Inf Retrieval (2008) 11:229–249
DOI 10.1007/s10791-008-9044-1

Document keyphrases as subject metadata: incorporating document key concepts in search results

Yi-fang Brook Wu · Quanzhi Li

Received: 16 March 2007 / Accepted: 7 January 2008 / Published online: 24 January 2008

Abstract Most search engines display some document metadata, such as title, snippet and URL, in conjunction with the returned hits to aid users in determining documents. However, metadata is usually fragmented pieces of information that, even when combined, does not provide an overview of a returned document. In this paper, we propose a mechanism of enriching metadata of the returned results by incorporating automatically extracted document keyphrases with each returned hit. We hypothesize that keyphrases of a document can better represent the major theme in that document. Therefore, by examining the keyphrases in each returned hit, users can better predict the content of documents and the time spent on downloading and examining the irrelevant documents will be reduced substantially.

Keywords Keyphrase extraction · Document metadata · Document keyphrase · Document surrogate · Search interface

1 Introduction

The goal of search engines is to help users fulfill their information need with minimal effort. However, search systems often return a large number of hits, and it is often difficult for users to find the most useful documents in a collection. Research efforts to solving this problem include query refinement, relevance feedback, visualization of returned results, and various ranking functions.

To help users better judge the relevance of a returned document to a query, most search engines provide some document metadata, such as a document title, snippets (short text description containing query terms), and/or the subject category associated with a

Y. B. Wu (✉)
Information Systems Department, New Jersey Institute of Technology, Newark, NJ 07102, USA
e-mail: wu@njit.edu

Q. Li
Avaya Labs Research, Avaya, Inc, Basking Ridge, NJ 07920, USA
e-mail: quanzhi@avaya.com

document. However, the document metadata provided by most retrieval systems is typically not sufficiently meaningful to help users predict the content of a document.

This paper describes a mechanism of enriching the metadata of returned hits by providing a set of document keyphrases automatically extracted from documents. Document keyphrases are the most important topical phrases for a given document, in that they address the main topics of that document. The combined set of extracted key phrases from a document can provide a concise summary of the document's content, offering semantic metadata which can characterize this document. In this paper, we distinguish between the two concepts of keyword and keyphrase. A keyword is a single-term word; a keyphrase is a single-term or multi-term phrase. In other words, keywords are a sub set of key phrases—a keyword could also be a key phrase, which contains only one word. We expect that, by examining the keyphrases, users can predict the content of a document more precisely, and therefore the time spent on downloading and examining the irrelevant ones will be reduced.

Previous research has demonstrated that document keyphrases and noun phrases play an important role in improving the efficiency and effectiveness of retrieval systems (Arampatzis et al. 1998; Croft et al. 1991). They can be used in many applications, such as automatic text summarization, search engines, document clustering, document classification, thesaurus construction, and browsing interface.

In Sect. 2, we discuss prior studies of keyphrase extraction algorithms and applications of document keyphrases. Section 3 describes the keyphrase extraction algorithm used in this study. Section 4 discusses how to incorporate document keyphrases in search results. Finally, the experiments and results are presented in Sect. 5, followed by discussion and conclusion.

2 Background

In this section, we start by briefly introducing document metadata, then discussing other existing keyphrase extraction algorithms, and finally describe the applications of document keyphrases, especially in the areas of retrieval systems and browsing interfaces.

2.1 Subject metadata

Document metadata describes basic information of a document, e.g., title, author, source, and keywords. One of the most well-known metadata standards is Dublin Core (Dublin Core Metadata Initiative http://dublincore.org/), comprised of a set of 15 document properties. They are highly useful for searching and browsing collections in a digital library environment because they are document surrogates which represent some aspects of a document; incorporating them into retrieval and browsing functions within a collection can help improve effectiveness and efficiency. For example, subject metadata can help improving indexing of documents, and therefore, improving retrieval effectiveness. Subject metadata can also aid users in determining what topics are covered in a digital collection. Since they are so important to building a digital library, the Open Archives Initiatives (OAI) has developed a standard called Protocol for Metadata Harvesting (PMH, http://www.openarchives.org/pmh/) for sharing metadata. Service providers can easily harvest metadata from other collections by making an OAI-PMH service request.

Many components of metadata can be extracted quite easily, such as title, author, and publication date, while other metadata items that are not so easily determined must be manually annotated, an expensive and time consuming activity. An example of an often manually annotated metadata item is 'subject' which according to Dublin Core is: "*The topic of the resource. Typically, the topic will be represented using keywords, key phrases, or classification codes. Recommended best practice is to use a controlled vocabulary.*" (http://dublincore.org/documents/dces/) Some journals require authors to submit keywords along with their paper, and such author keywords are a good source of subject metadata. However, if none are readily available, adding subject metadata would require a large amount of effort by domain experts to manually annotate documents. Since the manual processing of subject metadata is time consuming, many research efforts have focused on enriching subject metadata for large collections automatically.

Automatic document subject metadata generation is the process of automatically extracting contextual metadata from documents. In this study, we define subject metadata as simple text constructs, i.e., 'keyphrases' thereafter, able to suggest the topic of a document from which they are extracted. Our definition is similar in concept to that of the Dublin Core's subject metadata; and the term "key phrases" is also listed as a type of subject metadata in Dublin Core. Document keyphrases are ideal to serve as metadata, because by design, they are the important topical phrases identified and extracted directly from a document body. We review applications and generation techniques of automatic keyphrases in the search and retrieval environment in the following sub-sections.

2.2 Applications of keyphrases

Previous studies have shown that document keyphrases can be used in a variety of applications, such as retrieval engines (Jones and Staveley 1999), browsing interfaces (Jones and Paynter 2002; Gutwin et al. 2003), thesaurus construction (Kosovac et al. 2000), and document classification and clustering (Witten 1999). Some studies related to retrieval systems and browsing interfaces are described below.

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 they support both topic-based and inter-document navigation. An evaluation of Phrasier's keyphrase-based retrieval algorithm proves its effectiveness to be equivalent to full-text retrieval.

Gutwin et al. (2003) built a search engine, Keyphind, consisting of a mixture of searching and browsing mechanisms to help users to find interesting documents. Automatically extracted keyphrases are the basic units for 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.

The studies described above are about the applications of keyphrases. There are also some studies about the applications of noun phrases. Because document keyphrases as described in our study are a subset of a document's noun phrases, it is very possible that keyphrases can also be used in the same applications of noun phrases and better results may be acquired. Many studies utilizing noun phrases focus on the applications of retrieval systems and browsing interfaces (Liddy and Myaeng 1993; Anick and Tiperneni 1999;

Wacholder et al. 2001; Arampatzis et al. 1998); some others explore their applicatuions on document classification and clustering (Anick and Vaithyanathan 1997; Zamir and Etzioni 1999).

Anick and Vaithyanathan (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 the simple noun compound, defined as any contiguous sequence of words consisting of two or more adjectives and nouns that terminate in a head noun, as a phrase. Their study shows that noun phrases make better units for describing cluster contents than a list of single words, and within a user interface, phrasal information associated with clusters could be transformed into interactive contexts for the purposes of iterative query refinement and structured browsing.

Previous studies have not explored incorporating document keyphrases in the returned results of search engines. Since keyphrases possess descriptive power, it might be possible for them to help users judge the relevance of a returned hit, in addition to consideration of the title and snippets. Our goal is to use keyphrases as metadata to reduce recall effort—a measure which evaluates the number of documents a user has to read before finding the desired number of relevant documents.

2.3 Keyphrase extraction

In this paper, we focus on extracting document keyphrases to be used as subject metadata. Several automatic keyphrase extraction techniques have been proposed in previous studies. Turney (2000) is the first person who treats the problem of keyphrase 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, such as the frequency of a phrase occurring within a document, 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 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 receive the optimal performance. Once the training process is finished, Extractor alone can extract keyphrases using the optimal parameter setting obtained from the training stage. The experimental results show that a custom-designed algorithm (Extractor), incorporating specialized procedural domain knowledge, can generate better document keyphrases than a general-purpose algorithm (C4.5).

Kea uses a machine-learning algorithm which is based on naïve Bayes' decision rule (Frank et al. 1999). This software package has some pre-built models with each model consisting of a naive Bayes classifier and two supporting files that contain phrases frequencies and stop words. The models are learned from the training documents with exemplar keyphrases. A model can be used to identify keyphrases from other documents once it is refined from the training documents. Experimental results on a collection of technical reports in computer science show that Kea performs comparably to Extractor, and the quality of extracted keyphrases improves significantly when domain-specific information is exploited.

Our KIP Algorithm 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 a score for each one by looking up a domain-specific glossary database containing expert keyphrases and keywords in that domain. The candidate phrases with higher scores are extracted as this document's keyphrases. More details on the KIP algorithm are introduced in Sect. 3.

3 Automatically extracting document keyphrases

Only a small portion of documents, such as academic papers, have author-provided keyphrases. Because it is costly and time-consuming to manually assign keyphrases to existing documents, it is highly desirable to automate the keyphrase extraction process. In this study, we used KIP, a keyphrase identification program, to generate document keyphrases. In Sect. 3.1 we describe the algorithm of KIP and its main components. The performance of KIP is presented in Sect. 3.2.

3.1 KIP: An automatic keyphrase extraction algorithm

In this paper, two kinds of keyphrases are mentioned. One is the pre-defined domain-specific keyphrase, which is stored in the glossary database (described later); another one is the keyphrase automatically generated for a document. The former is used to calculate the score for the latter. And as mentioned previously in this paper, we also distinguish these two concepts: keyword and keyphrase. A keyword is a single-term word; a keyphrase is a single-term or multi-term phrase.

KIP is a domain-specific keyphrase extraction program, rather than a keyphrase assignment program, which means the generated keyphrases must occur in the document (Wu et al. 2006). KIP algorithm is based on the logic that a noun phrase containing pre-defined domain-specific keywords and/or keyphrases from an authoritative source is likely to be a keyphrase in the documents which are in the same domain. The more domain-specific keywords/keyphrases a noun phrase contains and the more significant these keywords/keyphrases are, the more likely that this noun phrase is a keyphrase. A keyphrase generated by KIP can be a single-term keyphrase or a multiple-term keyphrase up to six words long. KIP operations can be summarized as follows. KIP first extracts a list of keyphrase candidates, which are noun phrases gleaned from input documents. Then it examines the composition of each candidate and assigns a score to it. The score of a noun phrase is determined mainly based on three factors: its frequency of occurrence in the document, its composition (what words and sub-phrases it contains), and how specific these words and sub-phrases are in the domain of the document. To calculate scores of noun phrases, a glossary database, which contains domain-specific keywords and keyphrases, is used. Finally, the noun phrases with higher scores are selected as keyphrases of the document.

In the following subsections, we introduce KIP's main components: the part-of-speech (POS) tagger, the noun phrase extractor, and the keyphrase extraction tool.

3.1.1 Part-of-speech tagger

To identify noun phrases, the system requires knowledge of the part of speech of the words in the text. A part-of-speech tagger is used to assign the most likely part of speech tag to each word in the text. Our part-of-speech tagger is based on the widely used Brill tagger

(Brill 1995), which uses a transformation-based error-driven learning approach. The original Brill tagger was trained on the Penn Treebank Tagged Wall Street Journal Corpus. Our tagger was trained on two corpora, the Penn Treebank Tagged Wall Street Journal Corpus and the Brown Corpus. Tagging is done in two stages. First, every word is assigned its most likely tag. Next, contextual transformations are used to improve accuracy. During the training process, a list of contextual rules is learned. These rules are used to change a word's tag if the context of the word meets one of these contextual rules. This process is called contextual transformation. For example, if the word "mark" is initially tagged as a verb, and during the contextual transformation process it is found that the word proceeding "mark" is a determiner, such as "this," then the tag of "mark" will be changed to noun.

3.1.2 Noun phrase extractor

After all the words in the document are tagged, the noun phrase extractor will extract noun phrases from this document. KIP's noun phrases extractor (NPE) extracts noun phrases by selecting the sequence of POS tags that are of interest. The current sequence pattern is defined as {[A]} {N}, where A refers to Adjective, N refers to Noun, { } means repetition, and [ ] means optional. Phrases satisfying the above sequence patterns will be extracted as noun phrases. Users may choose to obtain noun phrases of different length by changing system parameters. Ramshaw and Marcus (1995) introduce a standard data set for the evaluation of noun phrase identification approach. Based on this data set, the F value (A combination of precision and recall) of our noun phrase extractor is 0.91. It is comparable to other approaches (Sang 2000; Cardie and Pierce 1999; Argamon et al. 1999; Muñoz et al. 1999), whose F values range from 0.89 to 0.93.

3.1.3 Extracting keyphrases

The noun phrases produced by the noun phrase extractor are keyphrase candidates. They will be assigned scores and ranked in this stage. Noun phrases with higher scores will be extracted as this document's keyphrases. In order to calculate the scores for noun phrases, we use a glossary database containing domain-specific pre-defined keyphrases and keywords, which provide initial weights for the keywords and sub-phrases within a candidate keyphrase (Wu et al. 2006).

The glossary database is a key component of KIP. When the system is applied to a new domain, a glossary database appropriate for the domain is needed. To build this database, we need to find a human-developed glossary or thesaurus for the domain of interest. It could be as simple as users manually inputting keyphrases that they already know, or it could be as elaborated as those from published sources. The glossary database has two lists (tables): (a) a keyphrase list and (b) a keyword list. We use the Information Systems (IS) domain as an example to illustrate how a domain-specific glossary database is built. This IS glossary database is also used in the experiment described in Sect. 3.2. For the IS domain, both lists were generated from two main sources: (1) author keyphrases from an IS abstract corpus, and (2) "Blackwell Encyclopedic Dictionary of Management Information Systems" by Davis (1997). The reason for combining the two sources to generate the lists was the need to obtain keyphrases and keywords that would cover both theoretical and technical aspects of IS literature as much as possible. We believe that there is a positive correlation between the number of comprehensive human identified keyphrases and keywords in the glossary database and the performance of KIP.

Keyphrase list. The keyphrase list was generated as follows. First, 3,000 abstracts from IS related journals were automatically processed, and all keyphrases, included in the abstracts and provided by original authors, were extracted to form an initial list. Second, this list was further augmented with keyphrases extracted from the Blackwell encyclopedic dictionary. The final keyphrase list contains 2,722 keyphrases.

Keyword list. The keyword list was automatically generated from the keyphrase list. Most of the keyphrases in the keyphrase list are composed of two or more words. To obtain the keywords, all the keyphrases were split into individual words and added as keywords to the keyword list. The final keyword list has 2,114 keywords.

The keyphrase table has three columns (keyphrases, weights, and sources) and the keyword table has two columns (keywords and weights). Keyphrases in the keyphrase table may come from up to two sources. Initially, they are all identified by the way described above. During KIP's learning process, the system may automatically learn new phrases and add them to the keyphrase table. KIP relies on keyphrases identified by a human as positive examples as initial inputs. Sometimes, such examples are not up to date or not even available. Therefore, an adaptation and learning function is necessary for KIP, so it grows as the domain of documents advances. KIP's learning function can enrich the glossary database by automatically adding new identified keyphrases to the database. KIP's learning process and how new phrases are added to the glossary database is detailed in (Wu et al. 2006). The weights of the domain-specific keyphrases and keywords in the glossary database are assigned automatically by the following steps:

(1). Assigning weights to keywords. A keyword can be in one of three conditions: (A) the keyword itself alone is a keyphrase and is not part of any keyphrase in the keyphrase table; (B) the keyword itself alone is not a keyphrase but is part of one or more keyphrases in the keyphrase table; and (C) the keyword itself alone is a keyphrase and also is part of one or more keyphrases in the keyphrase table. Each keyword in the keyword table will be checked against the keyphrase table to see which condition it belongs to. The weights are automatically assigned to keywords differently in each condition. The rationale behind this is that it reflects how specific a keyword is in the domain. The more specific a keyword is, the higher weight it has. For each keyword in condition (A), the weight is X (the system default value for X is 10); for each keyword in condition (B), the weight is Y divided by the times the keyword appears as part of a keyphrase (the system default value for Y is 5); for each keyword in condition (C), the weight is $\frac{X+\frac{Y}{N}}{2}$, where N is the number of times that the keyword appears as part of a keyphrase. The default values of X and Y were obtained by applying 50 training documents. Their values may be changed by users when KIP is used in different domains.

(2). Assigning weights to keyphrases. The weight of each word in the keyphrase is found from the keyword table, and then all the weights of the words in this keyphrase are added together. The sum is the final weight for this keyphrase. The weights of keyphrases and keywords assigned by the above method will be used to calculate the scores of noun phrases in a document.

A noun phrase's score (normalized and ranging from 0 to 1) is defined by multiplying a factor F by a factor S. F is the frequency of this phrase in the document, and S is the sum of weights of all the individual words and all the possible combinations of adjacent words (sub-phrases) within a keyphrase candidate. The score of a noun phrase $= F \times S$.

S is defined as: $S = \sum_{i=1}^{N} w_i + \sum_{j=1}^{M} p_j$, where w_i is the weight of a word within this noun phrase, p_j is the weight of a sub-phrase within this noun phrase, and N and M are the

number of single words and number of sub-phrases within this noun phrase, respectively. For example, suppose we have a noun phrase "noun phrase extraction." The score for this noun phrase is $F \times S_{noun\_pharse\_extraction}$, where F is the frequency of phrase "noun phrase extraction" in the document, and $S_{noun\_phrase\_extraction} = W_{noun} + W_{phrase} + W_{extraction} + P_{noun\_phrase} + P_{phrase\_extraction} + P_{noun\_phrase\_extraction}$.

The motivation for including the weights of all possible sub-phrases in the phrase score, in addition to the weights of individual words, is to decide if a sub-phrase is a manual keyphrase in the glossary database. If it is, this phrase is expected to be more important. KIP will consider the keyphrase table and the keyword table to obtain the weights for words and sub-phrases. All candidate keyphrases for a document are then ranked in descending order by their scores. The top candidates are extracted from the candidate list as the keyphrases of the document. The number of keyphrases extracted can be based on an absolute number (e.g., top N), the percentage of the candidate phrases (e.g., top 20%) or a threshold of keyphrase scores (e.g., candidates with a score greater than 0.7). When two candidate phrases have overlaps, e.g., one phrase is a sub phrase of the other one, they are treated the same as other candidate phrases—both of them are placed in the rank list based on their scores.

3.2 KIP's performance

We used the standard information retrieval measures, namely precision and recall, to evaluate KIP's effectiveness. The evaluation was performed in the Information Systems (IS) domain. The document keyphrases assigned by the original author(s) are used as the standard keyphrase set. The system-generated keyphrases are compared to the keyphrases assigned by the original author(s). Recall means the proportion of the keyphrases assigned by a document's author(s) that appear in the set of keyphrases generated by the keyphrase extraction system. Precision means the proportion of the extracted keyphrases that match the keyphrases assigned by a document's author(s). Measuring precision and recall against author keyphrases is easy to carry out, and it allows comparisons between different keyphrase extraction systems. Previous studies have used this measure and found it to be an appropriate method to measure the effectiveness of a keyphrase extraction system (Jones and Paynter 2002; Turney 2000; Jones and Mahoui 2000; Tolle and Chen 2000).

We compared KIP to Kea (Frank et al. 1999) and Extractor (Turney 2000). Five hundred papers from four journals and conference proceedings were chosen as the test documents. All 500 papers had assigned keywords as provided by the author. The length of most of these papers was between 5 and 15 pages. The average number of author-assigned keyphrases for these papers was 4.7. KIP and Kea were compared when the number of extracted keyphrases was 5, 10, 15 and 20, respectively. Due to the limitation of the commercial Extractor version we could obtain (It can generate at most eight keyphrases for a documentin the trial version), Extractor and KIP were compared only when the number of extracted keyphrases was 5 and 8, respectively. This is obviously one limitation of this experiment—the performance difference between KIP and Extractor when the number of extracted keyphrases is larger than eight. Before extracting keyphrases from the test documents, Kea and Extractor were first trained using 100 training documents chosen from the same sources as the test documents (There is no overlap between the training and test documents). (Frank et al 1999) shows that 100 training documents are enough for the two algorithms to obtain good performance. Tables 1 and 2 show the results.

Table 1 Precision and recall for KIP and Kea

Number of extracted keyphrases	Average precision ± Standard deviation		Significance test on precision difference (p-value < .05 ?)	Average recall ± Standard deviation		Significance test on recall difference (p-value < .05 ?)
	KIP	Kea		KIP	Kea	
5	0.27 ± 0.19	0.20 ± 0.18	Yes	0.31 ± 0.22	0.20 ± 0.17	Yes
10	0.19 ± 0.11	0.15 ± 0.12	Yes	0.44 ± 0.24	0.32 ± 0.26	Yes
15	0.15 ± 0.07	0.13 ± 0.10	Yes	0.50 ± 0.23	0.40 ± 0.27	Yes
20	0.12 ± 0.05	0.11 ± 0.08	No	0.54 ± 0.23	0.44 ± 0.28	Yes

Table 2 Precision and recall for KIP and Extractor

Number of extracted keyphrases	Average precision ± Standard deviation		Significance test on precision difference (p-value < .05 ?)	Average recall ± Standard deviation		Significance test on recall difference (p-value < .05?)
	KIP	Extractor		KIP	Extractor	
5	0.27 ± 0.19	0.24 ± 0.15	No	0.31 ± 0.22	0.26 ± 0.16	Yes
8	0.22 ± 0.13	0.20 ± 0.12	No	0.39 ± 0.24	0.35 ± 0.22	Yes

Table 1 shows the results for KIP and Kea. We also tested the statistical significance of the difference between the precision of each system, as well as their recalls, using a paired t-test. From Table 1, we can see that, in respect to precision and recall, KIP performs better than Kea. The results are significant at 95% confidence level ($p < .05$) except for the precision when the number of extracted keyphrase is 20. Table 2 shows that KIP performs better than Extractor, but the results are only statistically significant for recall; in terms of precision, it is not significant. For each paper used in the experiment, the same number of extracted key phrases was used in the calculation of precision and recall for each system. One may believe that the significance test results should be the same for both the precision difference and recall difference between the two systems. Table 2 shows that their results were different, one was significant and the other was not. This might be explained by the following two factors: first, based on our observation, KIP performed better than Extractor especially for papers with a lower number of author assigned keyphrases. Second, when calculating recall for each paper, the number of author assigned keyphrases of that paper, instead of a pre-defined number (e.g., 5 or 8 for precision calculation) for all papers, was used as the denominator. Therefore, for papers with a lower number of author assigned keyphrases, the recall difference between the two systems was bigger than the precision difference, which causes the overall recall difference between the two systems was statistically significant.

4 Incorprating keyphrase in search results

In Sect. 1, we have addressed the need for richer document metadata which could more precisely represent the document, and let the user predict the document content more accurately. To enrich the metadata of returned hits, our proposal is to incorporate document keyphrases in the query's returned hits.

In Sect. 4.1, we illustrate our proposed search interface which incorporates keyphrase in search results. In Sect. 4.2, we discuss what new indexes are needed in order to implement the proposed search interface.

4.1 The proposed search interface

We now introduce two kinds of retrieval interfaces: the traditional search interface and our proposed search interface. The traditional interface refers to the linear listing type of search interface used by most search engines which usually provides at least a title and a snippet as metadata for each hit (Fig. 1). The proposed interface refers to the search interface which presents search hits with document keyphrases as part of their metadata (Fig. 2), in addition to a title and a snippet. The only difference between these two interfaces is that the proposed one has keyphrases as part of the document surrogate, in addition to other document metadata.

From Fig. 2, we can see that each returned hit has a list of keyphrases. By looking at the keyphrases, users should be able to predict the content of a document more precisely. In addition, another feature of the proposed interface is that each displayed keyphrase is also a hyperlink. When users click on a keyphrase, all the documents containing this keyphrase will be retrieved and displayed. This feature provides a query refinement and browsing function.

4.2 Keyphrase-related indexing

To implement the proposed interface illustrated in Fig. 2, in addition to the indexes used by traditional search engines, the system needs two more indexes: the document-keyphrases index and keyphrase-document index. The document-keyphrase index is used to retrieve

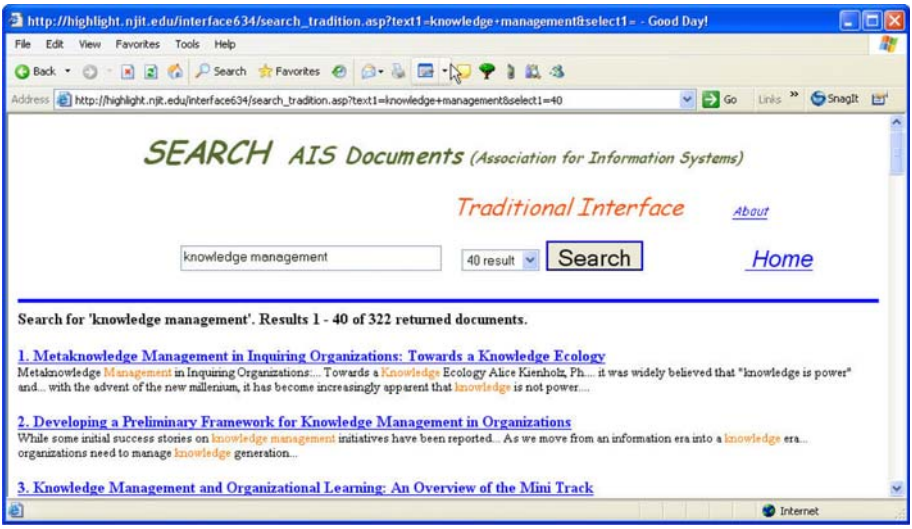

Fig. 1 The traditional search interface which does not provide document keyphrases

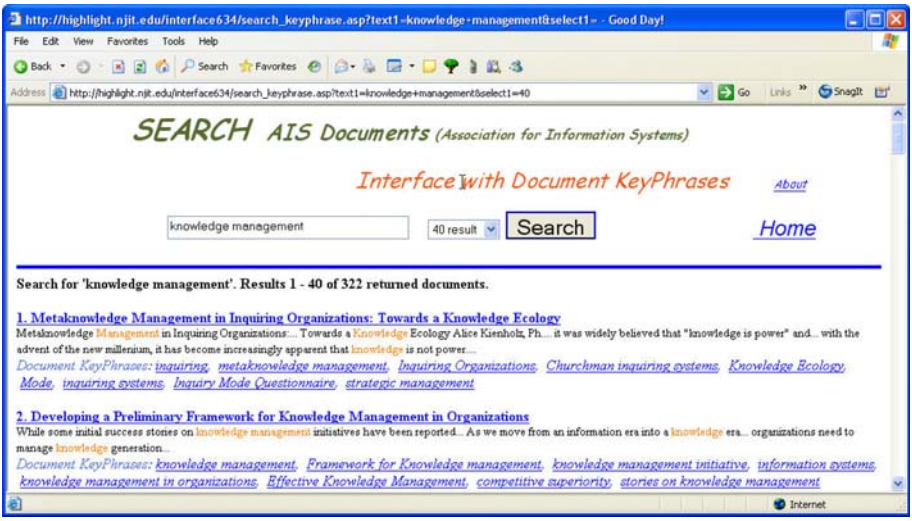

Fig. 2 The proposed search interface which provides document keyphrases

the corresponding keyphrases when a document is on a query's return list. This index contains each document identifier in the document collection and its corresponding keyphrases extracted automatically from that document text. In our experiment, by default, 10 keyphrases were produced for each document. The keyphrases are listed in a descending order according to their importance to the document.

After keyphrases are extracted for all the documents, all the keyphrases are put together and sorted into one keyphrase list. The duplicates are then removed. The remaining phrases on the list are used to form a keyphrase-document index. Keyphrase-document index contains each of the keyphrases on the keyphrase list and all the documents from which the keyphrase is extracted. In the proposed search interface, when a document keyphrase is clicked by users, the keyphrase-document index is used to retrieve all documents containing that keyphrase.

5 Evaluation of incorporating document keyphrases in search results

In this section, we evaluate the mechanism of incorporating document keyphrases into search results. We hypothesized that document keyphrases as part of a returned hit's metadata would help users judge the content of a returned document more accurately, and, therefore, recall effort will be reduced. Recall effort was measured by comparing the number of documents opened by the subjects using the traditional interface, in order to obtain a desired number of relevant documents, to the number of documents opened by the subjects using the proposed interface. We hypothesized that the number of documents opened by the subjects with the proposed interface would be less than the number of documents opened by the subjects using the traditional interface, which means, with the proposed interface, less time would be spent on downloading and examining the irrelevant documents.

The two kinds of search interfaces described in previous section were used in our experiment. This experiment was conducted with two different document collections.

The first one was the electronic library of Association for Information Systems (AIS), an electronic repository of papers published by AIS. Most of the documents in this collection are related to Information Systems. The second collection was related to the domain of politics, government, and economy. All of the documents in the second collection were from Drew Pearson's *Merry-Go-Round* newspaper columns. For this experiment, we recruited fifty-four subjects. Forty-two of them participated in the experiment with AIS collection, and 12 others participated in the experiment with *Merry-Go-Round* collection. In the following subsections, we describe our experimental procedures and results.

5.1 Experiment on the AIS collection

This prototype system has two kinds of search interfaces: traditional and the proposed. They are shown in Figs. 1 and 2. The only difference between these two interfaces is that the proposed one has a set of keyphrases for each returned hit, and the traditional does not. There were 6,965 documents in this collection. The length of most of these papers was between 4 and 17 pages. Only a small portion of the documents had author-assigned keyphrases. For this experiment, the keyphrases for each document in the AIS collection were generated by KIP, using the glossary database described in Sect. 3.1.3, which contained 2,722 pre-defined keyphrases in IS domain. We tested our hypothesis using a measure similar to "recall effort." Recall effort is the ratio of the number of relevant documents desired to the number of documents examined by the user to find the number of relevant documents desired. In this experiment, we asked the subjects to find four relevant documents for each query, so the number of relevant documents desired was same for each subject and each query.

Forty-two subjects participated in the experiment using the AIS collection. All subjects were graduate students majoring in computer science or information systems and all were familiar with common search systems. They were randomly divided into two groups, group A and group B, with each group containing 21 subjects. Four queries were designed for AIS collection. The queries were designed by the authors of this paper to ensure that each query would have at least four relevant documents in the returned list and be clear about the information request it tried to represent. The queries used in this experiment and the ones using the *Merry-Go-Round* collection are displayed in the appendix. The four queries were divided into two sets. Subjects in group A executed queries in query set 1 with the traditional interface first, then executed queries in query set 2 with the proposed interface; Subjects in group B executed queries in query set 1 with the proposed interface first, and then executed queries in query set 2 with the traditional interface. The subjects were asked to find four documents relevant to the query and record how many documents they had opened in order to obtain the four relevant documents. Due to the error-prone nature of asking is the subjects to record the number of opened documents by themselves, another method would have to be found. Obtaining this number through user logs would make the data more reliable. Unfortunately, this information was not recorded in the user logs in this experiment. The subjects were told that "a relevant document" means "the main theme of the document is about the topic of the query." The subjects entered the queries directly into the query box and decided which returned document was relevant based on their own judgment. We used a post-questionnaire to investigate users' opinions about the proposed search interface. The subjects were also encouraged to write down their experiment experiences and opinions about the two interfaces in a post-task questionnaire; these results will be presented and discussed in Sect. 5.3.

Table 3 Average number of documents opened by subjects for each query (AIS Collection)

Query	Number of documents opened before four relevant documents were found (Mean ± Standard deviation)		Significance test on the difference of the mean values (p-value)
	Traditional interface	Proposed interface	
1	6.9 ± 4.6	3.2 ± 2.7	<.01
2	9.9 ± 4.4	5.5 ± 4.4	<.01
3	5.9 ± 2.8	4.2 ± 2.7	<.05
4	7.4 ± 4.8	4.4 ± 3.5	<.01
All	7.5 ± 4.7	4.4 ± 3.4	<.01

The average number of documents opened with the traditional interface and the number opened using the proposed interface are displayed in Table 3. Table 3 shows that our proposed interface outperformed the traditional interface and the difference was statistically significant (based on t tests results) for each of these four queries and also significant when we examine them together. To obtain four relevant documents for each query, the subjects needed to open, on average, 4.4 documents with the proposed interface. In contrast, this number was 7.5 with the traditional interface. This means by looking at the keyphrases with each returned hit, the subjects could judge the content of a document more accurately, and therefore the effort spending on downloading and examining the irrelevant ones was reduced. The result of the experiment with AIS collection supports our hypothesis. For query 1, from Table 3 we can see that the average number of documents opened by the subjects using the proposed interface was even less than four, the required number of relevant documents. This means, for some returned documents, the users could determine their relevance by just looking at the document title, document keyphrases and the snippet, without opening the documents, and users were still able to identify 4 required relevant documents.

5.2 Experiment on the *Merry-Go-Round* collection

To understand whether the domain of the documents was a factor influencing the results, we also did an experiment on the *Merry-Go-Round* document collection. This collection had 3,425 documents. On average, each document had about 1,238 words. KIP was used to extract document keyphrases for *Merry-Go-Round* documents. A glossary database related to economy, politics and government was built for KIP to extract keyphrases for these documents. This glossary had 3,112 pre-defined keyphrases and was built by domain experts from Washington Research Library Consortium. The search system also had both the traditional interface and the proposed interface. The procedure for the experiment on *Merry-Go-Round* collection was the same as the basic procedure of the experiment on the AIS collection. The differences between the two experiments performed are the document collection, the number of subjects, and the queries.

We recruited 12 subjects to participate this experiment; these subjects had previous knowledge in business, sociology, and information systems. Everyone is familiar with the traditional search interface such as Google. These 12 subjects were not the same subjects participating in the experiment with the AIS collection. They were also randomly evenly

Table 4 Average number of documents opened by subjects for each query (*Merry-Go-Round* collection)

Query	Number of documents downloaded and examined before four relevant documents were found (Mean ± Standard deviation)		Significance test on the difference of the mean values (*p*-value)
	Traditional interface	Proposed interface	
1	7.7 ± 1.8	5.3 ± 1.7	<.05
2	9.0 ± 2.3	5.8 ± 1.7	<.01
3	8.3 ± 2.2	5.7 ± 1.6	<.01
4	8.8 ± 1.9	6.2 ± 2.2	<.01
All	8.5 ± 2.0	5.8 ± 1.8	<.01

divided into two groups, group A and group B. Four queries related to the domain of politics were designed and evenly divided into two sets. The rest of the procedure was same as that in the experiment on AIS collection. After executing each query, the subjects were asked to find four relevant documents and record how many documents they had opened in order to obtain four relevant documents. Each query was executed 12 times (once by each subject), six of which were with the traditional interface and the other six with the proposed interface. The post-questionnaire results from this experiment will also be presented in Sect. 5.3.

The average numbers of documents opened with the traditional interface and the proposed interface are shown in Table 4. The result shows that the proposed search interface outperformed the traditional interface, and the difference was statistically significant (based on t tests results) for each of these four queries. To obtain four relevant documents for each query, the subjects needed to open, on average, 5.8 documents with the proposed interface. In contrast, this number was much higher at 8.5 opened documents with the traditional interface.

5.3 Results of the post-questionnaire

The post-questionnaire was used to explore users' opinions about the two kinds of search interfaces, and it was the same for both the AIS collection and *Merry-Go-Round* collection. Therefore, we combined the post-questionnaire results for both experiments together to do the analysis.

Our post-questionnaire was adapted from that of Gutwin et al. (2003) which utilized a set of post-questions and interview questions for evaluating a new search engine, Keyphind, that supports browsing with keyphrase indexes. Four (Question 1, 2, 3 and 5) out of our five questions were adapted from their questionnaires. These 5 questions and subjects' responses are shown in Table 5. All 54 subjects' responses to the post-questionnaire are combined in Table 5. The numbers in the table cells represent the numbers of subjects who selected the corresponding answer.

From Table 5, we can see that 47 out of 54 subjects thought it was easier to carry out the task with one of the two kinds of search interfaces. Among these 47 users, 43 of them thought the proposed interface made the task easier. From subjects' responses (subjects were encouraged to write down their experiences, observations, or opinions about the experiment and the two kinds of interfaces after the experiment, although it was not

Table 5 Post-questionnaire and subjects' responses

Questions	Answers		
1. Was it easier to carry the task with one or the other of the two search interfaces?	Yes	Cannot tell	No
	47	1	6
2. If yes, which one?	The traditional one	The proposed one	
	4	43	
3. If yes, was the task: slightly easier, somewhat easier, or much easier?	Slightly easier	Somewhat easier	Much easier
	10 (three of them refer to the traditional one)	16 (one of them refers to the traditional one)	21
4. Did the document keyphrases make the screen too busy?	Yes	Cannot tell	No
	9	4	41
5. Would you use a search interface like the proposed one in your work?	Yes	Cannot tell	No
	43	6	5

required), we found the main reason was that document keyphrases made it easier for the user to predict the content of a document before opening it, as one subject described, "Document keyphrase Interface supersedes the traditional interface, because its keyphrases approximating seven or eight makes one to locate the relevant documents faster. The keyphrases are italic and are closely related to document heading." All 4 subjects who thought the traditional one was easier for the task were from the experiment on AIS collection. The main reasons they thought the traditional one was better are: the titles of some returned documents (they are all academic papers) could clearly indicate whether the document were relevant or not, so there was no need to check the keyphrases; the quality of some keyphrases were not useful and were not related to the query (the keyphrases were generated based on the main content of a document NOT in relation to the specific queries presented to the users).

In order to see if there is a significant difference between the number of subjects considering one interface better than the other and the number of subjects who did not think so or "cannot tell," we did a significance test. The simultaneous confidence intervals for multinomial proportion are calculated for all the three answers of question 1 (yes, cannot tell, and no) using Goodman's algorithm (Goodman 1965). Based on the simultaneous confidence intervals, we found that the number of subjects considering one interface appeared better than the other was significantly greater than the number of subjects who did not think so or cannot tell with a p-value less than .01. The simultaneous confidence intervals for multinomial proportion were also calculated for the two kinds of answers of question 2 (the traditional one and the proposed one). The result shows that the number of subjects considering the proposed interface was better than the traditional one is significantly greater than (p-value $< .001$) the number of subjects who thought the traditional one was better than the proposed one.

We added a list of keyphrases to each returned hit, so we also wanted to know if these keyphrases made the screen too busy or too cluttered. From subjects' responses to question 4, we can see that most of the subjects did not think the screen was too busy. The result is significant at the level of $p = .05$, using the significant test method mentioned above.

The last question asked the subjects if they would like to use a search interface like the proposed one in the future. Most subjects answered this question "yes." The result is also

significant at the level of $p = .05$, using the significant test method mentioned above, which means the number of users' who would like to use the proposed interface in the future is significantly greater than the number of users who would not or did not know.

As mentioned in Sect. 4, another feature of the proposed interface is that each displayed keyphrase is also a hyperlink. When users click on a keyphrase, all the documents containing this keyphrase will be retrieved and displayed. Actually, this feature provides a query refinement and browsing function. Although in this experiment we did not evaluate this feature, some subjects did try it and like this feature. One statement from a subject's response is "What I did like about the document key phrases was that it invoked a new search automatically to help refine the search. While this feature was not required for this task, I did experiment with it a bit."

6 Discussion

One major limitation of this study is the small number of queries used in the experiments of comparing the two different kinds of search interfaces. In the two experiments for the two collections, eight different queries were conducted. Although the number of queries was small, the results from the two experiments were statistically significant, and the analysis of the post-questionnaire results also supported the conclusion of the two experiments. This indicates that it is worth to conduct a future study which compares the two interfaces with a large number of queries over standard collections, e.g., TREC. In this future study, additional measures and steps may be employed, such as recording the time a subject spends on judging a document's relevance before opening it, which is discussed in detail in the next paragraph.

Another limitation that needs to be mentioned is the titles used to describe the two kinds of search interfaces in the user experiment, "the traditional interface" and "the proposed interface." The two terms "traditional and "proposed" would have made the subjects' answers in the questionnaire biased. When the subjects saw these two terms, they were very likely to give the "proposed" interface higher ratings. This is definitely a defect of the experimental design of this study.

By examining the keyphrases of a returned hit, users can better predict this document's main theme, and therefore reduce the time spending on downloading and checking the irrelevant ones. However, the cost of this is that the user needs to spend more time to check the keyphrases, though the time required is only on the order of seconds. In the experiments conducted in this study, the number of opened documents, instead of time, was used as the measure to evaluate the effectiveness of adding keyphrases to search results. One limitation with this measure is that it did not consider the extra time the subjects spent on reading the added keyphrases. However, using time instead of the number of opened documents as the measure also has a limitation: the recorded time may not be reliable—subjects may do other things irrelevant to the experiment during the recorded time period, such as answering phone calls, which are difficult to know if the experiments are not closely monitored by camera or people. But, considering that time is a better metric if properly recorded, one of this study's limitations was that the number of opened documents, instead of time, was used as the evaluation measure. If the subjects had done the experiment in a designated place with proper monitoring, the elapsed time would have been recorded and used as the evaluation metric. However, in this study, the subjects could do the experiment from any place

with Internet access; the time obtained from web logs was not reliable enough to be the metric due to the problem mentioned above.

In this study, the keyphrases used in the experiments of comparing the two types of interfaces were generated by KIP. One may wonder if the experimental results would be different if the keyphrases were generated by a different keyphrase extractor. Although the results from the experiment comparing KIP to Kea and Extractor show that KIP outperformed the other two extractors and the results were statistically significant, the differences were not large. Therefore, we anticipate that if the keyphrases were extracted by a different keyphrase extractor, such as Kea, the results of the experiments would not change significantly. However, it might be an interesting future research topic to evaluate if these keyphrase extractors would perform differently in comparing the two types of search interfaces. We can expect that the experiment will require a very large group of subjects, since it will involve two independent variables, namely search interface and keyphrase extractor.

Besides the possible future study just mentioned, there are also other factors or questions which warrant further exploration. One question is the affect on the results if we use author-assigned keyphrases instead of the automatically generated ones in the proposed search interface. We can expect that the author-assigned keywords will better help users better predict the content of a returned hit, since typically the document author(s) knows his/her work best. However, the problem is that not all the documents have keyphrases assigned by their authors. In the AIS collection, only a small portion of the documents have author-assigned keyphrases; documents in the *Merry-Go-Round* collection do not have any author-assigned keyphrase. We believe this is true for most of the document collections, and also for most of the documents indexed by the commercial search engines, such as Google.

Another question is the importance of the number of keyphrases presented with the returned hits. Presenting more phrases will give more information to users and better help them predicate the content of a document, but users also need more time to read them and more screen spaces are occupied. The ideal number of keyphrases displayed in the search hits should be further explored. In the experiments of this study, for each document, the noun phrases meeting the following requirements were extracted as keyphrases and presented to the subjects: (1) it was among the top 10 in the noun phrase rank list and (2) its score was greater than 0.5. For most returned hits, the number of keyphrases displayed was between 6 and 10. We think this range is reasonable, since it could provide a reasonable number of keyphrases without making the screen too cluttered. Authors of academic papers are usually suggested to provide four to eight keywords for their papers.

A more interesting question worth to explore is what will happen if we change the document snippet in terms of its existence and length: what will the results be when the returned hits have (1) keyphrases but not snippets, (2) both snippets and keyphrases, and (3) longer snippets but not keyphrases? An experiment comparing these three types of interfaces might tell us if keyphrases can replace snippets and if longer snippets can accomplish the same purpose of adding keyphrases to the returned hits demonstrated in this study. In most cases, the snippet appearing in a search result consists of a couple of sentences which are extracted from the document and contain part or the entire query words. It shows the context in which the query terms appear, but does not necessarily represent the main topics of this document. In contrast, document keyphrases describe the main content of a document. In other words, snippets are query-dependant and keyphrases are query-independent. Some retrieval

systems may use a document's abstract as its snippet appearing in the search results. In the case, the snippet is query-independent. Most retrieval systems are not able to use query-independent snippets, because not all the documents have an abstract, and automatically summarizing a document to get a quality abstract is still a challenging job. In McDonald and Chen's study (2006), they compare the generic document summary (query-independent snippet) and query-based summary (query-dependant snippet) in the context of browsing tasks and searching tasks. Their experimental results show that the query-based summary outperforms the generic summary in the searching tasks, while the generic summary is better than the query-based summary in the browsing tasks. Cutrell and Guan (2007) use eye-tracking techniques to explore the effects of changes in the length of query-based document snippets in search results. They find that increasing the length of snippets will significantly improve performance for informational tasks but degrade performance for navigational tasks. In navigational tasks, users are trying to find a specific website that they have in mind. The goal is simply to get to their destination. In informational tasks, the goal is to find some kind of information despite of where it might be located. These two studies show that longer query-based snippets will help users make relevance judgment in their search tasks. It will be very interesting to see among (1) and (3) which one will perform better in fulfilling users' search tasks. McDonald and Chen's study indicates that (3), which uses longer query-dependant snippets without document keyphrases, might perform better. One factor affecting the comparison results will be the quality of the snippets extracted from the documents. Query terms may appear in many sentences of a document, the search engine's ability to extract the most representative sentences as the snippet will affect the snippet quality.

In the last paragraph, we have discussed that the document snippets may be query-dependant or query-independent, and they might have different effects on helping users in their search tasks. Similarly, document keyphrases may also be classified into two types: query-dependant and query-independent. Previous studies, including this one, focus only on query-independent keyphrase extraction techniques. Query-independent keyphrases represent the main topics of a document, but they may not be related to the search query. Among others, when calculating a phrase's score, the two main factors that many keyphrase extraction algorithms usually consider are the location and frequency of a phrase. Usually phrases appearing in the document title and abstract are given higher weights, since phrases appearing in these places are usually related to the main theme of the document. Therefore, phrases appearing in the title and abstract are more likely selected as keyphrases, which is reasonable. This is why for certain returned documents, some of their keyphrases are redundant with the title and abstract. Query-independent keyphrases are extracted offline, and therefore, some complicated algorithms can be employed. In contrast, generating query-dependant keyphrases needs to take the query into consideration, and so the keyphrases are extracted on the fly and the algorithms should be robust enough to be able to extract keyphrases in time. One possible solution is to extract the noun phrases (and also verb phrases if necessary) from documents offline first, and then extract keyphrases from these noun phrases on the fly based on their relationships with the query terms, such as co-appearing in the same sentence. After implementing a query-dependant keyphrase extractor, it would become possible to further investigate how these two kinds of keyphrases perform in helping users in their search tasks. McDonald and Chen's study (2006) on comparing generic and query-based summaries in users' search tasks indicates that the query-dependant keyphrases might

perform better. As mentioned before, in their study, the query-based summary performs better than the generic summary.

Phrase indexing has been studied by many previous studies (Gutwin et al. 2003; Fagan 1989; Anick and Vaithyanathan 1997; Arampatzis et al. 1998). These studies have shown that phrase indexing can improve retrieval effectiveness. In this study, we built a key-phrase-document index. Besides its usage described in Sect. 4.2, it can also be used as a small-scale phrase index, which can be incorporated in the architecture of phrase indexing described in previous studies. We believe that the mechanism of incorporating keyphrases in search results described in this paper can be easily combined with other phrase-based retrieval or browsing solutions. Another possible usage of document keyphrases is document ranking in a retrieval system. One possible way to improve the ranking quality of returned documents for a query is to see whether there is any kind of match between phrases in the query and keyphrases of a retrieved document. If there is a match, perhaps this document tends to be more about the query, and it should have a higher rank in the return list.

7 Conclusion

A new search interface where the document keyphrases are incorporated in a query's returned hits is presented in this paper. It provides a solution to the problem that the metadata of a returned hit is not rich enough for users to predict the relevance of a document to a query. Keyphrases provide a concise summary of a document's content, offering semantic metadata characterizing a document. By looking at the keyphrases of each returned hit, the user can predict the content of a document more precisely, and therefore the effort expended on downloading and examining the irrelevant ones will be reduced. In this article, we describe our proposed search interface, our key phrase extraction algorithm, and the way of building search indexes to implement our solution. The results of the experiment clearly demonstrate that users preferred our proposed search interface that displayed document keyphrases.

Acknowledgements The authors would like to thank Allison Zhang, Manager of Digital Collections Production Center, Washington Research Library Consortium, for providing us the document collection and building the glossary for our experiment. Partial support for this research was provided by the United Parcel Service Foundation; the National Science Foundation under grants DUE-0226075, DUE-0434581 and DUE-0434998, and the Institute for Museum and Library Services under grant LG-02-04-0002-04.

Appendix A: Experimental queries and an example of search results

Table A.1 Queries used in the experiments

Collection	Query			
AIS	Knowledge management	Software development life cycle	Distance learning	Database design
Merry-Go-Round	Electric power industry	Economic security program	Agricultural policy	Capital investment

Fig. A.1 An example of search results with document keyphrases

References

Anick, P., & Tiperneni, S. (1999). The paraphrase search assistant: Terminological feedback for iterative information seeking. In *Proceedings of SIGIR'99: The 22nd Annual International Conference on Research and Development in Information Retrieval* (pp. 153–159). Berkeley, CA, USA.

Anick, P., & Vaithyanathan, S. (1997). Exploiting clustering and phrases for context-based information retrieval. In *Proceedings of SIGIR'97: The 20th Annual International Conference on Research and Development in Information Retrieval* (pp. 314–322). Philadelphia, PA: ACM Press.

Arampatzis, A. T., Tsoris, T., Koster, C. H., & Weide, T. (1998). Phrase-based information retrieval. *Information Processing and Management, 34*(6), 693–707.

Argamon, S., Dagan, I., & Krymolowski, Y. (1999). A memory-based approach to learning shallow natural language patterns. *Journal of Experimental and Theoretical Artificial Intelligence (JETAI), 11*(3), 369–390.

Brill, E. (1995). Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics, 21*(4), 543–565.

Cardie, C., & Pierce, D. (1999). The role of lexicalization and pruning for base noun phrase grammars. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence* (pp. 423–430). Orlando, Florida.

Croft, B., Turtle, H., & Lewis, D. (1991). The use of phrases and structured queries in information retrieval. In A. Bookstein, Y. Chiaramella, G. Salton, & V. V. Raghavan, (Eds.), *Proceeding of SIGIR'91: The 14th Annual International Conference on Research and Development in Information Retrieval, Philadelphia* (pp. 32–45). New York: ACM Press.

Cutrell, E., & Guan, Z. (2007). What are you looking for? An eye-tracking study of information usage in Web search. In *Proceedings of CHI 2007*, April 28–May 3, 2007, San Jose, CA.

Davis, G. B. (1997). *Blackwell encyclopedic dictionary of management information system*. Malden, MA: Blackwell Publishing.

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.

Frank, E., Paynter, G., Witten, I., Gutwin, C., & Nevill-Manning, C. (1999). Domain-specific keyphrase extraction. In *Proceeding of the Sixteenth International Joint Conference on Artificial Intelligence* (pp. 668–673). San Mateo, CA.

Goodman, L. A. (1965). On simultaneous confidence intervals for multinomial proportions. *Technometrics 7*, 247–254.

Gutwin, C., Paynter, G., Witten, I., Nevill-Manning, C., & Frank, E. (2003). Improving browsing in digital libraries with keyphrase indexes. *Journal of. Decision Support Systems, 27*(1–2), 81–104.

Jones, S., & Mahoui, M. (2000). Hierarchical document clustering using automatically extracted keyphrase. In *Proceeding of the Third International Asian Conference on Digital Libraries* (pp. 113–120). Seoul, Korea.

Jones, S., & Paynter, G. (2002). Automatic extraction of document keyphrases for use in digital libraries: evaluation and applications. *Journal of the American Society for Information Science and Technology, 53*(8), 653–677.

Jones, S., & Staveley, M. (1999). Phrasier: A system for interactive document retrieval using keyphrases. In *Proceedings of SIGIR'99: The 22nd International Conference on Research and Development in Information Retrieval* (pp. 160–167). Berkeley, CA: ACM Press.

Kosovac, B., Vanier, D. J., & Froese, T. M. (2000). Use of keyphrase extraction software for creation of an AEC/FM thesaurus. *Journal of Information Technology in Construction, 5*, 25–36.

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

McDonald, D., & Chen, H. (2006). Summary in context: Searching versus browsing. *ACM Transaction on Information Systems, 24*(1), 111–141.

Muñoz, M., Punyakanok, V., Roth, D., & Zimak, D (1999). A learning approach to shallow parsing. *Proceedings of EMNLP/WVLC-99*, University of Maryland, MD.

Ramshaw L. A., & Marcus, M. P (1995). Text chunking using transformation-based learning. In *Proceedings of the Third Workshop on Very Large Corpora*. Cambridge, MA.

Sang, E. F (2000). Noun phrase representation by system combination. *Proceedings of ANLP-NAACL 2000*, Seattle, WA.

Tolle, K. M., & Chen, H. (2000). Comparing noun phrasing techniques for use with medical digital library tools. *Journal of the American Society for Information Science, 51*(4), 352–370.

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

Wacholder, N., Evans, D. K., & Klavans, J. L. (2001). Automatic identification and organization of index terms for interactive browsin. In *Proceedings of the First ACM/IEEE-CS Joint Conference on Digital Libraries* (pp. 126–134). Roanoke, VA, USA.

Witten, I. H. (1999). Browsing around a digital library. In *Proceeding of Australasian Computer Science Conference* (pp. 1–14). Auckland, New Zealand.

Wu, Y. B., Li, Q., Bot, R., & Chen, X. (2006). Finding nuggets in documents: A machine learning approach. *Journal of the American society for information and technology (JASIST), 57*(6), 740–752.

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