search on title" is executed, the return hits contain only document titles and author names. If a "full-text search" is executed, only document title is displayed. This information is not rich enough for users to judge the relevance of a document to the query. Because of the lack of document metadata in the return results, users can not make relevance judgment effectively and efficiently. They usually spend much time on following links from page to page, waiting for downloads of the full-text documents, and examining the content of documents to check their relevance.

This paper describes a mechanism of enriching the metadata of the return hits by providing a set of keyphrases automatically extracted from the document for each return hit. By looking at the keyphrases, users can predict the content of the document more easily, quickly, and accurately, and they may save a lot of time spending on downloading and examining the irrelevant documents.

Previous research has shown that document keyphrases play an important role in improving efficiency and effectiveness of information retrieval (Arampatzis, Tsoris, Koster and Van, 1998; Croft et al, 1991). Keyphrases provide a concise summary of a document's content, offering semantic metadata characterizing and summarizing a document. They can be used in many

applications, such as automatic text summarization, development of search engines, document clustering, document classification, thesaurus construction, and browsing interface.

In the following sections, previous studies are reviewed first; then how keyphrases are extracted and incorporated into returned results are described in details; and finally the experiment and its results are presented.

## RELATED RESEARCH

### Applications of Keyphrases

Previous studies have shown that document keyphrases can be used in a variety of applications, such as retrieval engine (Fagan, 1989; Pedersen, 1991; Croft et al, 1991; Anick et al, 1997; Jones and Staveley, 1999; Larkey, 1999), browsing interface (Jones and Paynter, 2001; Gutwin et al, 1999), thesaurus construction (Kosovac, Vanier and Froese, 2000), and document classification and clustering (Anick et al,1997; Zamir and Etzioni, 1999; Wittem, 1999; Larkey, 1999). Some of the studies are described below.

Fagan's (1989) study shows that phrase-based automatic indexing helps to improve the precision of the overall document retrieval. Two primary conclusions are reported in his study: (1) in the experiments, the nonsyntactic phrase construction procedure did not consistently yield substantial improvements in effectiveness. (2) Many of the shortcomings of the nonsyntactic approach can be overcome by incorporating syntactic information into the phrase construction process.

Croft et al (1991) propose a method where phrases identified in natural language queries are used to build structured queries for a probabilistic retrieval model. Their experimental results show that retrieval performance can be improved by using phrases in this way, and phrases extracted automatically from a natural language query perform nearly as well as manually selected phrases.

Anick et al (1997) describe a model of context-based information retrieval. In their model, clustering and phrasal information are used together within the context of a retrieval interface. Phrases play the dual role of context descriptors and potential search terms, while cluster contexts act as a set of logical foci for query refinement and browsing. They use simple noun compound as a phrase. A noun compound is defined as any contiguous sequence of words consisting of two or more adjectives and nouns that terminate in a head noun.

Larkey (1999) develops a system for searching and classifying U.S. patent documents, based on inquiry. The system includes a "phrase help" facility, which can help users find and add phrases and terms related to those in their query. The phrases are built from historical patent text, using a set of heuristics. The text is segmented wherever items from a special list of delimiters are found. Part-of-speech tag is assigned to the terms in the resulting sequences using WordNet (Fellbaum, 1998). The sequences satisfying rules defining noun phrases and certain other criteria are retained as phrases.

Several browsing interfaces based on keyphrases extracted automatically from documents have been described in the literature. These papers either focus on system description (Liddy and Myaeng, 1993; Anick and Tiperneni, 1999; Gutwin et al, 1999; Jones and Staveley, 1999; Paynter, 2000), the quality of phrase indexing (Wacholder, Evans and Klavans, 2001) or users' experiences with a phrase browsing interface (Edgar, Nichols, Paynter, Thomson and Witten, 2003).

Gutwin *et al.* (1999) build a search engine, Keyphind, which is a mixture of searching and browsing mechanisms to help users find interesting documents. Automatically extracted keyphrases are the basic unit of both indexing and presentation, so users can interact with a document collection at the level of topics and subjects rather than words and documents. Keyphind's keyphrase index also provides a simple mechanism for refining queries, previewing results and clustering documents. They find that phrase-based indexing and presentation offer better support for browsing tasks than the traditional query engines.

Jones and Staveley (1999) develop an interactive system, Phrasier, which automatically introduces links to related material into documents as users browse and query a document collection. The links are identified using keyphrases extracted from documents and support both topic-based and inter-document navigation.

None of previous studies has explored incorporating document keyphrases in the returned results of search engines.

### Keyphrase Extraction

Turney (2000) is the first person who treats the problem of phrase extraction as supervised learning from examples. Keyphrases are extracted from candidate phrases based on examination of their features. Nine features are used by Turney to score a candidate phrase. Some of the features are: the frequency of a phrase occurring within a document, the number of words in a phrase, and whether or not the phrase is a proper noun. Turney introduces two kinds of algorithms: C4.5 decision

tree induction algorithm and GenEx. GenEx is more successful than C4.5. GenEx has two components, Extractor and Gentor. Extractor processes a document and produces a list of phrases based on the setting of 12 parameters. In the training stage, Gentor is used to tune the parameter setting to get the optimal performance. Once the training process is finished, Extractor alone can extract keyphrases using the optimal parameter setting obtained from training stage.

KIP (Wu, Li, Bot and Chen, 2004) extracts keyphrases by considering the composition of noun phrases extracted from documents. The more keywords a phrase contains and more significant these keywords are, the more likely this phrase is a keyphrase. It checks the composition of noun phrases and calculates scores for them by looking up a domain-specific database containing expert keyphrases and keywords for that domain. The candidate phrases with higher scores are extracted as this document's keyphrases.

Kea uses a machine learning algorithm which is based on naïve Bayes' decision rule (Witten, Paynter, Frank, Gutwin and Nevill-Manning, 1999). It has some pre-built models. A model is used to identify the keyphrases within a document. The model is learned from the training documents with exemplar keyphrases. A model can be used to identify keyphrases from other documents once it is learned from the training documents. Each model consists of a naive Bayes classifier and two supporting files that contain phrases frequencies and stopped words. Two attributes are used to identify keyphrases: its TF × IDF value and the distance into a document that a phrase first occurs

## INCORPRATING KEYPHRASE IN SEARCH RESULTS

There is a need for more concise document metadata which could precisely represent the document and let the user predict the document content accurately and quickly. Our proposal is to add keyphrases extracted automatically from document text to the query return hit.

In the following subsections, we describe how keyphrases are automatically extracted from a document text, what new indexes need to be added to the search engine, and how keyphrases are used in the search interface.

### Extracting Keyphrases from Document Text

Only a small portion of documents, such as academic papers, have author-provided keyphrases, and it is laborious to manually assign keyphrases to existing documents, so it is highly desirable to automate the keyphrase extraction process.

There are a few keyphrase extraction systems, such as Kea (Witten et al, 1999), Extractor (Turney, 2000) and KIP (Wu et al, 2004). In our study we use KIP, a keyphrase identification program which can automatically extract keyphrases for all the documents in a document collection.

KIP has the following main processes: tokenization, part of speech tagging, noun phrase extraction and keyphrase extraction,

#### *Tokenizer*

Tokenizer will separate all the words, punctuations and other symbols from a document to obtain the atom units.

Part-of-speech (POS) Tagger

At this stage each word is assigned an initial POS tag using WordNet lexical database (http://www.cogsci.princeton.edu/~wn/). This database contains words and the number of senses for each word used in the categories (noun, adjective, etc) it belongs to. A word's initial POS tag is determined by the category having the maximum number of senses of this word. If a word appears in more than one category it is marked as a multi-tag word. To determine a multi-tag word's correct tag, the sequence of the POS tags of its proceeding n tokens (n ranges from 2 to 4) is examined against a list of predefined syntactic rules.

#### *Noun Phrase Extractor*

After all the words are tagged, the noun phrase extractor will extract noun phrases by selecting the sequence of POS tags that are of interests. The current sequence pattern is {[A]} {N}, where A refers to Adjective, N refers to Noun, { } means repetition, and [ ] means optional. A set of exceptional rules is used, too. Noun phrases are ones satisfying the above sequence patterns or the exceptional rules.

#### *Keyphrase Extractor*

As previously described, KIP algorithm examines the composition of a noun phrase. If a noun phrase contains a keyword in it, it is more likely to be a keyphrase candidate. To use the composition of noun phrases to identify keyphrases, readily

available human identified keyphrases are parsed to form a domain-specific database. A score will be assigned to all the automatically identified noun phrases during this stage using the domain-specific database. The score S of a noun phrase is defined as the sum of weights of all the individual words and all the possible combinations of adjacent words within this noun phrase.

$$S = \sum_{i=1}^{N} w_i + \sum_{j=1}^{M} p_j$$

Where $w_i$ is the weight of a word within this phrase and $p_j$ is the weight of a sub-phrase within this phrase, including the phrase itself. KIP will access the database to obtain an individual word's weight and the weight for the combination of adjacent words, which is also a sub-phrase of the noun phrase. After the scores of all noun phrases in a document are calculated, all the noun phrases are ranked in a descending order according to their scores. The keyphrases of a document can be extracted from the ranked noun phrase list. Depending on the circumstance under which the keyphrase extraction program is used, the number of extracted keyphrases will be different. In our experiment this number is 10.

#### *Keyphrase Quality*

The quality of keyphrases will affect its effectiveness as document surrogate in search return hits. There are two basic approaches to evaluate automatically extracted keyphrases. The first one uses the standard information retrieval measures: precision and recall. The second one involves human evaluation of extracted keyphrases. According to the experiment results (Wu et al, 2004), KIP's precision was 0.44 and the recall was 0.70 when the number of extracted keyphrases is 10. The subjective evaluation of KIP also shows that it is very effective. Using a 1 to 5 scale with 1 meaning worst and 5 meaning best, the mean score for all the extracted keyphrases was 4.12 when the number of keyphrases was 10. Subjects also considered that 94% of the keyphrases extracted by KIP were *acceptable* when subjects were asked to evaluate the extracted keyphrases as "good," "bad" or "no opinion." In prior studies (Turney, 2000), *"acceptable"* means "not bad"; therefore, the percentage of "*acceptable*" is obtained by adding percentages of "good" and "no opinions." The experiments showed that KIP is very effective in generating document keyphrases.

### Indexing

To incorporate keyphrases in the return results, besides the indexes used in traditional search engines, we still need two more indexes: document-keyphrases index and keyphrase-document index. How to build and use them are described below.

#### *Document-keyphrases index*

This index contains each of the documents in the document collection and its corresponding extracted keyphrases. In our experiment, by default each document has 10 keyphrases. The index entry format looks like follows:

Document #, keyphrase1, keyphrase2, keyphrase3…

These keyphrases are ordered according to their scores calculated by KIP. This index is used to retrieve the corresponding keyphrases when a document is in a query's return list.

#### *Keyphrase-document index*

All keyphrases extracted from each of the documents are put together and sorted into one keyphrase list. The duplicates are removed. The keyphrase-document index contains each of the keyphrases on the keyphrase list and all the documents from which the keyphrase is extracted.

The index entry format looks like follows:

Keyphrase, document # x, document # y…

This index is used to retrieve all the documents containing the keyphrase when this keyphrase is clicked by the user.

### The Proposed Search Interface

Two kinds of search interfaces are introduced. The "traditional search interface" refers to the search interface used by most of today's search engines; in a traditional search interface, results do not contain document keyphrases. The "proposed search

interface" refers to the search interface which presents search hits with document keyphrases. Figure 1 shows a proposed search interface. A traditional interface is illustrated in Figure 2. The only difference between these two interfaces is that the proposed interface has keyphrases as part of the document surrogate, in addition to other document metadata. The documents shown in these two interfaces are from Drew Pearson's *Merry-Go-Round* newspaper columns in 1930s. All the documents were scanned and converted to plain text using OCR software.

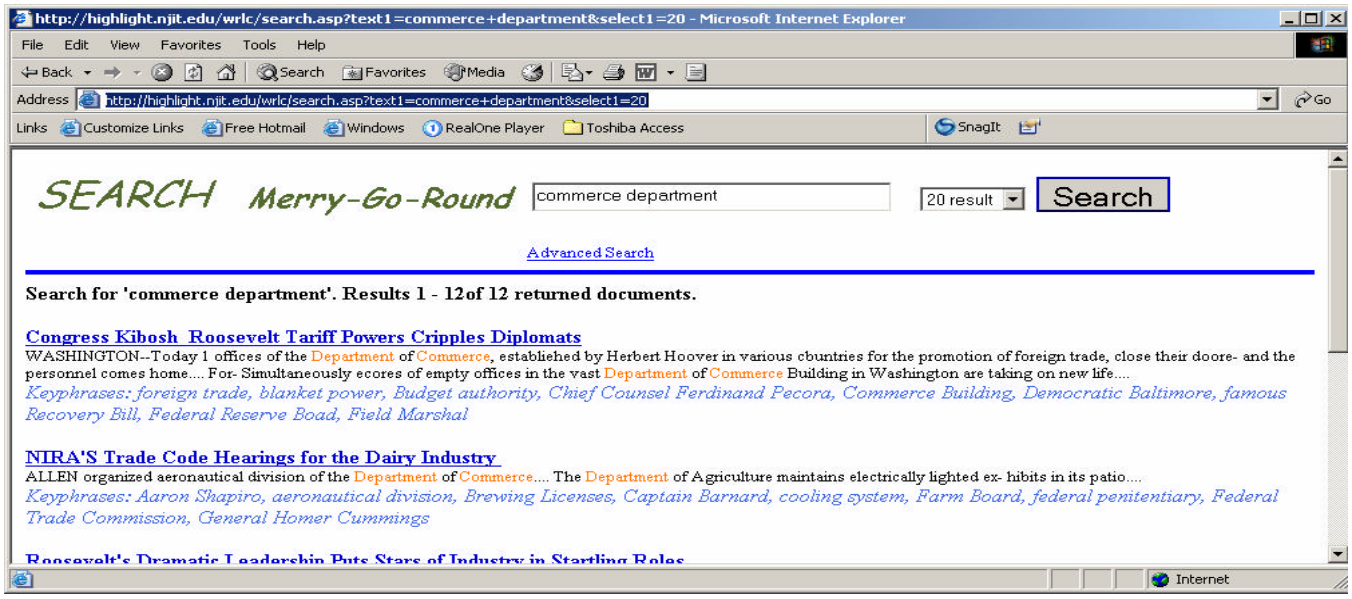


**Figure 1. The proposed search interface which provides document keyphrases**

From Figure 1 we can see that each return hit has a list of keyphrases. By looking at the keyphrases, users can predict the content of the document more precisely and quickly. Another feature of the proposed interface is that each displayed keyphrase is also a hyperlink. When the user clicks on a keyphrase all the documents containing this keyphrase will be retrieved and displayed. Actually, this feature provides a query refinement function.

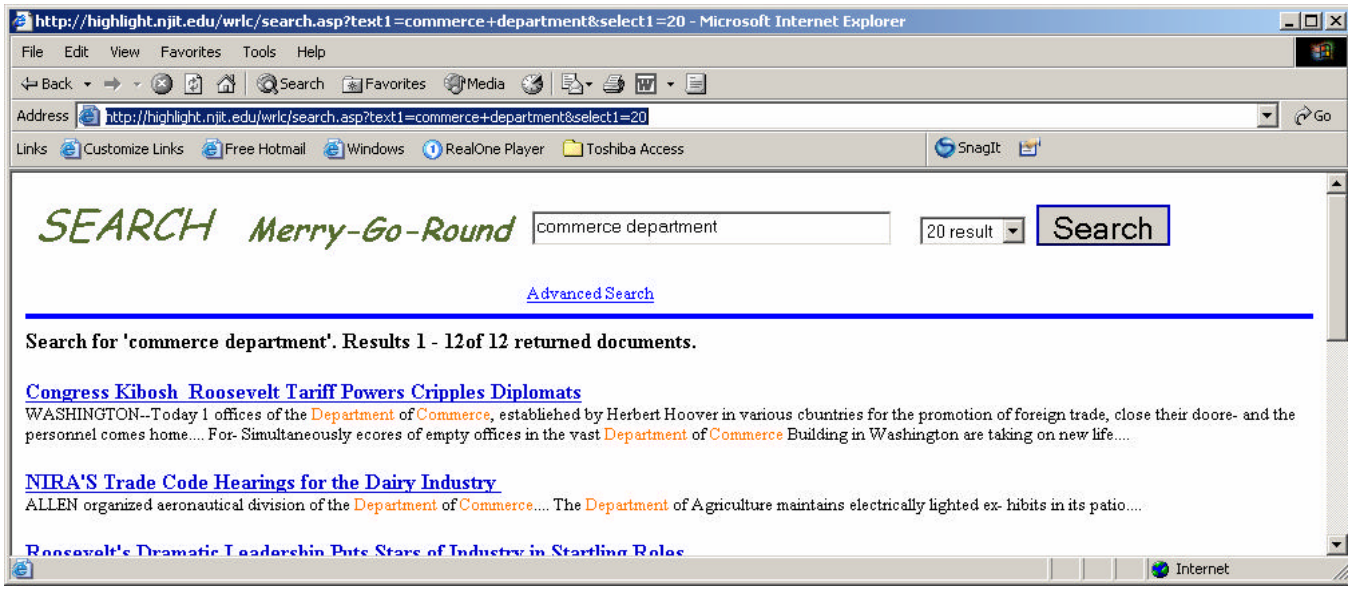


**Figure 2. The traditional search interface which does not provide document keyphrases**

## EXPERIMENT

We conducted an experiment to test the utility of using document keyphrases in the return hits of a query. We also wanted to investigate users' opinions on the proposed search interface. The two kinds of search interfaces described in previous section were compared. One is the traditional search interface, in which the return hits do not contain the document keyphrases; the other one is proposed search interface, in which the return hits contain document keyphrases.

### Method

Six subjects participated in this experiment. Each of them holds a master or PhD degree. Their backgrounds are chemistry, electrical engineering or information systems. Everyone is familiar with the traditional search interface. These six subjects were randomly divided into two groups, group A and group B. Each group had three subjects. Four natural language queries were designed and divided into two sets. Each set had two queries. Each subject would execute two queries in one query set first with one interface (traditional interface or proposed interface) and then execute the other two queries in another query set with the other interface. Subjects in group A executed queries in query set 1 with traditional interface first, and then queries in query set 2 with proposed interface; Subjects in group B executed queries in query set 2 with proposed interface first, and then queries in query set 1 with traditional interface. The order was counterbalanced. After executing each query, the subject was asked to find four documents relevant to the query and record how many documents they had downloaded and examined their contents before obtaining the four relevant documents. The queries were designed to ensure that for each query there would be at least four relevant documents in the return hits.

After executing all the queries, subjects were asked to answer a post-questionnaire, which consisted of 5 short questions. Gutwin et al (1999) develop a new search engine, Keyphind, which supports browsing with keyphrase indexes. They develop a set of post-questions and interview questions for their user study. Four (Question 1, 2, 3 and 5) out of our five questions were borrowed from theirs with little modifications. These 5 questions are shown below.

Questions asked after subjects finished the task:

1. Was it easier to carry the task with one or the other of the two search interfaces?
2. If yes, which one?
3. If yes, was the task: slightly easier, somewhat easier, or much easier?
4. Did the document keyphrases make the screen too busy?
5. Would you use a search interface like the proposed one in your work?

The document collection used in this experiment contained 1,000 documents. They were all from Drew Pearson's *Merry-Go-Round* newspaper columns in 1930s. Most of the documents were about government and politics.

### Results

The only difference between the traditional interface and the proposed interface is that in the proposed interface every return hit also includes document's keyphrases. We hypothesized that the keyphrases as part of the return hit's surrogate would help users to judge the content of result document more accurately and quickly. We measured this by comparing the number of documents the subjects needed to download and examine their contents before obtaining four relevant documents using traditional interface with that of using proposed interface. The average numbers of documents downloaded and examined for traditional interface and the proposed interface are shown in Table 1. There were totally four queries and each query was executed 6 times (one from each subject), three of which were with traditional interface and other three with proposed interface.

From Table 1, we can see that with keyphrases provided in the return hits, users could judge the content of the result documents more accurately and easily. If we assume it will cost the same amount of time to download and examine each document, we may say that the proposed interface will save users time by 32% ((8.2-5.6)/8.2 => 32%).

| | Traditional interface | Proposed interface |
|---|---|---|
| Number of documents downloaded and examined before four relevant documents were found | 8.2 | 5.6 |

**Table 1. Average number of documents examined by subjects for each query**

Subjects' responses to the post-questionnaire are shown in Table 2. All of the six subjects thought it was easier to use the proposed interface; three of them found it was much easier with proposed interface. Only one of the six subjects thought the screen was too busy with keyphrases added to the return hits. Finally, all of them would like to consider using the proposed search interface in their future search tasks. The answers are very positive and encouraging.

| Questions | Answers | | |
|---|---|---|---|
| 1. Was it easier to carry the task with one or the other of the two search interfaces? | Yes | No | |
| | 6 | 0 | |
| 2. If yes, which one? | Traditional | Proposed | |
| | 0 | 6 | |
| 3. If yes, was the task: slightly easier, somewhat easier, or much easier? | Slightly easier | Somewhat easier | Much easier |
| | 1 | 2 | 3 |
| 4. Did the document keyphrases make the screen too busy? | Yes | No | |
| | 1 | 5 | |
| 5. Would you use a search interface like the proposed one in your work? | Yes | No | |
| | 6 | 0 | |

**Table 2. Subjects' answers for post-questionnaire**

## CONCLUSION

In this paper, we present a new search interface in which the document keyphrases are incorporated into query's return hits. It provides a solution for the problem that the metadata coming with query's return hit are not rich enough for users to predict the relevance of a document to the query. Keyphrases provide a concise summary of a document's content, offering semantic metadata characterizing and summarizing document. By looking at the keyphrases in each return hit, the user can predict the content of the document more easily, quickly, and accurately. We described the proposed search interface, how to extract keyphrases for documents and how to build search indexes to implement our solution. The results of the experiment show that our solution may save users time up to 32% and users preferred our proposed search interface which provides extracted document keyphrases. Our future research will focus on how document keyphrases will affect document raking in a search engine.

## ACKNOWLEDGEMENTS

The authors would like to thank Dr. Allison Zhang for providing us the document collection for our experiment

## REFERENCES

1. Anick, P., and Tiperneni, S. (1999) The Paraphrase Search Assistant: Terminological Feedback for Iterative Information Seeking. *Proceedings of SIGIR'99: The 22nd Annual International Conference on Research and Development in Information Retrieval*, Berkeley, CA, USA, 153-159.
2. Anick, P., and Vaithyanathan, S (1997) Exploiting clustering and phrases for context-based information retrieval. *Proceedings of SIGIR'97: The 20th Annual International Conference on Research and Development in Information Retrieval*, Philadelphia, PA, USA, ACM Press, 314–322.
3. Brill, E. (1992) A simple rule-based part-of-speech tagger, *Proceeding of ANLP-92: Third conference on applied natural language processing*, Trento, Italy, 152-155.
4. Cover, T. M. and Thomas, J.A. (1991) *Elements of Information Theory*, JohnWiley.
5. Croft, B., Turtle, H., and Lewis, D. (1991) the use of phrases and structured queries in information retrieval, *Proceeding of SIGIR'91: The 14th Annual International Conference on Research and Development in Information Retrieval*, Philadelphia, ACM Press, 32-45

6. Edgar, K. D., Nichols, D. M., Paynter, G. W., Thomson, K. and Witten, I. H. (2003) A User Evaluation of Hierarchical Phrase Browsing, *Proceedings of the 7th European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2003)*, Trondheim, Norway, 313-324.

7. Fagan, J. L. (1989) the Effectiveness of a Nonsyntactic Approach to Automatic Phrase Indexing for Document Retrieval. *Journal of the American Society for Information Science*, 40(2), 115-132.

8. Fellbaum, C. (1998) *WordNet: An Electronic Lexical Database*, Cambridge: MIT Press.

9. Gutwin, C., Paynter, G., Witten, I. H., Nevill-Manning, C., and Frank, E. (1999) Improving Browsing in Digital Libraries with Keyphrase Indexes, *Journal of. Decision Support Systems*, 27(1-2), 81-104.

10. Jones, S. (1999) Design and evaluation of Phrasier, an interactive system for linking documents using keyphrases, *Proceedings of Human–Computer Interaction* (INTERACT'99), Edinburgh, UK: IOS Press, 483–490.

11. Kosovac, B. Vanier, D. J. and Froese, T. M. (2000) Use of Keyphrase Extraction Software for Creation of an AEC/FM Thesaurus, *Journal of Information technology in Construction, Vol. 5, 25-36*.

12. Larkey, L. S. (1999) A Patent Search and Classification System, Proceedings of Digital Libraries'99: The Fourth ACM Conference on Digital Libraries, Berkeley, CA: ACM Press, 179–187.

13. Liddy, E.D. and Myaeng, S.H. (1993) DR-LINK's linguistic-conceptual approach to document detection, *Proceedings of First Text Retrieval Conference (TREC-1)*. Washington, D. C., USA, 113-130.

14. Paynter, G. W., Witten, I. H, Cunningham, S. J., and Buchanan, G. (2000) Scalable Browsing for Large Collections: A Case Study., *In Proceedings of the Fifth ACM Conference on Digital Libraries*, San Antonio, TX, USA, 215-223.

15. Pedersen, J. O., Cutting, D. R., and Tukey, J. W. (1991) Snippet search: A single phrase approach *to text access, In Proceedings of the 1991 Joint Statistical Meeting,* American Statistical Association.

16. Smeaton, A. F. and Kelledy F. (1998) User-Chosen Phrases in Interactive Query Formulation for Information Retrieval, *Proceedings of the 20th BCS-IRSG Colloquium, Grenoble, France*.

17. Turney, P. D. (2000) Learning algorithm for keyphrase extraction, *Information Retrieval*, 2(4), 303-336

18. Wacholder, N., Evans, D. K., and Klavans, J. L. (2001) Automatic Identification and Organization of Index Terms for Interactive Browsing, *In Proceedings of the First ACM/IEEE-CS Joint Conference on Digital Libraries*. Roanoke, VA, USA, 126-134

19. Witten, I. H. (1999) Browsing around a digital library, *Proceeding of Australasian Computer Science Conference*, Auckland, NewZealand, 1-14

20. Witten, I. H., Paynter, G. W., Frank, E., Gutwin C., and Nevill-Manning C. G. (1999) KEA: Practical Automatic Keyphrase Extraction. *Proceedings of the Fourth ACM Conference on Digital Libraries*, 254-265.

21. Wu, B. Y., Li, Q., Bot, R. S., Chen, X. (2004) KIP: A Keyphrase Identification Program with Learning Functions. *Proceedings of International Conference on Information Technology: Coding and Computing (ITCC'04)*, Las Vegas, NV, Volume 2, 450-455.

22. Zamir, O., and Etzioni, O. (1999) Grouper: A dynamic clustering interface to Web search results. *Computer Networks and ISDN Systems*, 31(11–16), 1361–1374.