Finding Nuggets in Documents: A Machine Learning Approach

Yi-fang Brook Wu, Quanzhi Li, Razvan Stefan Bot, and Xin Chen
Information Systems Department, New Jersey Institute of Technology, Newark, NJ 07102.
E-mail:{wu, QL23, rsb2, xc7}@njit.edu

Document keyphrases provide a concise summary of a document's content, offering semantic metadata summarizing a document. They can be used in many applications related to knowledge management and text mining, such as automatic text summarization, development of search engines, document clustering, document classification, thesaurus construction, and browsing interfaces. Because only a small portion of documents have keyphrases assigned by authors, and it is time-consuming and costly to manually assign keyphrases to documents, it is necessary to develop an algorithm to automatically generate keyphrases for documents. This paper describes a Keyphrase Identification Program (KIP), which extracts document keyphrases by using prior positive samples of human identified phrases to assign weights to the candidate keyphrases. The logic of our algorithm is: The more keywords a candidate keyphrase contains and the more significant these keywords are, the more likely this candidate phrase is a keyphrase. KIP's learning function can enrich the glossary database by automatically adding new identified keyphrases to the database. KIP's personalization feature will let the user build a glossary database specifically suitable for the area of his/her interest. The evaluation results show that KIP's performance is better than the systems we compared to and that the learning function is effective.

Introduction

Successful knowledge management is a key to obtaining competitive advantages, and text mining is a highly useful tool for successful knowledge management. As textual information pervades the Web and local information systems, text mining is becoming more and more important for deriving competitive advantages. One of the keys to successful text mining applications is the ability to extract salient document features, to which the mining algorithm is applied in order to discover interesting patterns or relationships. However, many text mining applications do not have adequate natural language processing ability beyond simple keyword indexing, and as a result, there are too many textual elements (words) included in the analysis. We argue that noun phrases as textual elements are better suited for text mining and could provide more discriminating power, than single words. Discourse representation theory (Kamp, 1981) and language learning of children (Snow & Ferguson, 1997) show that a document's primary concepts are carried by noun phrases. Because noun phrase in a document are not equally important, we propose using them as candidates and identifying *keyphrases* from them. Document keyphrases are the most important topical phrases for a given document, and they address the main topics of that document. Our study proposes a Keyphrase Identification Program (KIP) to approach this problem by analyzing the composition of noun phrases.

Keyphrases provide semantic metadata that can characterize documents and produce an overview of the content of a document. Keyphrases can be used in many text-mining related applications. If keyphrases are used in automatic text summarization, applications can extract sentences with more keyphrases or higher keyphrase scores. If keyphrases are used as document metadata, applications can use them to efficiently classify or cluster documents into different categories. They may be utilized to enrich the metadata of the results returned from a search engine. Another use is that some search engines implement interactive query refinement using keyphrases, and also use them as a way of browsing a collection. Last, but not least, keyphrases may be extracted from documents to construct a domain glossary or thesaurus. The previous studies of the various applications of keyphrases will be presented in the next section.

Some documents, mostly scholarly papers, have a list of keyphrases provided by authors, but unfortunately, most documents do not have author-assigned keyphrases. Keyphrases can also be assigned manually by professional indexers. The indexers may choose phrases from the document text as keyphrases, or, more commonly, choose phrases from a predefined controlled vocabulary. However, manually assigning keyphrases to documents is costly and tedious, and the results

Received November 8, 2004; revised February 10, 2005; accepted March 15, 2005

 ● Published online 17 February 2006 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/asi.20341

may not be consistent. So there is a need for automatic keyphrase generation techniques.

Basically, there are two approaches for automatic keyphrase generation: keyphrase assignment and keyphrase extraction. Keyphrase assignment algorithms select, from a controlled vocabulary, phrases that best describe a document. Keyphrase extraction techniques choose phrases from the body of document text as keyphrases; they do not need a predefined controlled vocabulary. The problem with keyphrase assignment is that most controlled vocabularies are not updated frequently enough, and some potentially useful keyphrases will be ignored if they appear in the document but not in the controlled vocabulary. In some domains, such controlled vocabularies might not even be available. Therefore, automatic keyphrase extraction is desirable.

In this paper, we propose a keyphrase identification program, which has a learning function and a personalization feature. KIP is a keyphrase extraction technique, meaning the keyphrases generated by KIP must appear in the document. Its algorithm considers the composition of a noun phrase. To analyze a noun phrase and assign a score to it KIP uses a glossary database, which contains manually pre-identified domain-specific keyphrases and keywords that can be used for calculating scores for noun phrases in the document. The noun phrases having higher scores will be extracted as keyphrases. In this paper we will also discuss KIP's learning function, which can enrich its glossary database by automatically adding new keyphrases, extracted from documents, to the database. Consequently, the database will grow gradually and the system performance will be improved. KIP also has a personalization feature, which will let a user build a glossary database specifically tailored to the area of his or her interest. Thereafter, using this personalized glossary database, KIP can extract keyphrases more effectively from documents of interest to the user.

In the following sections, we first discuss prior studies of the applications of keyphrases and other keyphrase extraction algorithms; then we describe our methodology and KIP's learning function and personalization feature; and finally we present our experiments and research issues discovered.

Related Work

In this section, previous studies of the applications of document keyphrases are presented first, and then the related work of keyphrase extraction techniques is discussed.

Applications of Keyphrases

Previous studies have shown that document keyphrases can be used in a variety of applications, such as retrieval engines (Li, Wu, Bot, & Chen, 2004; Jones & Staveley, 1999), browsing interfaces (Gutwin, Paynter, Witten, Nevill-Manning, & Frank, 1999), thesaurus construction (Kosovac, Vanier, & Froese, 2000), and document classification and clustering (Jones & Mahoui, 2000). These studies are described below.

Li et al. (2004) proposed a mechanism of enriching the metadata of the returned results of a search engine by incorporating automatically extracted document keyphrases in each returned hit. By looking at the keyphrases in each returned hit, a user can predict the content of the document more accurately and more quickly. Keyphrases and snippets together can provide a better overall picture of a returned hit than a returned snippet alone does, for the latter only shows a very small fraction of the document body of the returned hit. Li, Wu, Bot & Chen's (2004) experimental results show that this solution may save users' time up to 32% and that most would like to use the proposed search interface with document keyphrases as part of the metadata of a returned hit.

Jones and Staveley (1999) developed an interactive system, Phrasier, which automatically introduces links between related materials and 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 interdocument navigation.

Gutwin et al. (1999) built a search engine, Keyphind, which is a mixture of searching and browsing mechanisms that help users 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 that of words and documents. Keyphind's keyphrase index also provides a simple mechanism for refining queries, previewing results, and clustering documents. Gutwin et al. found that phrase-based indexing and presentation offer better support for browsing tasks than the traditional query engines.

Kosovac et al. (2000) described a method of constructing a thesaurus in the construction domain. They use Extractor, a keyphrase extraction system that automatically extracts keyphrases from documents, to collect candidate thesaurus terms from Internet sources.

The studies described above are about the applications of keyphrases. Because document keyphrases are a selective subset of a document's noun phrases, it is very possible that keyphrases can also be used in some places where noun phrases are applied, and that better results may be obtained. In the following paragraphs we also describe several previous studies of the applications of noun phrases.

Croft, Turtle, and Lewis (1991) proposed a method in which 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 that phrases extracted automatically from a natural language query perform nearly as well as manually selected phrases.

Anick and Vaithyanathan (1997) described 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, whereas

DOI: 10.1002/asi

cluster contexts act as a set of logical foci for query refinement and browsing. Anick and Vaithyanathan use a 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 terminates in a head noun.

Larkey (1999) developed a system for searching and classifying U.S. patent documents, based on INQUERY. The system includes a *phrase help* facility, which can help users find and add phrases and terms related to those in their queries. 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 tags are assigned to the terms in the resulting sequences using WordNet (Fellbaum, 1998). The sequences satisfying rules that define noun phrases and certain other criteria are retained as phrases.

Keyphrase Extraction

Several automatic keyphrase extraction techniques have been proposed. Tomokiyo and Hurst (2003) reported a method of extracting keyphrases based on statistical language models. Their method uses pointwise KL-divergence (Cover & Thomas, 1991) between multiple language models for scoring both *phraseness* and *informativeness*, which are unified into a single score to rank extracted phrases. Phraseness and informativeness are two features of a keyphrase. *Phraseness* is an abstract notion describing the degree to which a given word sequence is considered to be a phrase. *Informativeness* means how well a phrase captures the key ideas in a document collection. In the calculations, they use the relationship between foreground and background corpora to formalize the notion of informativeness. The target document collection from which representative keyphrases are extracted is called the *foreground* corpus. The document collection to which this target collection is compared is called the *background* corpus. Tomokiyo and Hurst's approach needs a foreground corpus and a background corpus, and the latter is not easy to obtain. The phrase list it generates is for the whole corpus, not an individual document. Their paper does not report any evaluation of their approach.

Heuristics are used by Krulwich and Burkey (1996) to extract significant topical phrases from a document. The heuristics are based on documents' structural features, such as the presence of phrases in document section headers, the use of italics, and formats different from surrounding text. The approach proposed by Krulwich and Burkey is not difficult to implement, but the limitation is that not every document has explicit structural features.

Zha (2002) proposed a method for keyphrase extraction by modeling documents as weighted undirected and weighted bipartite graphs. Spectral graph clustering algorithms are used for partitioning sentences of the documents into topical groups. Within each topical group, the mutual reinforcement principle is used to compute keyphrase and sentence saliency scores. The keyphrases and sentences are then ranked according to their saliency scores. Then keyphrases are selected for inclusion in the top keyphrase list and sentences are also selected for inclusion in summaries of the document. In this approach, a phrase's frequency in the document is the dominant factor contributing to its score. Zha's paper does not give evaluation information for the extracted keyphrases.

Kea uses a machine learning algorithm based on Naive Bayes decision rule (Frank, Paynter, Witten, Gutwin, & Nevill-Manning, 1999; Witten, Paynter, Frank, Gutwin, & Nevill-Manning, 1999). It has some prebuilt models. A model is used to identify the keyphrases within a document. The model is learned from the training documents with exemplar keyphrases and corresponds to a specific corpus containing the training documents. Each model consists of a Naive Bayes classifier and two supporting files that contain phrase frequencies and stopped words. A model can be used to identify keyphrases from other documents once it is learned from the training documents.

The first person who treated the problem of phrase extraction as supervised learning from examples was Turney (2000). Turney used nine features to score a candidate phrase; two of the features are the location of the first occurrence of the phrase in the document, and whether or not the phrase is a proper noun. Keyphrases are extracted from candidate phrases based on examination of their features. Turney introduced two kinds of algorithms: C4.5 decision tree induction algorithm, and GenEx, which 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, Gentor is no longer used and Extractor alone can extract keyphrases using the optimal parameter setting obtained from the training stage. In Extractor's formula for calculating a phrase's score, the dominant factors are the frequency of the phrase, the frequencies of words within it, and the location of its first occurrence.

Both Kea and Extractor use supervised machine learning approaches. And both use corpora to train the program. For each document in the corpus, there must be a target set of keyphrases provided by authors or generated by experts. Initiation of training requires a lot of manual work, and in many applications there is no appropriate document set that can be used to train the algorithm. Another limitation is that the training corpora are usually not up-to-date and retraining the program with new corpora is not easy. So, the program implemented using these two approaches may not be effective at identifying keyphrases for documents with new topics.

Based on prior studies, we are looking for a method that can identify real keyphrases now and will also be able to adapt, gradually and automatically, to new developments and advances in the domain of documents from which it derives keyphrases. We describe our algorithm in detail in the following section.

DOI: 10.1002/asi

KIP: A Machine Learning Keyphrase Extraction Algorithm

We considered the following aspects when designing KIP:

1. It is a keyphrase extraction program, not a keyphrase assignment program.
2. The program has to be able to learn to adapt to new developments in a chosen domain.
3. KIP can be personalized to effectively extract keyphrases from documents of specific interest to a user.

KIP is a domain-specific keyphrase extraction program. We describe the algorithm in this section and will explain its learning function and personalization feature in the next section.

KIP algorithm is based on the logic that a noun phrase containing domain-specific keywords and/or keyphrases is likely to be a keyphrase. The more keywords/keyphrases it contains and the more significant the keywords/keyphrases are, the more likely that this noun phrase is a keyphrase. The pre-identified domain-specific keywords and keyphrases are stored in the system glossary database, which is used to calculate scores for noun phrases. Here a predefined domain-specific keyword means a single term word, and a predefined domain-specific keyphrase means a phrase containing one or more words. A keyphrase generated by KIP can be a single-term keyphrase or a multiple-term keyphrase. KIP may generate keyphrases of any length between one and six words. KIP operations can be summarized as follows. KIP first extracts a list of keyphrase candidates, which are noun phrases from input documents. Then it examines the composition of a keyphrase candidate (a noun phrase) 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 subphrases it contains), and how specific these words and subphrases are to the domain of the document. To calculate scores for noun phrases, readily available human identified domain-specific keyphrases are parsed to form a glossary database. Finally, all the noun phrases are ranked in descending order by their scores, and those with higher scores are selected as keyphrases of the document.

KIP has the following main components: a tokenizer, a part-of-speech (POS) tagger, a noun phrase extractor, a keyphrase extraction tool, a learning function, and a personalization feature. The first four components are introduced in this section, and the learning function and the personalization feature will be discussed in the next section. To distinguish the two kinds of keyphrases, i.e., human identified domain-specific keyphrases and KIP identified keyphrases, we refer to the former as manual keyphrases and the latter as automatic keyphrases.

Tokenizer

After documents are loaded into the system, Tokenizer will separate all the words, punctuation marks, and other symbols from the document text to obtain the atomic units.

Part-of-Speech Tagger

At this stage, each word is assigned an initial POS tag. We use the WordNet lexical database v2.0 (Fellbaum, 1998) to assign the right POS tag to each word. This database contains words grouped into four categories (noun, verb, adjective, and adverb) and, for each word, the number of senses used within categories it belongs to. A word is marked as a multi-tag word if it appears in more than one category. A word's initial POS tag is determined by the category having the maximum number of senses of this word. For every multi-tag word, the sequence of the POS tags of the preceeding n tokens (n is between 2 and 4) is examined against a list of predefined syntactic rules to determine its correct POS tag. For example, "hit" can be either a noun or a verb. If the preceeding word is a determiner (*the*, *a*, *this*, etc.), it will be tagged as a noun and the multi-tag mark removed. Heuristics are used to determine a word's POS tag if it is not found in any of the categories and if its POS tag cannot be solved by the syntactic rules. For instance, if a word is not found in the lexical database, but ends with *tion*, it is tagged as a noun.

Noun Phrase Extractor

Noun phrases are extracted after the document text is tagged. KIP's noun phrase 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. A set of optional rules is also used. Phrases satisfying the above sequence patterns or the optional rules will be extracted as noun phrases. Users may choose to obtain noun phrases of different lengths by changing system parameters. Ramshaw and Marcus (1995) put forward a standard data set for the evaluation of noun phrase identification approaches. Precision and recall are used to measure the performance of the algorithm. *Precision* measures how many noun phrases identified by the approach are correct, and *recall* measures the percentage of noun phrases identified by the approach. For the Ramshaw and Marcus data set, many evaluation results of noun phrase identification approaches have been published (Sang, 2000; Argamon, Dagan, & Krymolowski, 1999; Cardie & Pierce 1999; Muñoz, Punyakanok, Roth, & Zimak, 1999). The preliminary testing result of our noun phrase identification approach shows that the precision is 95% and the recall is 85%. Precision and recall can be combined in one measure: the F measure. $F = 2 \times \text{precision} \times \text{recall}/(\text{recall} + \text{precision})$. The F of our noun phrase extractor is 0.90. It is comparable to other approaches mentioned above, whose F values range from 0.89 to 0.93. At this stage, KIP produces a list of noun phrases, which will be used in the next stage, keyphrase extraction.

Extracting Keyphrases

The noun phrases identified by NPE are keyphrase candidates. They will be assigned scores and ranked at this stage.

DOI: 10.1002/asi

Noun phrases with higher scores will be extracted as the document's keyphrases. As previously described, KIP's algorithm examines the composition of a noun phrase to assign a score to it. In order to calculate the scores for noun phrases, we use a glossary database containing domain-specific manual keyphrases and keywords, which provide initial weights for the keywords and subphrases of a candidate keyphrase. In the following subsections, we will describe, first, how to build this database, then how to calculate a noun phrase's score, and finally how the keyphrases are extracted.

Building a glossary database. The glossary database is a key component of KIP. First, we need to find or build a human-developed glossary or thesaurus (it is called *predefined glossary* hereafter) for the domain of interest. It could be as simple as users manually inputting some of the known keyphrases, or it could be as elaborated as those from published sources. Next, before the learning function is activated, the glossary database initially is populated with two lists (tables): (a) a manual keyphrase list and (b) a manual keyword list. A manual keyphrase means a phrase containing one or more words; and a manual keyword means a single word parsed from list (a). Before using KIP, users will need a corresponding glossary database from a particular domain. When the system is applied to a new domain, the only requirement is to build or change to a new database specific to the domain. We use the Information Systems (IS) domain as an example to illustrate how a domain-specific glossary database is built.

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 if the database contains more comprehensive human identified keyphrases and keywords, the performance of KIP will be better.

Keyphrase list. The keyphrase list was generated as follows. First, 3,000 abstracts from IS related journals were automatically processed, and all keyphrases 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 (Davis, 1997). The final keyphrase list contains 2,722 manual keyphrases.

Keyword list. The keyword list, initially, 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 manual keywords, all manual keyphrases were split into individual words and added as keywords to the keyword list. The final manual keyword list has 2,114 keywords.

The database has two tables, one for keyphrases and another for keywords. The keyphrase table has three columns (keyphrases, weights, and sources) and the keyword table has two columns (keywords and weights). The first column of the tables represents keyphrases/keywords. The second column represents the weights of keyphrases/ keywords. The third column of the keyphrase table represents the sources of keyphrases. Keyphrases in the keyphrase table may come from up to three sources. Initially, they are all manually identified in the way described above. During KIP's learning process, the system may automatically learn new phrases and add them to the keyphrase table. Users may also choose phrases from the processed documents and add them to the glossary database manually, if they think an unidentified phrase qualifies as a keyphrase. We discuss the details of KIP's learning process and how new phrases are added to the glossary database in the next section. The weights of these 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 only 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 the method of assigning weights to a keyword is that it reflects how specific a keyword is within the domain. The more specific a keyword, the higher weight it has. For each keyword in condition (a), the weight is X (the system default value for X is 10, but users have the option of defining weight X for keywords in category a between 1 and 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 2, but users have the option of specifying weight Y for keywords in category (b) between 0 and 10); for each keyword in condition (c), the weight is $\frac{X + \frac{Y}{N}}{2}$, where N is the times that the keyword appears as part of a keyphrase. Users can change the system default values of X and Y.
2. Assigning weights to keyphrases. The weight of each word in the manual keyphrase is found from the keyword table, and then all the weights of the words in this manual keyphrase are added together. The sum is the weight for this manual keyphrase.

The weights of manual keyphrases and keywords assigned by the above method will be used to calculate the scores of noun phrases in a document.

Calculating scores for noun phrases. A noun phrase's score is defined by multiplying a factor F, which is the frequency of this phrase in the document, by a factor S, which is the sum of weights of all the individual words and all the possible combinations of adjacent words within the noun

DOI: 10.1002/asi

phrase (we call the combination of adjacent words a *subphrase* of this noun phrase): The score of a noun phrase = $F \times S$. The factor F is the frequency of the noun phrase in this document. The sum of weights 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 and p_j is the weight of a subphrase within this noun phrase. The following example will better illustrate this concept. How to get the values of w_i and p_j will be explained later.

Assume there is a noun phrase *ABCD*, where *A*, *B*, *C* and *D* are four words. The possible combinations of adjacent words are *AB*, *BC*, *CD*, *ABC*, **BCD**, and *ABCD*. The score for noun phrase *ABCD* will be the frequency of *ABCD* in this document multiplied by the summation of weights of *A*, *B*, *C*, *D*, *AB*, *BC*, *CD*, *ABC*, *BCD*, and *ABCD*. The motivation for including the weights of all possible subphrases in the phrase score, in addition to the weights of individual words, is to find out if a subphrase is a manual keyphrase in the glossary database. If it is, this phrase is expected to be more important.

Suppose we have a noun phrase *noun phrase extraction*. The score for this noun phrase will be $F \times S_{noun\_phrase\_extraction}$, where $S_{noun\_phrase\_extraction} = W_{noun} + W_{phrase} + W_{extraction} + P_{noun\_phrase} + P_{phrase\_extraction} + P_{noun\_phrase\_extraction}$, and F is the frequency of phrase *noun phrase extraction* in the document.

KIP will look up the keyphrase table to obtain the weights for all the subphrases of the noun phrase. If a subphrase is found, the corresponding weight in the keyphrase table is assigned to this subphrase; otherwise, a predefined low weight will be assigned to this subphrase. This predefined weight is usually much smaller than the lowest weight of a keyphrase or keyword in the database, because it is for a new and previously unidentified phrase. The user can adjust this value by changing the system parameter. Similarly, KIP obtains the weight of a word by looking up the keyword table. If it finds the word from the table, the corresponding weight in the keyword table will be the weight of the word. Otherwise, a very low predefined weight will be assigned to it.

Extracting keyphrases. After the scores of all noun phrases in the document are calculated, they are normalized from 0 to 1 (all scores are divided by the highest score). Noun phrases in the document are then ranked in descending order by their scores.

The keyphrases of a document can be extracted from the ranked noun phrase list. In order to be as flexible as possible, the KIP system has a set of parameters to let the users decide how many keyphrases they want, or to let the system produce a reasonable number of keyphrases for a specific document, based on the document's length.

The number of extracted keyphrases for a document can be defined in three ways:

1. asking for a specific number of keyphrases to be extracted (for example, 10 keyphrases are to be extracted),
2. specifying the percentage of noun phrases to be extracted (for example, top 20% of all the identified noun phrases are to be extracted), and
3. setting a threshold for keyphrases to be extracted (for example, only noun phrases with scores greater than 0.5 will be extracted as keyphrases).

KIP contains all the above basic options, as well as possible combinations of above three basic options. For example, the system can extract noun phrases that are in the top 30% of the noun phrase list and with scores greater than 0.5.

The system has several system parameters that users can adjust, such as *X* and *Y* mentioned previously. However, most of the parameters do not need users' adjustment. Their default values are acquired based on our testing and observations. The purpose of providing extra options is that we want to make the system as flexible as possible. Such options are mainly for advanced users, in case they need to use KIP in some special applications. For most users, the only parameter they want/need to change is the desired number of keyphrases.

KIP's Learning Function and Personalization Feature

KIP's learning function can enrich the glossary database by automatically adding new identified keyphrases to the database. KIP's personalization feature will let the user build a glossary database specifically suitable for his or her area of interest. As such, the personalization feature is based on the learning function.

KIP's Learning Function

Many keyphrase applications and keyphrase extraction research efforts rely on keyphrases identified by a human as positive examples. Sometimes, such examples are not up to date or even not available. Therefore, an adaptation and learning function are necessary for KIP, so that it grows as the field of documents grows. This function is optional and the user can enable or disable it. With the learning function enabled, whenever the system identifies a new keyphrase (*new* means this keyphrase is not in the database's keyphrase table, and it satisfies the inclusion requirements), this keyphrase will be automatically added to the keyphrase table and the contained words will be added to the keyword table. The inclusion requirement can be modified by the user in a way similar to defining how many keyphrases will be extracted from a document, as described in the last section.

Weights of the affected keyphrases and keywords in the database will be recalculated. With this feature enabled, the database will grow gradually, and the system performance will be improved (we will illustrate this in the section of

DOI: 10.1002/asi

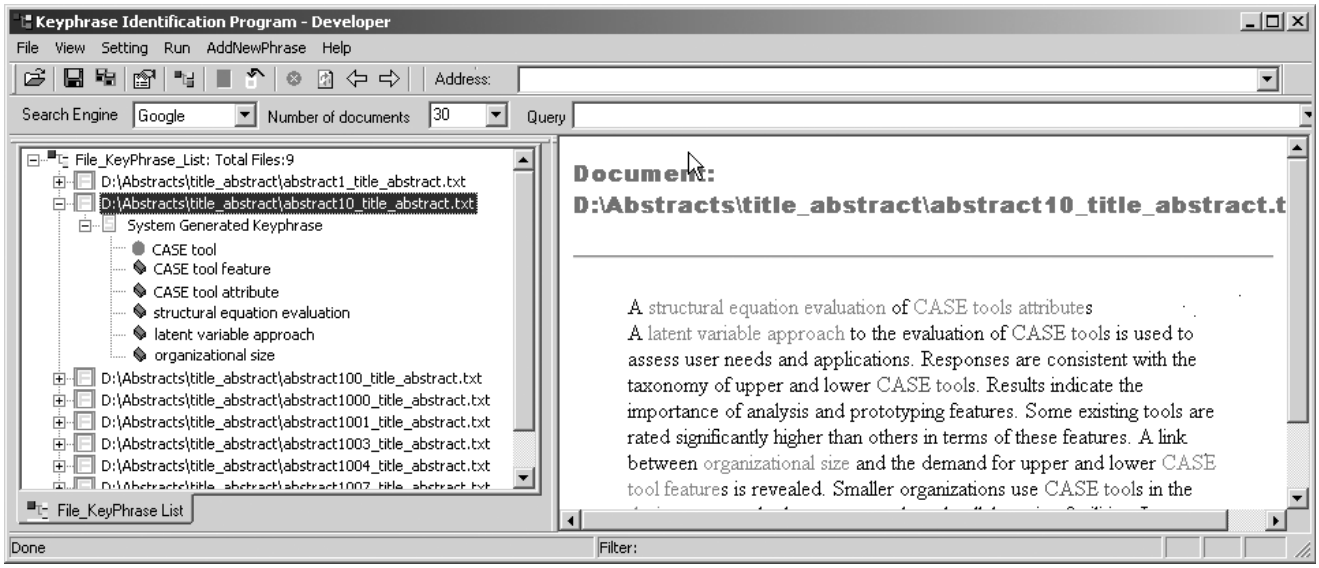

FIG. 1. A screenshot of KIP used to explain the learning function.

Evaluation 3). It will benefit future keyphrase extraction for new documents.

The learning function is especially useful when KIP is used in a domain where there are very few existing domain-specific keyphrases and keywords. When it is applied to such a domain, KIP can automatically learn new keyphrases, and finally build a glossary for this domain.

Figure 1 shows a screenshot of KIP. We use it as an example to explain how the learning function works. In this figure, nine documents are processed, and the file names and the extracted keyphrases are displayed in the left frame. We use one document as an example. KIP extracts six keyphrases for this document, and they are displayed in the left frame. Five of them are marked with a cube in front of each (*CASE tool feature*, *CASE tool attribute*, *structural equation evaluation*, *latent variable approach*, and *organizational size*), and one of them is marked with a pie in front (*CASE tool*). The five keyphrases marked with cubes are new to the glossary database, which means they are not contained in the keyphrase table. The one with a pie is already in the database. The text body of this document is displayed in the right frame. With the automatic learning function enabled, the system will add these five new keyphrases to the database automatically. To better control the quality of the new keyphrases added to the database, KIP has some parameters that allow the user to set inclusion requirements for adding new keyphrases. This option is similar to the procedure for defining the quantity of keyphrases extracted for an individual document. The system also has an option that allows the user to exclude some new keyphrases from being added to the database, if the user thinks they are not qualified.

KIP's Personalization Feature

Personalization is a feature based on KIP's learning function. The learning process can be automatic, without user involvement, and it can also be user-involved. If the automatic option is disabled, the user can decide whether he or she wants a newly identified keyphrase to be added to the database. In this way, the user can control the quality of new keyphrases added to the database. Only the new identified keyphrases that satisfy the user will be added to the database. Another useful feature is that if the user thinks a phrase is good and needs to be added to the database, but it is not identified by the system as a keyphrase, the user can highlight this phrase in the document text. The system will add this phrase to the database automatically.

Let us use Figure 2 to explain how KIP's personalization feature works. In this example, the system extracts six keyphrases, which appear under the category *System Generated Keyphrase*, from the first document. Initially, five of them are new to the glossary database and each is marked with a cube (*user participation*, *information system development*, *information system use*, *system success*, and *theoretical framework*), and one is marked with a pie (*field study*), which means it already exists in the database. The five new keyphrases are normally added to the database. However, suppose the user in this example is not satisfied with the phrase *theoretical framework* and does not want it to be added to the database; the user can exclude this phrase by right-clicking the phrase and choosing the corresponding option from the popup menu. The icon in front of this phrase then changes from a cube to an *X*. If the user does not think a phrase is an appropriate keyphrase for the document, the user can delete this phrase from the phrase list under the category *System Generated Keyphrase*, and this phrase will be removed by the system from the keyphrase list.

In the example in Figure 2, the user thinks that the phrase *user involvement* is a good keyphrase for this document and that it should be added to the glossary database, though it is not extracted by the system. The user can highlight this phrase in the right frame where the document text is displayed, and

DOI: 10.1002/asi

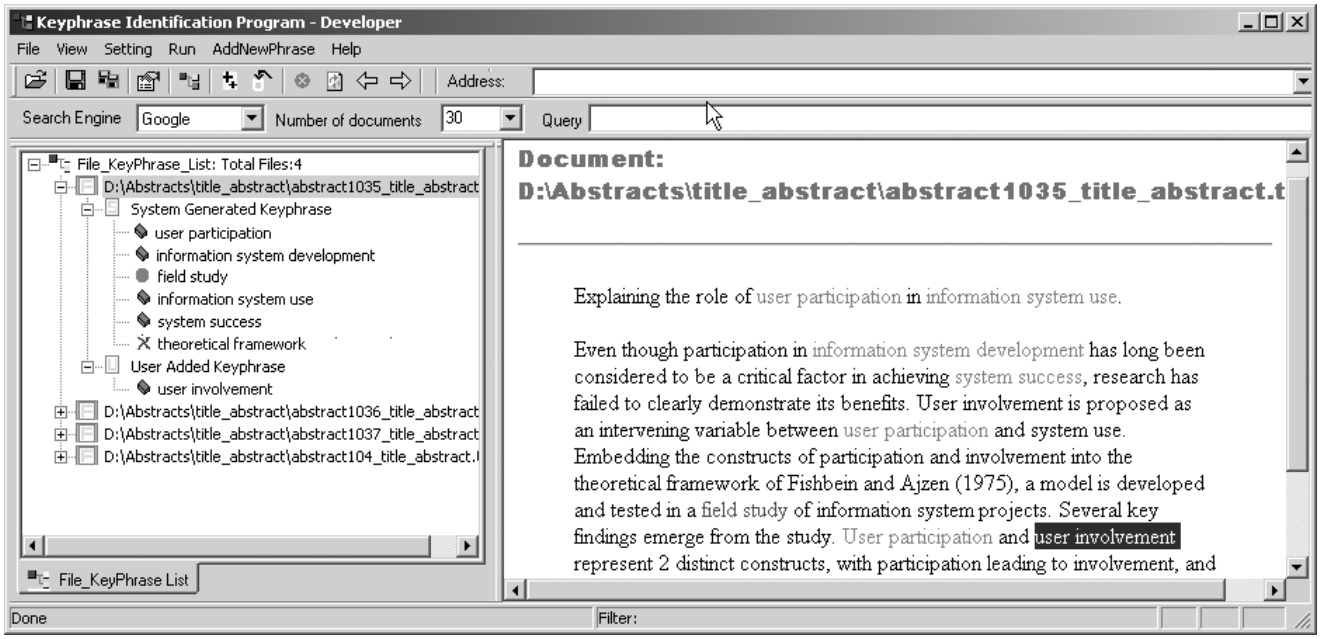

FIG. 2. A screenshot of KIP used to explain the personalization feature.

the phrase will be added to this document's keyphrase list under the category *User Added Keyphrase*. KIP will add this phrase to the glossary database automatically.

By using the features described above, the user can control the quality of the glossary database and the direction of its growth, easily and flexibly. After running KIP for a period of time, the glossary database will be personalized to this user's research area, and KIP will be more effective in identifying keyphrases from documents in the user's interest area. With the same starting glossary database, different users with different research interests within a domain will eventually have different evolved glossary databases, and, as a result, KIP will gradually be more and more effective and personalized for the user. For example, if one user's research area is Human Computer Interaction (HCI), and another user's area is Information Retrieval (IR), after a period of time, even with the same starting glossary database, these two users could gradually build up two different glossary databases independently. Finally, KIP could extract keyphrases effectively for both users. The user may be a single person, a research group, or an organization specializing in a certain area.

Evaluation

KIP has two versions: the base KIP system, which does not contain the learning function, and the KIP system with learning function, called advanced KIP. We conducted three evaluations (hereafter called evaluation 1, evaluation 2, and evaluation 3). In evaluation 1, the standard information retrieval measures, precision and recall, were calculated for base KIP, and KIP was compared to other similar keyphrase extraction systems. Evaluation 2 involved human evaluations of extracted keyphrases. The generated keyphrases were rated by subjects. In evaluation 3, we compared the performance of base KIP and advanced KIP. Both evaluation 1 and evaluation 2 used the base KIP. We used the IS domain to perform the evaluations. The process of building an IS domain database containing domain-specific keyphrases and keywords has been described.

Evaluation 1

Method. In this evaluation, we assessed KIP's effectiveness by computing its precision and recall using author-provided keyphrases for documents. We also wanted to know how well KIP performs, so we compared KIP to two other keyphrase extraction systems, Kea (Frank et al., 1999; Witten et al., 1999) and Extractor (Turney, 2000). Other reported systems were not available to us for comparison. The algorithms used by Kea and Extractor have been described in the literature review section. In this evaluation, we used Kea 1.1.4 with its built-in model, cstr, which gives the best results among all its models (Jones & Paynter, 2002). For Extractor, we used version 7.1. Each of these systems can take a document as input and generate a list of keyphrases for that document. In this experiment, precision means the proportion of the extracted keyphrases that match the keyphrases assigned by a document's 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. Measuring precision and recall against author keyphrases is easy to carry out, and it allows more precise comparison between different keyphrase extraction systems. Previous studies have used this measure and found it is an appropriate method to measure the effectiveness of a keyphrase extraction system (Jones & Paynter, 2002; Turney, 2000; Frank et al, 1999; Tolle & Chen, 2000).

DOI: 10.1002/asi

TABLE 1. Sources of documents used in evaluation 1.

Source of documents	Number of documents
Information Retrieval (2002)	10
Data Mining and Knowledge Discovery (2002)	10
Proceedings of ACM Conference on Human Factors 1997 (CHI'97)	6
Proceedings of Americas Conference on Information Systems 2002	9
Proceedings of Americas Conference on Information Systems 2001	15
All	50

We used 50 papers as the test documents in this evaluation. The sources of the 50 documents are listed in Table 1. We chose documents from different sources to make the evaluation results more generalizable. Although the test set was not large, it was comparable to those of other studies. Jones and Paynter (2002) used six papers to evaluate Kea's precision and recall. Tolle and Chen (2000) tested 10 documents to compare their algorithm to NPtool. Frank et al. (1999) used 20 journal articles and 35 *FIPS* Web pages to compare Kea and Extractor. The 50 papers used here are directly within or in some sense related to the IS domain, so our starting glossary database was appropriate. All 50 papers have author-assigned keywords. (Note: These 50 papers were not used to populate our glossary database.) Listings of author-assigned keyphrases were removed from the papers before they were processed by these three systems. The average length of these papers was 12 pages. The average number of author-assigned keyphrases for the 50 papers was 4.8. We compared the performance of KIP and Kea when the number of keyphrases extracted by them was 3, 6, 9, 12, 15, and 18, respectively. Because the version of Extractor we used could produce at most eight phrases for each document, we compared the performance of Extractor, KIP, and Kea only when the number of extracted keyphrases was 3, 6, and 8, respectively.

Results. Table 2 shows the results for base KIP and Kea. We also tested the statistical significance of the difference in precision between the two systems, as well as their recalls, using a paired *t*-test. From Table 2 we can see that, in respect to precision and recall, KIP performs better than Kea. The results are all significant at 95% ($p < .05$) or 99% ($p < .01$) confidence level.

Because of the reason described above, we compared the performance of KIP and Extractor at only three data points: the number of extracted phrases is 3, 6, and 8, respectively. The results are shown in Table 3.

Table 3 shows that KIP's precision and recall are better than Extractor's, when the number of extracted keyphrases is 3, 6, and 8, at the $p = .05$ level.

Kea and Extractor are compared in Table 4. The results show that there is no significant difference between Kea and Extractor in respect to the precision and recall, when the number of extracted keyphrases is 3, 6, and 8.

In this experiment, we used author-provided keyphrases to calculate the precision and recall. However, we need to point out that some author-provided keyphrases may not

TABLE 2. Precision and recall for base KIP and Kea.

Number of extracted keyphrases	Average precision ± standard deviation		Significance test on precision difference (*p*-value)	Average recall ± standard deviation		Significance test on recall difference (*p*-value)
	Base KIP	Kea		Base KIP	Kea	
3	0.29 ± 0.27	0.18 ± 0.27	= .04	0.21 ± 0.20	0.11 ± 0.16	= .01
6	0.24 ± 0.14	0.18 ± 0.16	= .03	0.33 ± 0.21	0.23 ± 0.20	= .01
9	0.20 ± 0.09	0.14 ± 0.11	< .01	0.41 ± 0.22	0.26 ± 0.20	< .01
12	0.16 ± 0.07	0.12 ± 0.08	< .01	0.45 ± 0.24	0.33 ± 0.20	< .01
15	0.15 ± 0.06	0.10 ± 0.06	< .01	0.50 ± 0.23	0.34 ± 0.21	< .01
18	0.13 ± 0.05	0.09 ± 0.06	< .01	0.54 ± 0.24	0.37 ± 0.23	< .01

TABLE 3. Precision and recall for base KIP and Extractor.

Number of extracted keyphrases	Average precision ± standard deviation		Significance test on precision difference (*p*-value)	Average recall ± standard deviation		Significance test on recall difference (*p*-value)
	Base KIP	Extractor		Base KIP	Extractor	
3	0.29 ± 0.27	0.19 ± 0.23	= .02	0.21 ± 0.20	0.13 ± 0.17	= .01
6	0.24 ± 0.14	0.18 ± 0.15	= .04	0.33 ± 0.21	0.24 ± 0.20	= .02
8	0.21 ± 0.09	0.15 ± 0.11	= .02	0.39 ± 0.21	0.27 ± 0.21	< .01

DOI: 10.1002/asi

TABLE 4. Precision and recall for Kea and Extractor.

Number of extracted keyphrases	Average precision ± standard deviation		Significance test on precision difference (p-value)	Average recall ± standard deviation		Significance test on recall difference (p-value)
	Kea	Extractor		Kea	Extractor	
3	0.18 ± 0.27	0.19 ± 0.23	> .05	0.11 ± 0.16	0.13 ± 0.17	> .05
6	0.18 ± 0.16	0.18 ± 0.15	> .05	0.23 ± 0.20	0.24 ± 0.20	> .05
8	0.15 ± 0.12	0.15 ± 0.11	> .05	0.25 ± 0.21	0.27 ± 0.21	> .05

occur in the document they are assigned to. In experiments reported by Turney (2000), about only 75% of author-provided keyphrases appear somewhere in the document. That means the highest recall for a system could only be 0.75. We will discuss the issues of author-assigned keyphrases further in the discussion section.

Evaluation 2

Method. Several previous studies have used human assessment to evaluate system-generated keyphrases (Turney, 2000; Barker & Cornacchia, 2000; Jones & Paynter, 2002). Human evaluation reflects how human readers feel about the keyphrases when dealing with them in the real world. In this experiment, we used human judges to assess the quality of the keyphrases generated by base KIP. Twenty short papers were used in this evaluation. They were from the AMCIS '01, '02, and '03 Proceedings. The average length of the documents was three pages. The document set used in this experiment was different from the one used in evaluation 1. We changed the evaluation set because this experiment involved human judges, and we wanted the length of test documents to be shorter. We recruited 10 information systems researchers as domain experts. Each expert was given all of the 20 documents and the extracted keyphrase list for each document. They were asked to read a document fully first and then go over the extracted keyphrase list for that document. For each keyphrase, the subject rated the quality of the keyphrase in terms of "how well it represented major issues in that document," using a 5-point scale ranging from 1 to 5, where 1 means *worst*, 5 means *best*, and 3 means *neutral*. In the experiment of human evaluation of keyphrases generated by Extractor, the only instruction given by Turney (2000) to the subjects was "to rate the quality of the keyphrase, choose Good or Bad." In Jones and Paynter's experiment for evaluating keyphrases generated by Kea, the subjects were asked to rate the keyphrases in terms of "How well does each of the following phrases represent what the document is either wholly or partly about?" For each document, KIP extracted 15 keyphrases. To avoid bias the judges did not know, until finishing the experiment, how these phrases were generated and what system they were evaluating. KIP ranks keyphrases in order of their scores. To avoid bias, keyphrases were shuffled before being given to the judges. After judges finished their tasks, the keyphrases were put back to their original order, so we could easily interpret and analyze the results.

Results. The average scores assigned by the judges for the 20 documents when the number of extracted keyphrases was 3, 6, 9, 12, and 15 are shown in Table 5.

The table shows that the mean scores are all greater than the midpoint, 3. So, on average, the keyphrases were rated positively by the subjects.

From Table 5, we can also see that when the number of extracted keyphrases decreases, the mean score increases. This is what we expected, because KIP outputs the keyphrases in descending order of their importance to the document. The results in Table 5 also show that the system is effective in ranking the phrases. We can see this trend from Figure 3. We used a paired t-test to assess the significance of this trend. The result shows that there is a significant difference between any two evaluation points (e.g., when the number of extracted keyphrases is 3 and when it is 6) at the $p < .01$ level. This is especially useful and important when only a limited number of keyphrases are required, because

TABLE 5. Mean scores of base KIP.

	Number of extracted keyphrases				
	3	6	9	12	15
Mean score	3.75	3.61	3.50	3.35	3.26
Standard deviation	0.32	0.27	0.23	0.21	0.20

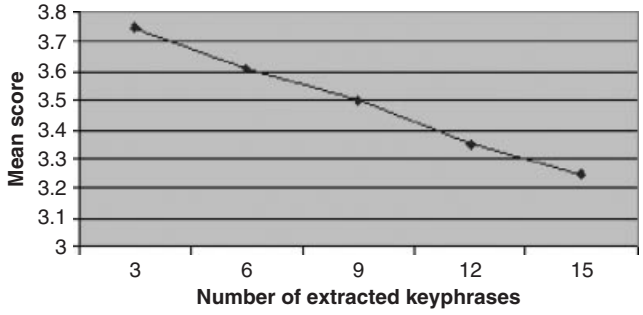

FIG. 3. Mean score for extracted keyphrases.

DOI: 10.1002/asi

we will be confident that the extracted set of keyphrases consists of the best ones among all the candidate keyphrases.

The inter-judge agreement in this experiment is important. The Kappa Statistic *K* (Carletta, 1996) and the Kendall Coefficient of Concordance *W* (Siegel & Castellan, 1988) are two common methods used to measure the inter-judge agreement. *K* method considers agreement on unordered categories; *W* method is good at measuring the agreement between subjects' relative rankings of keyphrases (Jones & Paynter, 2002). So we used *W* to test our subjects' inter-agreement. The value of *W* ranges from 0 to 1. A value of 1 means complete agreement between subjects, and a value of 0 means a chance level of agreement. The average *W* value for all the 20 documents is 0.57, which means a good agreement among subjects.

Evaluation 3

Method. We have discussed KIP's learning function in the earlier section. With this feature enabled, the database will grow gradually, and we hope that the system performance will be improved, too. The purpose of this experiment is to test the effectiveness of the leaning function.

In this evaluation, the precision and recall of base KIP were compared to those of advanced KIP. Initially, the databases used by base KIP and advanced KIP contained the same content. The database of advanced KIP will grow, if it finds and adds new phrases to the database while processing documents. We compared their performance after advanced KIP had processed 150 technical papers and learned about 600 new keyphrases. These 150 papers were chosen from Communications of the ACM (CACM). We used the same 50 papers used in evaluation 1 as the test documents for this evaluation.

Precision and recall of base KIP for the 50 documents had already been calculated in evaluation 1. After advanced KIP had processed the 150 CACM papers and learned new phrases, we used it to extract keyphrases for the same 50 test documents. The precision and recall of advanced KIP were calculated by comparing the keyphrases generated by advanced KIP with author-assigned keyphrases.

Results. Table 6 shows the precision and recall for base KIP and advanced KIP. The results in Table 6 show that advanced KIP is better than the base system when the number of extracted keyphrases is 12 and 18, and the result is statistically significant. However, when this number is 6, the result is not significant at the 95% confidence level. The results illustrate that the KIP's learning function is effective when the number of extracted keyphrases is larger. It should be noted that not only the amount, but also the quality of the new phrases added to the system database will affect the future effectiveness of KIP. We will address this in the discussion section.

This evaluation was based on the fact that the advanced KIP had processed 150 documents and learned about 600 new phrases. We expect that the more new phrases learned, the better its performance.

Discussion

Author Keyphrases

In order to compare our results with those reported in prior studies, and also because there are no large test collections with identified keyphrases available for evaluation, we used author-assigned keyphrases to calculate precision and recall in our evaluations 1 and 3. In prior studies this is the conventional way of measuring precision and recall for extracted keyphrases. It is easier to carry out and is less time-consuming than human evaluations like the one conducted in our evaluation 2. However, using author keyphrases for evaluation has some disadvantages. First, by design, automatic keyphrases are extracted based on their importance to a document, whereas author keyphrases are not always the best phrases to represent the content of documents. The latter sometimes are chosen based on other considerations. *Best phrases* means the most important, topical phrases for a given document. As document keyphrases, they must capture the main topics of the document. There are several reasons that the authors do not always choose the best phrases as the document keyphrases, such as choosing a phrase to increase the document's chance of being searched. This is most commonly seen in Web documents. Second, some author-assigned keyphrases may not occur in the documents they are assigned to. For example, *laptop* and *notebook* can both be chosen as keywords for a document, although the author usually uses only one of them in the document. In this

TABLE 6. Precision and recall of base KIP and advanced KIP.

Number of extracted keyphrases	Average precision ± standard deviation		Significance test on precision difference (p-value)	Average recall ± standard deviation		Significance test on recall difference (p-value)
	Base KIP	Advanced KIP		Base KIP	Advanced KIP	
6	0.24 ± 0.14	0.26 ± 0.15	> .05	0.33 ± 0.21	0.36 ± 0.22	> .05
12	0.16 ± 0.07	0.21 ± 0.10	< .05	0.45 ± 0.24	0.52 ± 0.24	< .05
18	0.13 ± 0.05	0.17 ± 0.08	< .05	0.54 ± 0.24	0.64 ± 0.25	< .05

DOI: 10.1002/asi

case, even if all of the words and phrases are extracted from the document the recall still cannot reach 100%. In experiments reported by Turney (2000), about only 75% of author-provided keyphrases appear in the document. That means the highest recall for a keyphrase extraction system could only be 75%. Finally, of all available documents, only very few of them, mostly academic papers, have author-assigned keyphrases. The above issues are the reasons why we also conducted evaluation 2, which required subject judgments. However, subject assessments require considerable resources and time, because the subjects need to thoroughly read each paper before starting to evaluate the quality of the generated keyphrases.

Quality of Noun Phrases

The quality of identified noun phrases is an important factor that affects the performance of KIP. By checking the extracted keyphrases receiving very low human assessment scores, we found that some of them were not really noun phrases. That is the main reason they received a very low score from judges. It also means that in order to improve the performance of KIP, we need to improve our part-of-speech tagger and noun phrase extractor. The performance of keyphrase extraction depends on the correctness of part-of-speech assignment and noun phrase identification. Fortunately, the precision of our noun phrase extractor was above 95%; there were only very few non-noun phrases extracted.

Phrase Length

Phrase length will also affect the evaluation results of keyphrase extraction. Jones and Paynter (2002) reported that, on average, two-word keyphrases receive the highest human assessment scores. They also report that one-word phrases receive the lowest scores. One reason for the low score of one-word phrases may be the limited context that a one-word phrase conveys, for example, *data* versus *data mining*. KIP can extract phrases of any length between one and six words. To give users more flexibility, the system has options to let users specify the length of keyphrase she or he wants.

According to KIP's algorithm, a phrase's length will affect its weight, but this effect is small. A phrase's weight is mainly decided by what words or subphrases it contains and its frequency in the document. It's weight depends on the importance of the words and subphrases it contains. Our observations also prove this point. However, if two phrases have the same frequency and contain the same important words and subphrases, then the longer one will have a slightly higher weight. For example, for *noun phrase* and *noun phrase chunking*, if their frequencies in the document are the same, then the latter will have a higher score. This is reasonable, because generally, longer phrases give more information, which means they provide more context and are more specific than more general, shorter ones.

The Quality of the Glossary Database

We consider the heart of KIP to be the glossary database containing domain-specific keywords and keyphrases. Even though we claim that KIP is a domain-specific application, the real domain-specific part is actually the initial glossary database. We can make the database a plug-in or a parameter that allows users to switch between different databases. This is useful for users who have a wide range of research interests and have a need to identify keyphrases from different domains of interest. As a result, KIP becomes a domain-independent application.

According to KIP's algorithm, the weight calculation for a candidate keyphrase partly depends on the weights of keyphrases and keywords in the glossary database, so the quality of the database will affect the quality of extracted keyphrases. There are a number of factors affecting the quality of the database. An important one is the comprehensiveness (depth and breadth of coverage) of keyphrases in this database. If a paper is about a very new topic, and the database does not contain any related keyphrases or keywords of this new topic, it is very possible that the score assigned to a potential keyphrase of this new topic will not be high enough for it to be extracted as a keyphrase of this document the first time. However, this is rectified by our learning function and personalization feature. All new phrases have a very high chance of being identified later.

The appropriateness of the database is another issue. As we described earlier, for evaluation 1, KIP only used a database containing manual keyphrases from the IS domain. This made the performance of identifying keyphrases in the information retrieval domain not optimal. However, we see a possibility of using KIP to build a glossary for new domains. An example would be to build a glossary for *text mining* domain using manual keyphrases from *data mining* and *information retrieval* as a starting point. Putting manual keyphrases from those two domains in the database and supplying KIP with a collection of text mining related papers, should enable KIP to gradually generate a text mining glossary.

The Learning Function and Personalization Feature

One issue associated with the learning function is the threshold. We do not intend to have a lot of junk terms added to the database. Therefore, carefully setting a threshold to reject unqualified phrases is important when the system is set to learn new terms automatically without user involvement. The good thing is that KIP has the option of letting the user control the quality of phrases to be added to the system database by manually rejecting unqualified system-generated phrases and adding good phrases unidentified by the system.

We have applied and evaluated KIP in the IS domain. In another study (Li et al., 2004), we proposed a mechanism of enriching the metadata of the returned results of a search engine by incorporating document keyphrases in each returned hit. In the experiment for that study, we applied KIP

DOI: 10.1002/asi

to the politics domain. The overall evaluation of the study was encouraging. By looking at the keyphrases in each returned hit, the user can predict the content of the document more easily and accurately. The experimental results show that our solution may save users time up to 32% and that users would like to use our proposed search interface with document keyphrases as part of the metadata of a returned hit. We will also explore how well advanced KIP performs when it is applied to a new domain that has very few domain-specific keyphrases and keywords.

Conclusion

A new keyphrase extraction algorithm, its learning function, and personalization feature are introduced in this paper. The evaluation results show that KIP performs better than other reported keyphrase extraction algorithms. KIP's learning function and personalization feature make it easier to apply KIP to different domains. Its learning function is effective and does improve system performance. The features and performance of KIP will make it useful for a variety of applications, such as metadata generation for search-returned documents and text mining. Our future research will focus on the application of its personalization feature and on testing KIP in different domains.

References

Anick, P., & Vaithyanathan, S. (1997). Exploiting clustering and phrases for context-based information retrieval. In N.J. Belkin, A.D. Narasimhalu, & P. Willett (Eds.), Proceedings of SIGIR '97: The 20th Annual International Conference on Research and Development in Information Retrieval (pp. 314–323). New York: ACM Press.

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

Barker, K., & Cornacchia, N. (2000). Using noun phrase heads to extract document keyphrases. In H.J. Hamilton (Ed.), Proceedings of the Thirteenth Canadian Conference on Artificial Intelligence (pp. 40–52). London: Springer-Verlag.

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). Menlo Park: CA: American Association for Artificial Intelligence Press.

Carletta, J. (1996). Assessing agreement on classification tasks: The kappa statistic. Computational Linguistics, 22(2), 249–254.

Cover, T.M., & Thomas, J.A . (1991). Elements of information theory. New York: John Wiley.

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.), Proceedings of SIGIR'91: The 14th Annual International Conference on Research and Development in Information Retrieval (pp. 32–45). New York: ACM Press.

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

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

Frank, E., Paynter, G., Witten, I.H., Gutwin, C., & Nevill-Manning, C. (1999). Domain-specific keyphrase extraction. In Thomas Dean (Ed.), Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (pp. 668–673). San Francisco: Morgan Kaufmann.

Gutwin, C., Paynter, G., Witten, I.H., Nevill-Manning, C., & Frank, E. (1999). 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 Proceedings of the Third International Asian Conference on Digital Libraries (pp. 113–120).

Jones, S., & Paynter, G.W. (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 F. Gey, M. Herst, & R. Tong (Eds.), Proceedings of SIGIR'99: The 22nd International Conference on Research and Development in Information Retrieval (pp. 160–167). New York: ACM Press.

Kamp, H.A. (1981). Theory of truth and semantic representation. In J. Groenendijk, T. Janssen, & M. Stokhof (Eds.), Formal methods in the study of language (pp. 277–322). Amsterdam, Holland: Mathematische Centrum.

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

Krulwich, B., & Burkey, C. (1996). Learning user information interests through the extraction of semantically significant phrases. In M. Hearst & H. Hirsh (Eds.), AAAI 1996 Spring Symposium on Machine Learning in Information Access (pp. 15–18). Menlo Park, CA: American Association for Artificial Intelligence Press.

Larkey, L.S. (1999). A patent search and classification system. In Proceedings of Digital Libraries '99: The Fourth ACM Conference on Digital Libraries (pp. 179–187). New York: ACM Press.

Li, Q., Wu, Y.B., Bot, R.S., & Chen, X. (2004). Incorporating document keyphrases in search results. In J. Luftman (Ed.), Proceedings of the Tenth Americas Conference on Information Systems (pp. 3255–3263). Waco, TX: Reagan Ramsower.

Muñoz, M., Punyakanok, V., Roth, D., & Zimak, D. (1999). A learning approach to shallow parsing. In F. Pascale (Ed.), Proceedings of EMNLP/WVLC-99 (pp. 168–178). Cambridge, MA: MIT Press.

Ramshaw L.A., & Marcus, M.P. (1995). Text chunking using transformation-based learning. In D. Yarovsky & K. Church (Eds.), Proceedings of the Third Workshop on Very Large Corpora (pp. 82–94). Somerset, NJ: Association for Computational Linguistics.

Sang, E.F. (2000). Noun phrase representation by system combination. In A. Bosch & H. Wiegaard (Eds.), Proceedings of ANLP-NAACL 2000 (pp. 50–55). San Francisco: Morgan Kaufmann.

Siegel, S., & Castellan, N.J. (1988). Nonparametric statistics for the behavioral sciences (2nd ed.). New York: McGraw Hill College Division.

Snow, C.E., & Ferguson, C.A. (1997). Talking to children: Language input and acquisition. Cambridge: Cambridge University Press.

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.

Tomokiyo, T., & Hurst, M. (2003). A language model approach to keyphrase extraction. In F. Bond, A. Korhonen, D. McCarthy, & A. Villavicencio (Eds.), Proceedings of the Association for Computational Linguistics (ACL) 2003 Workshop on Multiword Expressions; Analysis, Acquisition and Treatment (pp. 33–40). Cambridge, MA: MIT Press.

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

Witten, I.H., Paynter, G.W., Frank, E., Gutwin, C., & Nevill-Manning, C.G. (1999). In N. Rowe & E. A. Fox (Eds.), KEA: Practical Automatic Keyphrase Extraction. Proceedings of the Fourth ACM Conference on Digital Libraries (pp. 254–255). New York: ACM Press.

Zha, H. (2002). Generic summarization and keyphrase extraction using mutual reinforcement principle and sentence clustering. In K. Järvelin (Ed.), Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 113–120). New York: ACM Press.

DOI: 10.1002/asi