ELSEVIER
Available online at www.sciencedirect.com
ScienceDirect
Journal of Biomedical Informatics 39 (2006) 668–679
Journal of Biomedical Informatics
www.elsevier.com/locate/yjbin

Identifying important concepts from medical documents

Quanzhi Li *, Yi-Fang Brook Wu

Information Systems Department, New Jersey Institute of Technology, Newark, NJ 07102, USA

Received 8 September 2005
Available online 2 March 2006

Abstract

Automated medical concept recognition is important for medical informatics such as medical document retrieval and text mining research. In this paper, we present a software tool called keyphrase identification program (KIP) for identifying topical concepts from medical documents. KIP combines two functions: noun phrase extraction and keyphrase identification. The former automatically extracts noun phrases from medical literature as keyphrase candidates. The latter assigns weights to extracted noun phrases for a medical document based on how important they are to that document and how domain specific they are in the medical domain. The experimental results show that our noun phrase extractor is effective in identifying noun phrases from medical documents, so is the keyphrase extractor in identifying important medical conceptual terms. They both performed better than the systems they were compared to.

Keywords: Noun phrase extraction; Keyphrase extraction; Medical documents; Medical concepts; Document keyphrase; Text mining

1. Introduction

The pervasion of medical information via the WWW has created a continuously growing need for the development of techniques for discovering, accessing, and sharing knowledge from medical literature. In recent years, there have been a remarkable number of studies in discovering various kinds of knowledge by mining the medical literature, such as studies on protein–protein interactions [1–7] and relations between drugs, genes, and cells [8–10]. In these applications, term identification is the most crucial step for accessing information stored in documents [11]. Terms (e.g., names of proteins, genes, gene products, organisms, and drugs, etc.) are usually used to identify important document concepts. Concepts in textual documents are usually described by noun phrases, and noun phrases carry the primary information of documents. Since the vast majority of concept terms are noun phrases, noun phrase identification becomes one of the fundamental problems for many applications in mining medical documents. Evidences have shown that noun phrases help readers understand, organize, access, and share information of a document.

Keyphrase identification in medical documents, which is more advanced, has been a challenging research topic in recent years, because, as opposed to noun phrases, keyphrases are more domain related and more selective. The most important topical terms for a document are usually referred to as "keyphrases." Document keyphrases provide a concise summary of a document's content, offering semantic metadata summarizing and characterizing a document. Previous studies have shown that document keyphrases can be used in a variety of applications, such as retrieval engine [12,13] and browsing interface [14]. For example, they may be utilized to enrich the metadata of the results returned from a search engine [12]. They may also be used to efficiently classify or cluster documents into different categories [15]. In this paper, we will introduce two algorithms (systems): a noun phrase extraction algorithm and a keyphrase identification algorithm.

The first system is a noun phrase extractor. The main differences between our noun phrase extractor and other systems are that our system uses a lexicon database which integrates WordNet lexical database [16] and SPECIALIST Lexicon (http://umlsks.nlm.nih.gov/) for the part-of-

* Corresponding author. Fax: +1 973 596 5777.
E-mail addresses: QL23@njit.edu (Q. Li), wu@njit.edu (Y.-F.B. Wu).

1532-0464/$ - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.jbi.2006.02.001

speech tagging, and our system does not need any training data. One limitation with a noun phrase extractor is that it extracts all the noun phrases in a medical document, which might be too general to be useful in medical text mining. If we integrate the domain knowledge and the characteristics of a document with the noun phrase extractor, we can extract the concepts which are semantically relevant to the main topical theme of the document. This is the goal of our second system, which is built on top of the noun phrase extractor and is called keyphrase identification program (KIP) [17]. After all the noun phrases of a document are extracted, KIP ranks all the noun phrases in terms of their degree of relevance to the main theme of the document, and select only the important ones. KIP fulfills this task by considering the domain of a document and the characteristics of the extracted noun phrases. To extract keyphrases from documents, KIP's 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 pre-identified medical terms, to calculate scores of noun phrases in a document. The noun phrases having higher scores will be extracted as keyphrases. In this study, the glossary database is built from Medical Subject Headings (MeSH), which is NLM's controlled vocabulary thesaurus and consists of a lot of medical terms in a hierarchical structure.

In the following sections, previous studies on noun phrase extraction, keyphrase extraction, and their applications in medical domain are discussed first. Then we describe our noun phrase extractor and its performance in medical domain. Finally, we present the algorithm of our keyphrase identification program and its evaluation based on medical documents.

2. Previous studies

In this section, we first present previous studies about the applications and identification of noun phrases and then we describe the related research on applications and identification of document keyphrases.

2.1. Noun phrase applications and identification in medical domain

Noun phrases have been used in various applications in medical domain and other domains. Many studies pertaining applications of noun phrases focus on retrieval system and browsing interface [18–23]; some others explore their applications on document classification and clustering [24,25]. Croft et al. [18] propose a method where phrases identified in natural language queries are used to build structured queries for a probabilistic retrieval model. Their experimental results show that retrieval performance can be improved by using phrases this way, and phrases extracted automatically from a natural language query perform nearly as well as manually selected phrases.

Many medical document analysis or retrieval studies have used documents from MEDLINE, the premier bibliographic database of NLM. Blake and Pratt [26] use noun phrase as the concept terms to detect the connections among medical literature. Their study based on MEDLINE shows that using noun phrases would be more effective in finding complementary literature than using single words. Kumar et al. [27] describe an approach, called BioMap, of using noun phrases extracted from the abstracts of MEDLINE to build a knowledge base for medical literature.

Researchers have developed programs to map free medical text to a biomedical knowledge source. Among them, MetaMap, developed by Aronson at the NLM, is a program that maps medical text to the UMLS Metathesaurus to discover Metathesaurus concepts referred to in the text [28]. It finds Metathesaurus concepts in five steps: (1) parsing—the SPECIALIST minimal commitment parser [29] is used to parse the text into noun phrases; (2) variant generation—variants are generated for each phrase using the SPECIALIST lexicon and a supplementary synonyms database; (3) candidate retrieval—the candidate set of all Metathesaurus strings containing at least one of the variants is retrieved; (4) candidate evaluation—each Metathesaurus candidate is evaluated against the input text by first computing a mapping from the phrase words to the candidate's words and then calculating the strength of the mapping using a linguistically principled evaluation; (5) mapping construction—complete mappings are constructed by combining candidates involved in disjoint parts of the phrase, and the strength of the complete mappings is computed just as for candidate mappings. The noun phrase parser used in step 1 is primarily a barrier category parser, relying on parts of speech that have been already assigned to determine the beginnings and endings of phrases. This noun phrase extraction method is also the basis of the SPECIALIST NLP Text Tools' Noun Phrase Parser (http://specialist.nlm.nih.gov/). We compared the performance of our noun phrase extractor and the SPECIALIST NLP Text Tools' Noun Phrase Extraction Parser in Section 3.2.1.

One application of the MetaMap is the Indexing Initiative System Project (IND) at NLM [21]. The objective of IND is to investigate methods whereby automated indexing methods partially or completely substitute for NIW's manual subject indexing. The IND system consists of software for applying alternative methods of discovering MeSH headings for citation titles and abstracts and then combining them into an ordered list of recommended indexing terms. MetaMap indexing is one of the methods used by ISD to create a list of indexing terms.

Another program that can automatically map medical text to standardized coding system, such UMLS, is MedLEE [30]. MedLEE's goal is to extract, structure, and encode clinical information in textual patient reports so that the data can be used by subsequent automated processes. It uses NLP techniques to generate structured encoded output consisting of findings and corresponding modifiers. The method attempts to find the most granular corresponding code by matching the structured output, which is generated as a result of parsing the sentences, with structures in a coding table in which the structures have been associated with codes. A code is obtained by successfully matching a finding along with modifiers, based on an assumption that a match consisting of a finding with the most modifiers is preferable to a match consisting of the same finding with fewer modifiers because it is the most specific.

In [22], Nadkarni et al. explore the feasibility of using UMLS Metathesaurus as the basis to identify concepts in medical text for indexing. They use a commercial phrase-identification program, which is not introduced in detail in the paper, to identify concepts from medical text. Then they search the UMLS Metathesaurus to find the matches between the identified concepts and the Metathausurus entries. The matched concepts are used as indexing terms. This method is similar to MetaMap [21,28].

Johnson develops a semantic lexicon by matching the words and phrases in the SPECIALIST Lexicon against strings in the 1997 Metathesaurus of UMLS [31]. In the developed semantic lexicon, each word or phrase is associated with one or more syntactic types. Each of the syntactic types can have one or more semantic types. The resulted semantic lexicon can be used to assign semantic types to words or phrases occurring in medical text.

Bodenreider et al. [32] describe an approach to automatically extend UMLS Metathesaurus concepts. The proposed approach is to compare phrases extracted from MEDLINE to current UMLS phrases. They capitalize on differences in modification structure between the MEDLINE phrase and the UMLS phrase to determine the candidates for inclusion in the Metathesaurus. The crucial difference is between a phrase containing adjectival modification and a similar phrase "demodified" by removing its adjectives. A phrase from MEDLINE becomes a candidate concept in the Metathesaurus if the following two requirements are met: (1) a demodified term created from this phrase is found in the terminology; and (2) similarly modified terms exist in the terminology, for a given semantic category. To extract noun phrases, they use PhraseX, a program that extracts noun phrases from text such as MEDLINE abstracts. It does so by referring to the syntactic structure provided by the SPECIALIST minimal commitment parser [33,29], which relies on the SPECIALIST lexicon as well as the XEROX stochastic tagger to resolve part-of-speech ambiguity [34]. The notion of barrier words within sentences is then used to delimit phrase boundaries.

Other examples of studies related to the applications of noun phrases are: protein name identification and protein structure [16,35], protein–protein interactions [1–7], and relations between drugs, genes, and cells [8–10,36].

Besides the methods mentioned in above studies, several other noun phrase extraction techniques have been introduced in previous studies. Some of them are described below.

Chen et al. [37] develop a noun phrase extraction system called FastNPE. It mainly relies on concatenation of adjacent tokens to identify phrases. Later, they revise the above system and develope a new system, which is called AZ Phraser [38]. AZ Phraiser's part-of-speech tagging is based on earlier work of Brill [39]. Their tagger is divided into two main phases of operation—lexical analysis and contextual analysis. The lexicon mostly comprises the Wall Street Journal corpus and the Brown corpus. The contextual analysis uses several contextual rules. The contextual analysis phase is to ensure that the part-of-speech tags are disambiguated. NPtool [40] is a commercial noun phrase extraction program. After preprocessing the documents, it has the following three steps: morphological analysis, constraint grammar parsing, and NP-hostile and friendly finite state parsing and NP extraction. Majoros et al. [41] describe a method of improving the quality of automatically extracted noun phrases by employing prior knowledge during the Hidden Markov Model training procedure for the tagger. When combined with appropriate training data, this enhancement can improve the quality and relevance of the extracted phrases. Huang et al. [42] describe a noun phrase identification module that is composed of a sentence boundary detector, a statistical natural language parser (based on the Maximum Entropy Modeling) trained on a non-medical domain, and a noun phrase tagger.

The main differences between our noun phrase extractor and other systems are that our system uses a lexicon database which integrates the WordNet lexical database [16] and SPECIALIST Lexicon (http://umlsks.nlm.nih.gov/) for the part-of-speech tagging, and our system does not need any training data.

2.2. Keyphrase applications and identification for medical documents

Previous studies have shown that document keyphrases can be used in a variety of applications, such as retrieval engines [12,43], browsing interfaces [14], thesaurus construction [44], and document classification and clustering [15]. In the remaining of this section, we review several well-known automatic keyphrase extraction techniques proposed in previous studies.

Krulwich and Burkey [45] use some heuristics 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 the different formatting structures. This approach is not difficult to implement, but the limitation is that not every document has explicit structural features.

Zha [46] proposes 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 a document 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.

Turney [47] is the first person who treats the problem of keyphrase extraction as supervised learning from examples. Turney uses nine features to score a candidate phrase; some 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's program is called Extractor.

Kea, a keyphrase extraction program developed by Frank et al. [48], uses a machine learning algorithm which is based on naïve Bayes' decision rule. It has some pre-built models. A model is used to identify the keyphrases from 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, which contain phrase frequencies and stopped words.

Both Kea and Extractor use a similar way to identify a candidate keyphrase: the input text is split up according to phrase boundaries (numbers, punctuation marks, dashes, and brackets); non-alphanumeric characters and all numbers are deleted; then a phrase is defined as a sequence of one, two, or three words that appear consecutively in the text; finally, it eliminates those phrases beginning or ending with a stopped word. Kea and Extractor both use supervised machine learning approaches. They all need training corpora to train their programs. For each document in the corpus, there must be a target set of keyphrases provided by authors or generated by experts. The above approach to identifying candidate keyphrases and keyphrases is different from ours, which is described in detail in Section 4.

We have not seen any studies about how the above mentioned keyphrase extraction methods perform in the medical domain. In this study, we explore how well KIP performs when it is applied in the medical domain, and how it performs compared to other systems.

3. Extracting noun phrases form medical documents

In this section, we first describe the algorithm of our noun phrase extractor (NPE). Then we present two experiments used to evaluate its performance with medical documents.

3.1. Noun phrase extractor

The noun phrase extractor has two main components: a part-of-speech tagger and a noun phrase extraction component.

3.1.1. Part-of-speech tagger

In the following paragraphs, we first introduce why we need our own part-of-speech tagger, and then we describe how our tagger works.

Part-of-speech (POS) tagging is a starting point of processing textual information, such as identifying documents' main concepts or gene and protein names. There are several existing POS taggers in medical domain, such as MedPost [49] and the POS tagger developed by Tanabe and Wilbur [50]. But most of them are specializing mainly on processing MEDLINE citations and abstractions. They are not generalized enough to process general medical documents or have other limitations. For example, the main purpose of Tanabe and Wilbur's POS tagger is to extract gene and protein names from MEDLINE. It first applies generated rules from Brill POS tagger [39] to extract single gene and protein names, and then the results are filtered using manually generated rules based on MEDLINE. MedPost is based on the hidden markov model. It currently only accepts text in either MEDLINE format or XML. One main purpose of this study is to develop a system which can automatically identify important topical concepts (called document keyphrases) for medical documents. Document keyphrases provide a concise summary of a document's content, offering semantic metadata summarizing and characterizing a document. The medical documents can be MEDLINE abstracts or some more general medical documents, such as a paper from journal of Biomedical Informatics. Although a medical paper is talking about topics related to medical domain, its main concepts, such as the keywords provided by the authors, may not necessarily be medical terms, such as gene or protein names. For example, if a paper is talking about NLP techniques applying in medical domain, it is very possible that some of its main concepts (keyphrases) may be related to NLP, not the medical domain, like the keywords we assigned for this paper. Because the keyphrases of a medical document may or may not be medical terms, to identify keyphrases for medical documents, we need a POS tagger and noun phrase extractor (NPE) who can work well with not only the medical terms but also the general terms not in medical domain. This is why we want to develop our own POS tagger and NPE. Other available existing ones are either too specialized to gene or protein name identification, such as Tanabe and Wilbur's, or have some other limitations, such as MedPost.

Our tagger uses a lexical database. To cover both the medical terms and the general terms, we integrate the WordNet lexical database [16] and the UMLS SPECIALIST Lexicon. The integration information is shown in Table 1. In our POS tagger, the main word categories considered are noun, verb, adjective, and adverb. WordNet has 151,692 entries for these four categories. SPECIALIST Lexicon has 256,476 entities for these four categories; among the 256,476 entries, 148,589 of them are single words. Because the tagger is going to assign a tag to each single word, only the single words from SPECIALIST Lexicon are combined with WordNet. There are 49,773 overlapped terms between WordNet and SPECIALIST Lexicon single words. The combined database has a total of 250,508 entries. Among them, 98,816 terms are from just

Table 1
Number of terms in the combined lexical database

Number of terms from WordNet	Number of terms from SPECIALIST Lexicon	Number of overlapped terms between these two sources	Number of terms from only WordNet	Number of terms from only SPECIALIST Lexicon	Total number of terms in the combined lexical database
151,692	148,589	49,773	101,819	98,816	250,508

SPECIALIST Lexicon, 101,819 terms are from just WordNet, and only 49,773 terms are from both. This means we have two-third more terms by combining them together than just using any single one of them.

Our POS tagger works as follows. A document is first parsed into sentences after being loaded into the system. Then all the sentences are tokenized to obtain the atom units, each of which could be a punctuation mark or a word. Each word is assigned with an initial part-of-speech (POS) tag. To assign the right tag, we use the integrated lexical database mentioned above, which contains words divided into four categories (noun, verb, adjective, and adverb) and the number of senses of each word used in the categories it belongs to. If a word is found in more than one category, it is marked as a multi-tag word. The initial POS tag for a word is determined by the category having the maximum number of senses of this word. The next step is multi-tag disambiguation. For every multi-tag word, the sequence of the POS tags of the proceeding n tokens (n ranges from 2 to 4) is examined against a list of predefined syntactic rules. For example, "hit" can be either a noun or a verb. If the proceeding word is a determiner (the, a, this, etc.), it will be tagged as a noun rather than a verb and the multi-tag mark is removed. If a word is not found in any of the categories and its POS tag cannot be solved by the syntactic rules, some heuristics are used to determine its POS tag. For instance, if a word is not found in the lexical database, but ends with "tion," it is tagged as a noun.

We evaluated our POS tagger's performance in the medical domain using 10 MEDLINE abstracts. 2,081 words were identified from these 10 documents. We only evaluated noun, verb, adjective, and adverb, which were the main four categories used in our POS tagger. 1415 out of the 2081 words were identified as one kind of the four categories. Among them, 1387 words were correctly identified, resulting in an accuracy rate of 98%.

3.1.2. Noun phrase extraction

People mostly use noun phrase as concept terms. In general, a noun phrase means a sequence of words that usually gives us very useful information. After tagging the text, our NPE extracts noun phrases by selecting the sequence of POS tags that are of interests. The current sequence pattern is defined as [A]N, where A refers to Adjective, N refers to Noun, [ ] means optional, and { } means repetition. A set of exceptional rules is used as well. The system has a system parameter to set the minimum and maximum numbers of words of a noun phrase. By changing the parameter value, users can get noun phrases with different length. The system can extract noun phrases with length of one word to eight words.

A screenshot of the NPE is shown in Fig. 1. The extracted noun phrases are displayed in the left frame, and the related paragraph is displayed in the right frame. The generated noun phrases are also automatically sent to a file, with related information, like the phrase frequency in a document. This program also has other functions, but in

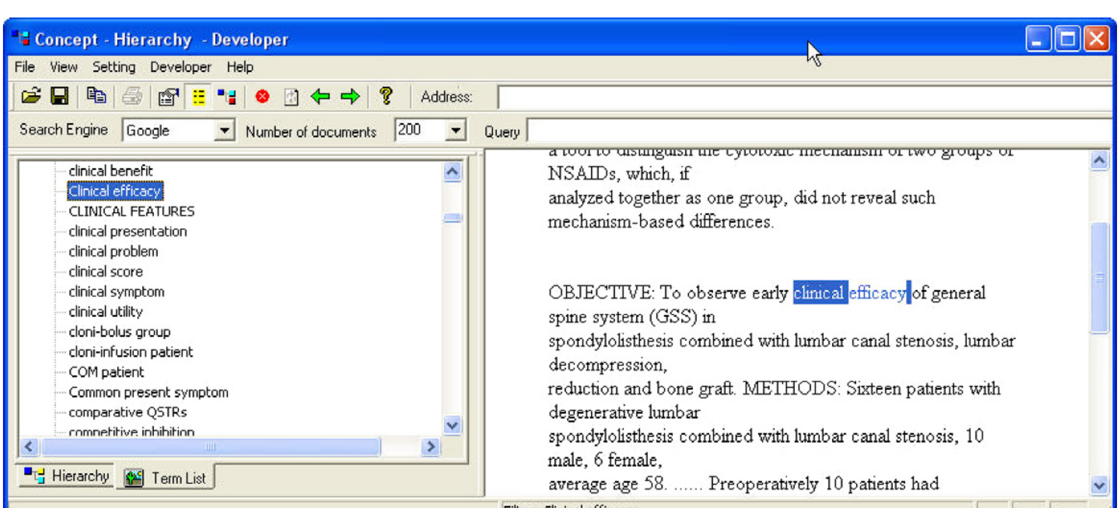

Fig. 1. A screenshot of the noun phrase extractor.

this paper we just describe the function of extracting noun phrases from documents.

3.2. Experiments

We conducted two experiments to evaluate our NPE's performance in medical domain. In Experiment 1, we calculated the precision and recall based on a small document collection, which contained 60 medical documents. In Experiment 2, we computed only the precision based on 1000 medical documents.

3.2.1. Experiment 1

We assessed the effectiveness of the system by computing its precision and recall in this experiment. Precision is the number of noun phrases *correctly* identified by the NPE, divided by the total number of system-identified noun phrases. Recall is defined as the number of noun phrases *correctly* identified by the system, divided by the total number of noun phrases in the documents. To find out the number of noun phrases *correctly* identified by our NPE, human experts are needed to examine the system output. For recall, to find out the total number of noun phrases in the documents, human experts have to be called upon to identify them from documents manually. Many previous studies have used precision and recall to evaluate the performance of noun phrase extraction systems [38,51].

In this experiment, two medical professionals were recruited to identify the noun phrases from our test documents. The test documents were collected from the website of National Center for Biotechnology Information (NCBI) of NLM. *Entrez* is an integrated, text-based search and retrieval system used at NCBI for the major medical databases. We used *Entrez* to collect the test documents. To calculate the recall, all the noun phrases of the test documents should be manually identified in advance. Because manually identifying noun phrases from documents is time-consuming, in this experiment we used a small test collection containing only 60 documents. Each of the two experts was asked to identify all the simple noun phrases for all these 60 documents. Simple noun phrases are less complex noun phrases. A simple noun phrase is a noun phrase without relative clauses, and its head is the rightmost element and thus it has no right modification [21]. Many noun phrase identification programs only identify simple noun phrases [21,42]. Simple noun phrases contain only adjectives and nouns. There are no prepositions in a simple noun phrase. For example, "important concepts" is a simple noun phrase, but "important concepts from the corpus" is not. The steps of collecting the test documents are as follows: first we performed a search using the query "pain relief" at *Entrez*, and 18,937 hits were returned; among the returned hits, we randomly selected 60 documents. The 60 documents were pre-processed, so that each of them contained only the abstract and title.

From the 60 documents, expert 1 identified 4078 noun phrases, while expert 2 identified 4050 noun phrases. They agreed on 3871 noun phrases, which means a 95% agreement rate. Because the experts were asked to identify only simple noun phrases and it was very straightforward, the high agreement rate was what we expected. In similar previous studies [42,51], only one expert was used to identify noun phrase. For the noun phrases they disagreed with, a third expert was asked to decide if they were simple noun phrases. Among the phrases the two experts disagreed on, 140 of them were identified as simple noun phrases by the third expert. Finally, we had 4011 (3871 + 140) noun phrases identified as our ground truth.

We also wanted to know how our NPE performed in medical domain compared to other noun phrase extraction. Several noun phrase extractors have been mentioned in previous studies, like FastNPE, NPTool, Chopper, and AZ Phraser in [38,40], and the one described in [42], but we could not be able to get these programs due to various reasons. Some of them are unavailable for download, such as FastNPE, Chopper, and some others are not really a single useable program, like the method described in [42], which requires many manual steps that will make the result too subjective. But we did find a one that could be used to compare to our NPE. It is from the set of SPECIALIST NLP tools developed by the Lexical Systems Group of The Lister Hill National Center for Biomedical Communications (http://specialist.nlm.nih.gov/). This noun phrase extractor is called SPECIALIST NLP Text Tools' Noun Phrase Parser. We have described this parser in Section 2.1 [28,21]. This parser is primarily a barrier category parser, relying on parts of speech that have been already assigned to determine the beginnings and endings of phrases. For example, determiners and prepositions always indicate the beginning of a phrase. There are several studies involving noun phrase identification are based on this parser or methods similar to the one used by this parser [32,23,28]. This parser identifies the simple noun phrases and prep-phrases. A prep-phrase is a simple noun phrase with one or more prepositions in front of it, such as "to New York City." Because a prep-phrase is also a noun phrase after removing the leading preposition, we considered the prep-phrase also as a noun phrase when measuring the performance of SPECIALIST Text Tools' Noun Phrase Parser [21].

All the 60 documents were processed by our NPE and the SPECIALIST Parser. Noun phrases were identified by these two systems. The results are shown in Table 2.

There is usually a trade-off between precision and recall, and either of them alone does not paint a complete picture of system effectiveness. Therefore, the F measure was invented to show the combined results. The formula for F is: $F = 2 \times \text{precision} \times \text{recall}/(\text{recall} + \text{precision})$. From Table 2 we can see that our NPE performed better than the SPECIALIST Text Tools Parser in identifying noun phrases. Significant tests on the difference between two corresponding proportions (e.g., our NPE's precision to SPECIALIST Text Tools Parser's precision) were also conducted. The results were significant for all the three

Table 2
Precision and recall of SPECIALIST Text Tools' noun phrase parser and our noun phrase extractor

System	Total number of noun phrases identified by human experts	Total number of noun phrases extracted by the noun phrase extractor	Total number of noun phrases correctly identified by the noun phrase extractor	Precision	Recall	*F* measure
SPECIALIST text Tools' noun phrase parser	4011	3912	3598	0.920	0.897	0.908
Our noun phrase extractor		3888	3813	0.981	0.951	0.965
Significant test on the difference between these two systems (*p* value)	NA	NA	NA	<0.01	<0.01	<0.01

measures (precision, recall, and *F* measure) at $p = 0.01$ level.

Huang et al. [42] describe a noun phrase identification module which is composed of a sentence boundary detector, a statistical natural language parser trained on a non-medical domain, and a noun phrase tagger. They also used UMLS Specialist Lexicon to augment their program. In their experiment, their test set was 50 randomly selected clinical radiology reports in Health Level 7 Clinical Document Architecture compatible format. Their overall noun phrase identification precision and recall were 78.9 and 81.5% before using the UMLS Specialist Lexicon and 82.1 and 84.6% after. To compare the performance of the three noun phrase extractors (FastNPE, Chopper, and AZ Phraser), Bennett et al. [38] did an experiment based on 40 medical documents abstracts. The reported results are as follows: for FastNPE, the precision was 0.80 and the recall was 0.50; for Chopper, the precision was 0.90 and the recall was 0.97; and for AZ Phraser, the precision was 0.86 and the recall was 0.92. The *F* values for FastNPE, Chopper, and AZ Phraser were 0.62, 0.93, and 0.88, respectively.

Because these systems are not available for us to do a direct comparison with our NPE, we do not directly compare our system to those systems. We just report their results here for reference.

3.2.2. Experiment 2

Due to the cost for manually identifying noun phrases from documents for calculating recall, we used only 60 documents in Experiment 1. Compared to computing recall, computing precision requires less human effort, since it does not require manually identifying noun phrases from documents, but from system output. So, it is possible to use a larger document collection to compute the precision. In this experiment, we used 1000 documents to calculate the precision of our NPE. Each of the 1000 documents contained only the abstract and title. They were randomly selected from MEDLINE. The result is shown in Table 3. the precision was 98.2%, which is close to the result of Experiment 1, 98.1%. It also shows that the performance of our program is consistent when dealing with medical documents. Based on the result, we may say that, when being applied to medical documents, the noun phrases identified by our noun phrase extractor are of high precision.

4. Extracting keyphrases from medical documents

In the last section, we have discussed our NPE and its evaluation using medical documents. One limitation with a noun phrase extractor is that it extracts all the noun phrases in the documents, which might be too general to be useful in medical text mining. If we integrate the domain knowledge and the characteristics of the document with the noun phrase extractor, we can extract the concepts which are semantically relevant to the main topical theme of a document. This is the goal of our second system, which is built on top of the noun phrase extractor and is called keyphrase identification program (KIP). In the following sections, we describe KIP's algorithm first, then we present its evaluation.

4.1. KIP's algorithm

KIP is a domain-specific keyphrase extraction program, not a keyphrase assignment program, which means the generated keyphrases must occur in the document text. KIP is designed by mimicking "learning by example" that humans do when they learn new things. Identifying things in the environment which is already in our minds is easy. However, learning to identify new things needs to rely on what we already know before. Besides the input documents, from which keyphrases will be extracted, KIP requires a database which is similar to the domain knowledge of the input documents. When KIP examines a key-

Table 3
Precision of the noun phrase extractor based on 1000 medical documents

Number of documents	Total number of noun phrases extracted by the noun phrase extractor	Total number of noun phrases correctly identified by the noun phrase extractor	Precision
1000	75,174	73,823	0.982

phrase candidate, it looks for characteristics (words and sub-phrases) in the phrase which are already in the background knowledge base and assigns weights (how domain-specific a word or a sub-phrase is) accordingly.

After the NPE has extracted all the noun phrases from a document, KIP will assign scores to these phrases, rank them, and extract the ones with higher scores. Its algorithm is based on the logic that a noun phrase containing domain-specific keywords and/or keyphrases is likely to be a keyphrase of the document. 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 a glossary database, which is used to calculate scores of noun phrases. Here a keyword means a single term word, and a 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 up to eight words long. KIP operations can be summarized as follows. KIP first gets a list of keyphrase candidates, which are noun phrases generated by the NPE. Then it examines the composition of a keyphrase 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, readily available pre-identified domain-specific keyphrases are parsed to form a glossary database. Finally, the noun phrases with higher scores are selected as keyphrases of the document.

To calculate the scores for noun phrases, we use a glossary database containing domain-specific keyphrases and keywords, which provide initial weights for the words and sub-phrases of a candidate keyphrase. In the following sections, we will first describe how to build this database, then how to calculate a noun phrase's score, and finally how the keyphrases are extracted.

4.1.1. Building glossary database

The glossary database has two lists (tables): (a) a keyphrase list and (b) a keyword list. A keyphrase is an entry in the pre-defined keyphrase list, and it could contain one or more words; and a keyword means a single word parsed from list (a). Before using KIP, users will need a corresponding glossary database pertaining to the domain of input documents. When the system is applied to a different domain, the only thing required is to build or change to a new database specific to the new domain. In this study, we extracted keyphrases for medical documents. Therefore, we used MeSH to build our glossary database. MeSH is NLM's controlled vocabulary thesaurus.

The keyphrase list was generated by adding all the MeSH terms to it. The keyword list was automatically generated from the keyphrase list. To obtain the keywords, all keyphrases (MeSH terms) were split into individual words and added as keywords to the keyword list. The glossary database has two tables, one for keyphrases and another for keywords. The keyphrase table and keyword table all have two columns (keyphrases/keywords and weights).

The weights of these domain-specific keyphrases and keywords in the glossary database are assigned automatically through 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 this is that it reflects how domain-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; for each keyword in condition (B), the weight is Y divided by the times the keyword appears as part of a keyphrase; for each keyword in condition (C), the weight is $\frac{X+(Y/N)}{2}$, where N is the number of times that the keyword appears as part of a keyphrase.

(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 weight for this keyphrase.

The default values of the parameters mentioned above are obtained based on our testing using a number of test documents. KIP will use the weights of keyphrases and keywords in the database to calculate the scores of noun phrases in a document.

4.1.2. Calculating scores for keyphrase candidates

A noun phrase's score 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 within the noun phrase (we call a combination of adjacent words a "sub-phrase" of this noun phrase). So we have the following equation:

$$\text{The score of a noun phrase} = F \times S. \tag{1}$$

The sum of weights S is defined as

$$S = \sum_{i=1}^{N} w_i + \sum_{j=1}^{M} p_j, \tag{2}$$

where w_i is the weight of a word within this noun phrase and p_j is the weight of a sub-phrase within this noun phrase.

The following example is used to explain how a noun phrase's score is calculated. Assume there is a noun phrase "ABC," where A, B, and C are three words. The possible combinations of adjacent words are AB, BC, and ABC. The score for noun phrase "ABC" will be the frequency of "ABC" in this document multiplied by the summation of weights of A, B, C, AB, BC, and ABC. The motivation for including the weights of all possible sub-phrases into the phrase score, in addition to the weights of individual words, is to find out if a sub-phrase is a keyphrase in the glossary database. If it is, this phrase is expected to be more important. KIP will lookup the keyphrase table to obtain the weights for all the sub-phrases of the noun phrase. If a sub-phrase is found, the corresponding weight in the keyphrase table is assigned to this sub-phrase; otherwise, a predefined low weight will be assigned. 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 predefined weight will be assigned to it.

4.1.3. Extracting keyphrases

All the scores of keyphrase candidates are normalized to range from 0 to 1 after they are calculated. All candidate keyphrases for a document are then ranked in descending order by their scores. The keyphrases of a document can be extracted from the ranked list. To be as flexible as possible, the KIP system has a set of parameters for users to decide the number of keyphrases they want from a document. The number of extracted keyphrases for a document can be defined in three ways: (1) defining a specific number of keyphrases to be extracted; (2) specifying the percentage of noun phrases to be extracted (for example, top 10% 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 0.7 are to be extracted). KIP contains all the above basic options, as well as possible combinations of them.

An example of KIP is shown in Fig. 2. In this example, the system extracts five keyphrases for the document shown in the right frame. These five keyphrases are listed in the left frame.

4.2. Experiment

Usually, there are two ways to evaluate the effectiveness of a keyphrase extraction system. One is to use human judgment, asking domain experts to rate the keyphrases generated by the system. The second way, less costly, is to measure how well the system-generated keyphrases match the author-provided keyphrases. We chose the second approach and assessed KIP's effectiveness with medical documents by computing its precision and recall using author-provided keyphrases for documents. 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 are extracted by the keyphrase extraction system. Measuring precision and recall against author keyphrases is easy to carry out, since it does not involve human experts. Previous studies have used this measure and found it is an appropriate method to measure the effectiveness of a keyphrase extraction system [13,47,48]. We used 400 medical papers as the test documents in this evaluation. The sources of these papers are listed in Table 4. We also wanted to know how well KIP performs, so we compared KIP to another keyphrase extraction system, Kea [48]. The algorithm used by Kea has been described in Section 2.2. We could only get a demo version of Extractor [47], and it did not allow us to train it with new documents (medical documents). Other reported systems were not available to us for a comparison. So we only compared our system with Kea. In this experiment, we used Kea 3.0. We first trained it with 50 medical documents. The amount of trained documents was recommended by Kea's developers. After it was trained, a model for medical documents was created and

Fig. 2. A screenshot for KIP.

Table 4
Sources of documents used in KIP experiment

Source of documents	Number of documents
Journal of Biomedical Informatics (2003, 2004, 2005)	160
Journal of Computers in Biology and Medicine (2004, 2005)	100
Journal of Applied Clinical Medical Physics (2003, 2004, 2005)	70
Journal of Medical Informatics and the Internet in Medicine (2002, 2003)	70
All	400

used in this experiment. All these 400 papers have author-assigned keyphrases. Author-assigned keyphrases were removed from the papers before the documents were processed by KIP and Kea. The average length of these papers was 13 pages. The average number of author-assigned keyphrases for these papers was 4.1. We calculated the precision and recall for both systems when the number of extracted keyphrases was 5, 10, 15, and 20, respectively. The result is shown in Table 5. We also tested the statistical significance of the difference between precisions of the two systems, as well as their recalls, using a paired t test. From Table 5, we can see that, in respect to precision and recall, KIP performed better than Kea at all the comparison points. The results are significant at 99% confidence level.

We need to point out that some author-provided keyphrases may not occur in the document they are assigned to. According to Turney [47], about only 75% of author-provided keyphrases appear somewhere in the documents. That means the highest possible average recall for a system could only be 0.75, even when all the phrases are extracted from the documents. In our experiment, the average number of author-provided keyphrases for all the documents was only 4.1, so the precision would not be high when the number of extracted keyphrases was large. For example, when the number of extracted keyphrases for each document is 20, the highest possible average precision is only about 0.153 ($4.1 \times 0.75/20 = 0.153$).

To see how KIP performs in different domains, we compared KIP's performance in medical domain and Information Systems (IS) domain. Table 6 shows its precision and recall in both IS domain and medical domain when the number of extracted keyphrases is 5, 10, 15, and 20. The results show that the performance, in terms of recall, is basically the same in both domains. In terms of precision, KIP performs better in medical domain than in IS domain. The results in IS domain were obtained with 500 academic papers as test documents [17]. The average number of author-provided keyphrases per paper for these 500 papers was 4.7. For medical documents, the average number of author-provided keyphrases for each paper was 4.1. Among others, the difference between the average numbers of author-assigned keyphrases per paper was one factor affecting the results. If other conditions are the same, when this number is higher, the precision will be higher, since, statistically, the probability an extracted keyphrase is an author-assigned keyphrase will be greater when this number is higher. From Table 6 we can see that even though the average number of author-assigned keyphrases for medical document was lower than it was for IS documents, the precision is still basically the same for both. This means, in terms of precision, KIP also performed better in the medical domain than in IS domain. Besides the difference of the test documents, the underlying databases of KIP were also different for these two domains. When applying to IS domain, KIP used WordNet lexical database to do POS assignment and noun phrase identification; in contrast, it used an integrated database, combining terms from both WordNet and SPECIALIST Lexicon,

Table 5
Precision and recall for KIP and Kea in medical domain

Number of extracted keyphrases	Average precision ± SD		Significant test on precision difference (p value)	Average recall ± SD		Significant test on recall difference (p value)
	KIP	Kea		KIP	Kea	
5	0.26 ± 0.14	0.16 ± 0.16	<0.01	0.34 ± 0.19	0.20 ± 0.21	<0.01
10	0.19 ± 0.08	0.12 ± 0.08	<0.01	0.50 ± 0.20	0.31 ± 0.20	<0.01
15	0.15 ± 0.06	0.10 ± 0.06	<0.01	0.57 ± 0.23	0.35 ± 0.20	<0.01
20	0.12 ± 0.04	0.08 ± 0.05	<0.01	0.60 ± 0.23	0.39 ± 0.21	<0.01

Table 6
KIP's performance in the information System (IS) domain and medical domain

Number of extracted keyphrases	Average precision		Average recall	
	Medical domain	IS domain	Medical domain	IS domain
5	0.26	0.27	0.34	0.31
10	0.19	0.19	0.50	0.44
15	0.15	0.15	0.57	0.50
20	0.12	0.12	0.60	0.54

for medical domain. And the glossary databases for identifying keyphrases were also different for the two domains. MeSH was used to the medical documents, and an IS glossary database was used to the IS documents.

5. Conclusion

In Summary, a noun phrase extractor and a keyphrase identification program specialized for medical domain are described in this paper. We also report our experimental results based on medical documents. The experimental results show that the noun phrase extractor is effective in identifying noun phrase for medical documents, and the keyphrase identification program can effectively extract topical concepts for medical documents. They both performed better than the systems they were compared to.

Acknowledgments

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.

References

[1] Ono T, Hishigaki H, Tanigami A, Takagi T. Automated extraction of information on protein–protein interactions from the biological literature. Bioinformatics 2001;17:155–61.

[2] Jenssen T-K, Laegreid A, Komorowski J, Hovig E. A literature network of human genes for high-throughput analysis of gene expression. Nat Genet 2001;28:21–8.

[3] Marcotte E, Xenarios I, Eisenberg D. Mining literature for protein–protein interactions. Bioinformatics 2001;17:359–63.

[4] Stapley BJ, Benoit G. Biobibliometrics: information retrieval and visualization from co-occurrences of gene names in Medline abstracts. In: Proceedings of the Pacific symposium on biocomputing; 2000. p. 526–37.

[5] Ng S-K, Wong M. Toward routine automatic pathway discovery from on-line scientific text abstracts. Genome Inform 1999;10: 104–12.

[6] Blaschke C, Andrade MA, Ouzounis C, Valencia A. Automatic extraction of biological information from scientific text: protein–protein interactions. In: Proceedings of the AAAI conference on intelligent systems for molecular biology; 1999. p. 60–7.

[7] Thomas J, Milward D, Ouzounis C, Pulman S, Carroll M. Automatic extraction of protein interactions from scientific abstracts. In: Proceedings of the pacific symposium on biocomputing. Hawaii; 2000. p. 541–51.

[8] Friedman C, Kra P, Yu H, Krauthammer M, Rzhetsky A. GENIES: a natural-language processing system for the extraction of molecular pathways from journal articles. Bioinformatics 2001;17:74–82.

[9] Rindflesch TC, Tanabe L, Weinstein JN, Hunter L. Edgar: extraction of drugs, genes and relations from the biomedical literature. In: Proceedings of the Pacific symposium on biocomputing. Hawaii; 2000.

[10] Tanabe L, Scherf U, Smith LH, Lee JK, Hunter L, Weinstein JN. MedMiner: an internet text-mining tool for biomedical information, with application to gene expression profiling. BioTechniques 1999;27: 1210–7.

[11] Krauthammer M, Nenadic G. Term identification in the biomedical literature. J Biomed Inform 2004;37:512–26.

[12] Li Q, Wu YB, Bot RS, Chen X. Incorporating document keyphrases in search results. In: Proceedings of the tenth Americas conference on information systems, New York, New York; 2004.

[13] Jones S, Paynter GW. Automatic extraction of document keyphrases for use in digital libraries: evaluation and applications. J Am Soc Inf Sci Technol 2000;53(8):653–77.

[14] Gutwin C, Paynter G, Witten IH, Nevill-Manning C, Frank E. Improving browsing in digital libraries with keyphrase indexes. J Decis Support Syst 1999;27(1–2):81–104.

[15] Jonse S, Mahoui M. Hierarchical document clustering using automatically extracted keyphrase. In: Proceedings of the third international Asian conference on digital libraries. Seoul, Korea. 2000; p. 113–20.

[16] Fellbaum C. WordNet: An electronic lexical database. Cambridge: MIT Press; 1998.

[17] Wu YB, Li Q, Bot R, Chen X. Domain-specific keyphrase extraction. In: Conference on information and knowledge management, Bremen, Germany; 2005.

[18] Croft B, Turtle H, Lewis D. The use of phrases and structured queries in information retrieval. In: Proceedings of the SIGIR'91: the 14th annual international conference on research and development in information retrieval. Philadelphia, PA: ACM Press; 1991.

[19] Edgar KD, Nichols DM, Paynter G, Thomson K, Witten IH. A user evaluation of hierarchical phrase browsing. In: Proceedings of the 7th European conference on research and advanced technology for digital libraries, Trondheim, Norway; 2003.

[20] Fagan JL. The effectiveness of a nonsyntactic approach to automatic phrase indexing for document retrieval. J Am Soc Inf Sci 1989;40(2):115–32.

[21] Aronson AR, Rindflesch TC, Browne AC. Exploiting a large thesaurus for information retrieval. Proc. RIAO 1994:197–216.

[22] Nadkarni P, Chen D, Brandt D. UMLS concept indexing for production databases: a feasibility study. J Am Med Inform Association 2001;8:80–91.

[23] Aronson AR, Bodenreider O, Chang HF, Humphrey SM, Mork JG, Nelson SJ, The NLM indexing initiative. In: Proceedings of the AMIA Symposium; 2000. p. 17–21.

[24] Zamir O, Etzioni O. Grouper: a dynamic clustering interface to web search results. Comput Netw ISDN Syst 1999;31 (11–16):1361–74.

[25] Larkey LS. A patent search and classification system. In: Proceedings of digital libraries'99: the fourth ACM conference on digital libraries. Berkeley, CA: ACM Press; 1999.

[26] Blake C, Pratt W. A semantic approach to identify candidate treatments from existing medical literature. In: AAAI symposium on knowledge-based approaches. Stanford, CA; 2002.

[27] Kumar K, Palakal M, Mukhopadhyay S, Stephens M, Li H. BioMap: toward the development of a knowledge base of biomedical literature. In: 2004 ACM symposium on applied computing, Nicosia, Cyprus; 2004. p. 121–27.

[28] Aronson AR. Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. In: Proceedings of AMIA Symposium; 2001. p. 17–21.

[29] McCray AT, Srinivasan S, Browne AC. Lexical methods for managing variation in biomedical terminologies. In: Ozbolt JG, editor. Proceedings of the 18th SCAMC; 1994. p. 235–39.

[30] Friedman C, Shagina L, Lussier Y, Hripcsak G. Automated encoding of clinical documents based on natural language processing. J Am Med Inform Assoc 2004;11:392–402.

[31] Johnson SB. A semantic lexicon for medical language processing. J Am Med Inform Assoc 1999;6(3):205–18.

[32] Bodenreider O, Rindflesch C, Burgun A. Unsupervised, corpus-based method for extending a biomedical terminology. In: Proceedings of the workshop on natural language processing in the biomedical domain. Philadelphia; July 2002. p. 53–60.

[33] Rindflesch TC, Rajan JV, Hunter L. Extracting molecular binding relationships from biomedical text. Appl Nat Lang Process 2000:188–95.

[34] Cutting DR, Kupiec J, Pedersen JO, Sibun PA. practical part-of-speech tagger. In: Proceedings of the third conference on applied natural language processing; 1992.

[35] Demetriou G, Humphreys K, Gaizauskas R. Two applications of information extraction to biological science journal articles: enzyme interactions and protein structures. In: Proceedings of the pacific symposium on biocomputing. Hawaii, USA; 2000.

[36] Sekimizu T, Park HS, Tsujii J. Identifying the interaction between genes and gene products based on frequently seen verbs in Medline abstracts. Genome Inform Ser Workshop 1998.

[37] Chen H, Ng TD, Martinez J, Schatz BR. A concept space approach to addressing the vocabulary switching problem in scientific information retrieval: an experiment on the worm community system. J. Am. Soc. Inform. Sci. Technol. 1997;48(1):17–31.

[38] Bennett NA, He Q, Powell K, Schatz BR. Extracting noun phrases for all of MEDLINE. In: Proceedings of the AMIA Symposium; 1999. p. 671–5.

[39] Brill E. Transformation-based error-driven learning and natural language processing: a case study in part-of-speech tagging. Comput Linguist 1995;21(4):543–65.

[40] Voutilainen A. NPtool: a detector of english noun phrases. In: Proceedings of the workshop on very large corpora. Columbus, OH; 1993.

[41] Majoros1 WH, Subramanian GM, Yandell MD. Identification of key concepts in biomedical literature using a modified Markov heuristic. Bioinformatics 2003;19(3):402–7.

[42] Huang Y, Lowe H, Klein D, Cucina R. Improved identification of noun phrases in clinical radiology reports using a high-performance statistical natural language parser augmented with the UMLS specialist lexicon. J Am Med Informat Assoc 2005;12(3).

[43] Jones S, Staveley M. Phrasier: A system for interactive document retrieval using keyphrases. In: Proceedings of SIGIR'99, Berkeley, CA; 1999.

[44] Kosovac B, Vanier DJ, Froese TM. Use of keyphrase extraction software for creation of an AEC/FM thesaurus. Electron J Inf Technol Constr 2000;5:25–36.

[45] Krulwich B, Burkey C. Learning user information interests through the extraction of semantically significant phrases. In: AAAI 1996 spring symposium on machine learning in information access. California: AAAI Press; 1996.

[46] Zha H. Generic summarization and keyphrase extraction using mutual reinforcement principle and sentence clustering. In: Proceedings of the 25th annual international ACM SIGIR conference on research and development in information retrieval, Tampere, Finland; 2002.

[47] Turney PD. Learning algorithm for keyphrase extraction. Inf Retrieval 2000;2(4):303–36.

[48] Frank E, Paynter G, Witten IH, Gutwin C, Nevill-Manning, C. Domain-specific keyphrase extraction. In: Proceeding of the sixteenth international joint conference on artificial intelligence, San Mateo, CA; 1999.

[49] Smith L, Rindflesch T, Wilbur WJ. MedPost: a part-of-speech tagger for biomedical text. Bioinformatics 2004;20(14):2320–1.

[50] Tanabe L, Wilbur WJ. Tagging gene and protein names in biomedical text. Bioinformatics 2002;18(8):1124–32.

[51] Tolle KM, Chen H. Comparing noun phrasing techniques for use with medical digital library tools. J Am Soc for Inf Sci 2000;51(4): 352–70.