A Knowledge-based Approach for Retrieving Scenario-specific Medical Text Documents

Wesley W. Chu, Zhenyu Liu, Wenlei Mao and Qinghua Zou

Computer Science Department, University of California, Los Angeles, California, 90095

{wwc.vicliu,wenlei,zou}@cs.ucla.edu

Keywords

Knowledge-Based Information Retrieval, Indexing, Vector Space Model, Query Expansion

Abstract

Medical free-text queries often share the same scenario. A scenario represents a repeating task in healthcare. For example, a specific scenario is searching for treatment methods for a specific disease, where "treatment" is a term indicating the scenario. To support scenario-specific retrieval, in this paper we present a new knowledge-based approach to address these problems. In addition, we describe a testbed system developed using the approach. Our specific implementation uses the UMLS Metathesaurus and semantic structure to extract key concepts from a free-text. The approach uses phrase-based indexing to represent similar concepts, and query expansion to improve matching query terms with the terms in the document. The system formulates the query based on the user's input and the selected scenario template such as "disease, treatment" or "disease, diagnosis." Thus, it is able to retrieve documents relevant to the specific scenario. Evaluating the system using the standard OSHMED corpus, our empirical results validate the effectiveness of this new approach over the traditional text retrieval techniques.

A. Introduction

The volume of medical information and clinical data is growing at explosive rates. Ten years ago, medical publications were added to the world's biomedical journal collections at the rate of approximately 3,000 per month. Today, the volume is growing at 1,000 per day in Medline alone [NLM02]. As an artifact of patient care, hospitals generate huge amounts of healthcare data that is digitally available. As has been stated in the Institute of Medicine's report defining a new health system for the 21st century [IOM01], the delivery of quality healthcare to consumers requires the availability and use of accurate information/knowledge compiled from this large volume of information. The demand by society and professional organizations for the use of evidence-based practices to help improve the quality of care also adds great pressure on healthcare professionals to regularly access the highest quality information during the processes of healthcare planning, decision and delivery. Therefore, computer-assisted information retrieval and processing are necessary today for supporting quality decision-making and helping to overcome human cognitive constraints [Chu02].

A Medical Digital Library (MDL) consists of three types of data: structure data such as patient lab data and demographic data; multi-media images such as MRI's; and free-text documents such as patient reports, medical literature, teaching files and news articles. Previous research focused on the effective retrieval of structure data and image data [Chu98a, Chu98b]. However, many medical records are in free-text form and access to these records usually follows well-defined information gathering scenarios. A scenario can be defined as a reoccurring information need where the specific contextual information changes. For example, a physician may pose the following two queries, one for diagnosis and the other for treatment of a disease:

- diagnosis scenario: "***diagnosis*** of large cell lung cancer," from all patient reports

- treatment scenario: "***treatment*** of large cell lung cancer," from the collection of medical literature articles (e.g. MEDLINE references).

Scenario-specific information retrieval is not adequately addressed either by current search engines (e.g., Google[1] or Yahoo! search[2]) or traditional information retrieval systems (e.g. SMART [SM83] or INQUIRY [JCH92]).

- Search engines are optimized for general Web queries. In a recent study of multiple Web query logs, Rose et al. have reported less than 36% of such Web queries belong to *scenario-specific queries* [RL04]. For best retrieval performance for the majority of Web queries, search engines usually apply techniques (e.g. Pagerank [BP98]) that are not specialized in scenario-specific information retrieval.
- Traditional information retrieval systems are useful for retrieval of general documents; however these systems cannot support scenario-specific information retrieval because of:
  1. The lack of effective techniques to extract key features from free-text for indexing.
  2. The lack of effective techniques to identify phrases with similar concepts in free-text.
  3. The terms used in a query are often mismatched with those from the document containing information on the same scenario.

To address this problem, we have developed the following knowledge-based techniques to provide scenario specific information retrieval:

Extracting key concepts from free-text. A knowledge-based (e.g. UMLS) approach is used to automatically extract key concepts from a free-text by permuting the set of words in the input free-text and generating all valid concepts defined in the knowledge base.

---

[1] http://www.google.com/

[2] http://search.yahoo.com/

Phrase-based vector space model (VSM). Phrase-based VSM [Mao02] identifies terms with similar meanings and represents them based on both concepts and stems. As a result, phrase-based VSM yields a significantly better retrieval performance than the stem-based VSM.

Knowledge-based query expansion. Traditional expansion techniques append all statistically co-occurring terms into the original query, many of which may not be scenario-specific. We use a knowledge-based approach that only appends the query with terms related to the scenario of the query.

B. Extracting Key Concepts from Documents

B.1 The UMLS knowledge source

Since our approach is leveraged on knowledge bases, we shall first briefly describe the Unified Medical Language System's (UMLS) [NLM03] knowledge sources and then present an index tool called *IndexFinder*, which is used for extracting key concepts from free-texts. UMLS is a standard medical knowledge source developed by the National Library of Medicine. The knowledge source consists of the *UMLS Metathesaurus*, the *SPECIALIST lexicon*, and the *UMLS Semantic Network*.

The Metathesaurus is a central vocabulary component that contains 1.6M phrases representing over 800K concepts from more than 60 vocabularies and classifications. We use it as the controlled vocabulary to detect concepts, and derive the conceptual relations using the hypernym relations encoded in it.

A concept unique identifier (CUI) identifies each concept. The Metathesaurus encodes "broader-narrower-than" types of relations among the concepts. For example, "lung cancer" is a broader concept than "lung neoplasm." A class of concepts in the Metathesaurus is abstracted into

one *semantic type* in the Semantic Network. For example, the concept "lung cancer" belongs to the semantic type "disease and syndrome." Each semantic type has several semantic relationships with other types, e.g., "disease and syndrome" is "treated by" "therapeutic or preventive procedures," "pharmacological substance" and "medical devices." These semantics are used for knowledge-based query expansion (see section D).

B.2 Indexing for Free-text Documents

Indexing free-text is a very difficult task. First, documents are not written using a controlled vocabulary. Similar concept terms, synonyms, and other attributes in the free-text significantly complicate the indexing task. This also applies to ad hoc queries since they share the same problems. Unlike medical literature, where the author(s) provides key words which may be used for indexing purposes, many free-text documents do not provide such information. To effectively retrieve these free-texts, we are motivated to extract the key concepts from these documents. To rapidly retrieve the relevant information/knowledge for a query from a large number of documents, we propose to develop an intelligent directory system for free-text where the document can be retrieved based on a set of index terms. Having located a group of documents that satisfy the key concept terms, traditional IR techniques can then be used to rank these documents.

Thus, extracting key concepts from free-texts automatically is a critical task. Words or word stems are commonly used for indexing, and these indexing techniques do not require any knowledge source. However, synonyms and some morphological differences between the texts in the target documents and the search words used often hamper the search results and are beyond the technological spectrum of word/stem indexing and matching techniques. This issue is particularly problematic in healthcare, wherein the biomedical language is packed with many interchangeable terms, such as common cold and coryza, mass and lump, fever and pyrexia, weakness and paresis, and many others.

Therefore, we developed indexing systems based on some standard descriptors or dictionaries, such as UMLS. Using search terms generated from standard dictionaries also helps resolve the synonym and morphological differences, and thus reduces user frustrations by minimizing the rates of missed-hits/failed searches. A significant amount of research has aimed at developing effective methods for mapping free-text into UMLS concepts. Examples of such efforts include SENSE [Zieman97], MicroMeSH [Elkin88], Metaphrase [Tuttle98], KnowledgeMap [Denny03], PhraseX [Srinivasan02], *MetaMap* [Aronson01]. Many of these efforts use natural language processing (NLP) techniques to parse passages of free-text to generate noun phrases, which are in turn mapped into UMLS phrases. Although this approach achieves some success, some important concepts can never be discovered through the identification of noun phrases. Table 1 provides examples of texts that reveal the shortcomings of the use of noun phrases.

In example 1, the key concept is actually formed using a word from the first line (prostate) with a word from the second line (hyperplasia) corresponding to concept ID 33577 in the UMLS Metathesaurus. The second example, a word from the subject and two words from the location phrase combine to form the key concept, "left lung mass," which corresponds to concept ID 746117 in the UMLS Metathesaurus. In both cases, noun phrase identification would fail to find the key concepts of the texts.

A second weakness is that noun phrase identification and natural language processing (NLP) requires significant computing resources. As a result, most of the NLP systems work in an offline mode and thus are not suitable for mapping large volumes of free-text into UMLS concepts in real time. To remedy these shortcomings, we developed a new tool called *IndexFinder* to extract key concepts from free-text.

B.3 IndexFinder

We developed a novel approach to detect medical concepts from free-text by permuting words in a sentence to generate concept candidates that match the UMLS-controlled vocabulary. Since the generated valid controlled vocabulary may not be relevant to the query, syntactic and semantic filters based on a specific scenario are used to filter out irrelevant concepts. The specific processing stages of the IndexFinder are discussed below.

Text Preprocessing

Since *IndexFinder* uses the UMLS normalized string table for indexing and also supports certain types of abbreviations, we need to preprocess the input text to normalize words [Aro 01], detect undefined and ambiguous abbreviations as well as remove stop words to increase the accuracy of the extraction.

IndexFinder first converts the UMLS controlled vocabulary into an efficient concept indexing structure that resides in the main memory and thus avoids disk access. To detect the concepts embedded in a free-text sentence, *IndexFinder* scans through the sentence word by word, looks up the indexing structure and marks every concept where all the words representing that concept have appeared in the sentence. We use the UMLS SPECIALIST lexicon for word normalization, and handle synonyms by mapping different wording of the same concept into one entry in the indexing structure. This indexing and matching technique is efficient and able to generate responses in real-time for free-text indexing.

Figure 1 shows the web interface for *IndexFinder*. The interface has two text panes: the upper text pane takes free-text as input and the lower one outputs the identified UMLS concepts. Each line in the output pane shows one identified concept which contains the concept ID, the concept's phrase string, and the concept's semantic type. Part of the UMLS concepts detected from the input

pane is shown in the output pane. Three buttons for adding synonyms, removing inflection, and configuring options are at the top of the input window. Results appear when a user clicks the "IFinder Search" button below the input window. Eighteen phrases were found when no filters were applied. Each line has a UMLS concept identifier, phrase text, and corresponding semantic type.

Syntactic and Semantic Filtering

Although word permutation detects more concept candidates, some concepts may be irrelevant to the original sentence. *IndexFinder* applies filters that use syntactic or semantic information from the original sentence and the knowledge source to filter out irrelevant concepts. For example, a physician wants to know what kind of diseases a patient suffers. Rather than returning all concepts to the physician, returning disease-related UMLS phrases are much more desirable. In our current implementation, we consider six types of filters as shown in Figure 2.

The first three filters are applied during the mapping process:

- *Symbol Type filter*: specifies the symbol types of interests. For example, if a user wants to ignore digits like *MetaMap* did, she can simply not check the Digits box as in Figure 2.
- *Term Length filter*: specifies the length limitation of candidate phrases.
- *Coverage filter*: to specifies the coverage condition for a candidate phrase. It has three options, *at least one*, *majority*, and *all*. By default, it is *all* where every word in a candidate phrase should be present in the input text.

The latter three filters are used for further pruning the candidate phrases:

- *Subset filter*: removes phrases if they are subsets of some other phrases. For example, if results are {*lung cancer*} and {*cancer*}, then {*cancer*} will be removed since it is a subset of the former.

- *Range filter*: removes a phrase if the phrase is found from words in the input text to exceed a specific distance.
- *Semantic filter*: removes the phrases of semantic types that the user is not interested in. In UMLS, 134 semantic types are defined and each concept maps to one or several semantic types. For example, the user can select Disease or Syndrome and its two sub types, as shown in Figure 2, so that the resulting phrases will be of these three types. As a result, the filter also eliminates those irrelevant phrases from the set of phrase candidates. we have identified five semantic filters: Diseases, Findings, Drugs, Medical Procedures, and Body Parts. Each of them consists of a set of UMLS semantic types. We also noted that the semantics in a section heading of a document are useful in selecting the type of semantic filter(s) to effectively filter out irrelevant terms.

Figure 3 shows the filtering result for the sample input in Figure 1, (also depicted at the top of Figure 3). When a subset filter is used, 8 phrases are returned. If the Pathologic Function is selected, four answers will be returned. The two phrases, prostate and focal, will be given if the user wants to know body parts or spatial characteristics. There is only one diagnostic procedure used, which is prostate biopsy.

Evaluation

The *IndexFinder* is written in *C#,* and is running on a 1.2GHz PC machine with 512MB main memory. We have implemented the algorithm as a web-based service named *IndexFinder* that provides web interfaces for users and programs. We tested the web service using 5,783 reports of 128 patients from the UCLA Hospital. The total size of the documents is 10,8M bytes. There are 910K concepts found in 254 seconds. Therefore, the throughput is about 42.7 K bytes per second, which validates that the system can extract key concepts from clinical free-texts in real-time. Next, we manually examined the mapping results for 100 topic sentences from the above set of patient re-

ports. We consider a concept both with and without negation as relevant to the original sentence. For example, both "*evidence*" and "*no evidence*" are relevant to the input "*no evidence of malignancy*." There is a total of 456 UMLS phrases found of the 100 topic sentences. We noticed 18 concepts that are not from a single noun phrase and thus cannot be detected by NLP-based methods. Further, we note that all the concepts detected by *IndexFinder* are relevant. Filtering is effective in eliminating the irrelevant terms from the validated candidates.

Comparison with NLP approach

We performed a comparison study between *IndexFinder* and *MetaMap*, which uses the NLP method. We noticed that the NLP tends to break each sentence into small fragments. Conversely, *IndexFinder* considers all the possible word combinations in the input unit that are valid in UMLS. As a result, NLP does not yield concepts as specific as *IndexFinder*, as shown in Figure 4.

While these results are promising, further evaluation of our method is needed. Future evaluation will include generating a test dataset by randomly selecting a set of topic sentences from patient reports and then comparing the accuracy of the indexing terms generated by the *IndexFinder* in terms of the numbers of false negatives and false positives [FH 98].

The key terms extracted by *IndexFinder* can be used for: 1) indexing the free-text documents; 2) formulating scenario-specific queries for content correlation; and 3) transforming the ad hoc query terms to controlled vocabulary, thus increasing retrieval effectiveness.

An Example

As a specific clinical application for this research, we have focused on using the *IndexFinder* to intelligently filter all clinical free-text in an electronic medical record for documents that specifically mention brain tumor-related content. It is not uncommon for a brain tumor patient to have as many as 50 clinical documents in their medical record. Many of these documents will have nothing to do with the treatment of the brain tumor, but are concerned with other health problems. These docu-

ments consist of primary care clinical notes, specialist clinical notes, pathology reports, laboratory results, radiology reports, and surgical notes. Figure 5 shows an excerpt from a radiology report.

Since our interests focus on brain tumor-related concepts, we can specify a semantic filter work list of pertinent documents based on brain tumor characteristics including: cancer type, anatomical location, and medical interventions. These characteristics are then mapped to relevant UMLS semantic types to define semantic filters, as shown in Table 3.

A clinician looking for specific documents that address a certain type of brain tumor (i.e. *meningioma*) would have to carefully search the individual documents. With *IndexFinder*, only two key terms, *meningioma* and *encephalomalacia*, are returned for the above text excerpt as shown in Table 3. The two concepts, in fact, are important in the excerpt and thus are good terms for indexing.

C. Phrase-based Vector Space Model for Automatic Document Retrieval

IndexFinder is able to extract key concepts from free-text for the directory system. Based on a given query, the directory system is able to identify a group of documents that match with the key concepts in the query from a corpus. We need to rank and order this set of documents by their similarity with the target document (query). The Vector Space Model (VSM) can be used in information retrieval to perform such a ranking. In this section, we shall first present an overview of the Vector Space Model. Next we introduce the phrase Vector Space Model, which is a new paradigm for representing documents. Finally, we present the performance improvement of this new model and its computation complexity.

Retrieval systems consist of two main processes, *indexing* and *matching*. Indexing is the process of selecting *content identifiers*, also known as *terms* in this setting, to represent a text. Matching is

the process of computing a measure of similarity between two text representations. It is possible for human experts to manually index documents. However, it is more efficient and thus more common to use computer programs to automatically index a large collection of documents.

A basic automatic indexing procedure for English usually consists of: (1) splitting the text into words (tokenization), (2) removing frequently occurring words such as prepositions and pronouns (removal of stop words), and (3) conflating morphologically related words to a common word stem (stemming). The resulting word stems would be the terms for the given text.

In early retrieval systems, queries were represented as Boolean combinations of terms, and the set of documents that satisfied the Boolean expression was returned in response to the query. Since its inception, the vector space model (VSM) [SWY75] is the most popular model in information retrieval. In this model, documents and queries are represented by vectors in an n-dimensional space, where n is the number of distinct terms. Each axis in this n-dimensional space corresponds to one term. Given a query, a VSM system produces a ranked list of documents ordered by their similarities to the query. The similarity between a query and a document is computed using a metric on their respective vectors.

C.1 The Problem

Although word stems have been shown to be quite effective indexing terms, a recurring question in document retrieval concerns what should be used as the basic unit to identify the content in the documents or what should be identified as a term?

The problem of using word stems as terms is manifested in several ways:

1. The component words of a phrase sometimes has only a remote, if any, relation with the phrase. For example, separating "photo synthesis" into "photo" and "synthesis" could be misleading.
2. Words could be too general. For example, the individual words "family" and "doctor" are not specific enough to distinguish between "family doctor" and "doctor family."

3. Different words could be used to represent the same thing. For example, both "hyperthermia" and "fever" indicate an abnormal body temperature elevation.
4. The same word could mean different things. For example, "hyperthermia" can indicate an abnormal body temperature elevation, as well as a treatment in which body tissue is exposed to high temperature to damage and kill cancer cells.

As a result, many researchers proposed both phrases and concepts in place of words or word stems as content identifiers. However, neither the phrases nor the concepts had been shown to produce significantly better results than word stems in automatic indexing for general document collection. On the other hand, through manual indexing, [GVC98] showed the potential of concept-based indexing to produce significant improvements over the stem-based scheme. [JC99] showed that using n-word combination indexing yields improved retrieval performance for query containing n-word terms. The high potential shown there and the low performances of current automatic indexing schemes using phrases and concepts led us to the search of such a scheme.

Also, to facilitate discussion, we use the following example query from the medical domain throughout the discussion, "Hyperthermia, leukocytosis, increased intracranial pressure, and central herniation. Cerebral edema secondary to infection, diagnosis and treatment." The first part of the query is a brief description of the patient; the second part is the information desired.

C.2 Vector Space Models

C.2.1 Stem-based Vector Space Mode

In a stem-based VSM, morphological variants of a word like "edema" and "edemas" are conflated into a single word stem, e.g., "edem" using the Lovins stemmer [Lov68], and the resulting word stems are used as terms to represent the documents. Using the Lovins stemmer, the example query becomes "hypertherm," "leukocytos," "increas," "intracran," "pressur," etc.

Not all word stems are equally important. Authors usually repeat words as they elaborate the major aspects of a subject. Therefore, a frequent word stem in a document is often more important than an infrequent one. On the other hand, a word stem that appears in many documents is less specific than one that appears in only a few. Combining these two aspects, we often evaluate the importance of a word stem following a *term-frequency-inverse-document-frequency* (tf-idf) scheme. We define the weight of stems *s* in document *x* as, $w_{s,x} = \tau_{s,x} \iota_s$, where $\tau_{s,x}$ is the number of times *s* occurs in *x*, often called the term frequency of *s*, and ι_s is the inverse document frequency of stem *s*. One way to compute the inverse document frequency is $\iota_s = \log_2(N / n_s) + 1$, where *N* is the number of documents in the collection and n_s is the number of documents containing stem *s*, often called the document frequency of *s* [SM 87].

To compute the document similarity in the stem-based VSM, we define the *stem-based inner product* between documents *x* and *y* as $\langle x, y \rangle^s = \sum_{s \in S} w_{s,x} w_{s,y} = \sum \iota_s^2 \tau_{s,x} \tau_{s,y}$, and define their similarity as the cosine of the angle between their respective document vector [s,

$$sim^s(x, y) = \frac{\langle x, y \rangle^s}{\sqrt{\langle x, x \rangle^s \langle y, y \rangle^s}}.$$

C.2.2 Concept-based Vector Space Model

Using word stems to represent document results in the inappropriate fragmentation of multi-word concepts such as “increased intracranial pressure” into their component stems like “increas,” “intracran,” and “pressur.” Clearly, using concepts instead of word stems as content identifiers should produce a vector space model that better captures the document’s content, and therefore results in more effective document retrieval.

However, using concepts is more complex than using word stems, because, 1) concepts are usually represented by multi-word phrases and, 2) there exist polysemous and synonymous phrases. A phrase is *polysemous* if it can be used to express different meanings, and two phrases are *synonymous* if they can be used to express the same meaning. For example, "fever" and "hyperthermia" are synonyms since both can be used to denote "an abnormal elevation of the body temperature." On the other hand, "hyperthermia" is polysemous, because it can be used to mean either "fever" or a type of "treatment." Using concepts is more complex also because 3) some concepts are related to one another.

Assuming that we can partition the documents into phrases, and ignoring the polysemy, our example query using the UMLS concept unique identifiers (CUI) becomes (C0015967), (C0023518), and (C0151740) etc., representing "hyperthermia," "leukocytosis," and "increased intracranial pressure," etc., respectively [Med01].

Not all concepts are equally important, just as not all stems are equally so. We define the weight of a concept c in document x following the tf-idf scheme just like before,

$w_{c,x} = \tau_{c,x} \iota_c = \tau_{c,x} \left(\log_2 \left(N / n_c \right) + 1 \right)$, where $\tau_{c,x}$ is the number of times c appears in x, N is the number of documents in the collection, and n_c is the number of documents containing c.

Unlike in the stem-based VSM, where different word stems are considered unrelated, we define the *concept-based inner product* between documents x and y as

$$\langle x, y \rangle^c = \sum_{c \in C} \sum_{d \in C} \iota_c \tau_{c,x} \iota_d \tau_{d,y} s^c(c,d) \qquad (1)$$

where we take $s^c(c,d)$, the conceptual similarity between concepts c and d, into consideration. Conceptual similarity will be discussed in later sections. The similarity between documents x and

y is defined to be the cosine of the angle between their respective document vectors,

$$sim^c(x,y) = \frac{\langle x,y\rangle^c}{\sqrt{\langle x,x\rangle^c \langle y,y\rangle^c}}.$$

C.2.3 Phrase-based Vector Space Model

Concepts in controlled vocabularies such as UMLS are used in the concept-based VSM. Conceptual similarities needed there are often derived from knowledge sources. The qualities of such vector space models therefore depend heavily on the qualities of the controlled vocabularies and the knowledge sources. Some concepts could be missing from the controlled vocabularies. For example, if we detect only concept C0021852 for "small bowel" in the phrase "infiltrative small bowel process" and find no concepts matching either the entire phrase, or the fragments "infiltrative" and "process," then we are losing important information when we represent documents using concepts only. Furthermore, missing certain conceptual relations in the knowledge sources potentially degrades retrieval effectiveness. For example, treating "cerebral edema" and "cerebral lesion" as unrelated is potentially harmful. To remedy the incompleteness of the controlled vocabularies and the knowledge sources, we propose a phrase-based VSM.

In the phrase-based VSM, a document is represented as a set of phrases. Each phrase may correspond to multiple concepts (due to polysemy) and consist of several word stems. For example, "infiltrative small bowel process" is represented by phrases (; "infiltr"), (C0021852; "smal", "bowel"), (; "proces"). Our example query now becomes (C0015967, C0203597; "hypertherm"), (C0023518; "leukocytos"), and (C0151740; "increas", "intracran", "pressur") etc.

A phrase is represented by two sets. The first set consists of ordered pairs of the phrase's word stems (s) and their occurrence counts in the phrase ($\pi_{s,p}$). The second set consists of ordered pairs of the phrase's concepts (c) and their occurrence counts ($\pi_{c,p}$). Formally, a phrase (p) is defined as the pair of sets where $p = (\{(s, \pi_{s,p})\}s\in S, \{(c, \pi_{c,p})\}c\in C)$. We denote the set of all phrases by *P*.

Furthermore, we require that there is at least one stem in each phrase, i.e., for each phrase $p \in P$, there exists some stem s such that $\pi_{s,p} \geq 1$. We use a *phrase vector* x^p to represent a document x, $x^p = \{(p, \tau_{p,x})\}_{p \in P}$, where $\tau_{p,x}$ is the number of times phrase p occurs in document x. And we define the *phrase-based inner product* as

$$\langle x, y \rangle^p = \sum_{p \in P} \sum_{q \in P} \tau_{p,x} \tau_{q,y} s^p(p, q)$$

where we use $s^p(p; q)$ to measure the similarity between phrases p and q. We call $s^p(p; q)$ the *phrase similarity* between phrases p and q, and define it as

$$s^p(p, q) = \max\left(\left(f^s \sum_{s \in S} \iota_s^2 \pi_{s,p} \pi_{s,q} \right), \left(f^c \sum_{c \in C} \sum_{d \in C} \iota_c \pi_{c,p} \iota_d \pi_{d,q} s^c(c, d) \right) \right)$$

where $\iota_s, \iota_c, \iota_d > 0$ are the inverse document frequencies of stem s, concept c, and concept d respectively, and $s^c(c; d)$ is the conceptual similarity between concepts c and d. As in the concept-based VSM, we ignore polysemy and assume each phrase expresses only one concept,

$$\pi_{c,p} = \delta_{c,c_p} = \begin{cases} 1 & \text{if } c = c_p \\ 0 & \text{if } c \neq c_p \end{cases}$$

where c_p is the concept that phrase p expresses.

The similarity between two concepts must also be defined. Among the many possible conceptual relations, we concentrate on the *is-a* relation, also called *hypernym* relation. A simple example is that "fever" is a hypernym of "body temperature elevation." Hypernym relations are transitive [Lyo77]. We derive the similarity between a pair of concepts using their relative position in a hypernym hierarchy. For a pair of ancestor-descendant concepts, c and d, in the hypernym hierarchy, we define their conceptual similarity as

$$s^c(c,d) = \frac{1}{l(c,d)\log_2(D(c)+D(d)+1)} \quad (2)$$

where $l(c,d)$ is the number of hops between *c* and *d* in the hierarchy, and $D(c)$ and $D(d)$ are the descendant counts of *c* and *d* respectively.

Then the phrase similarity is reduced to

$$s^p(p,q) = \max\left(\left(f^s \sum_{s\in S} \iota_s^2 \pi_{s,p}\pi_{s,q}\right), \left(f^c \iota_{c_p}\iota_{d_q} s^c(c_p,d_q)\right)\right) \quad (3)$$

where c_p is the concept phrase *p* expresses, and d_q is the concept *q* expresses. Here we use two contribution factors, *f*s and *f*+, to specify the relative importance of the stem contribution and the concept contribution in the overall phrase similarity. The stem contribution

$$f^s \sum_{s\in S} \iota_s^2 \pi_{s,p}\pi_{s,q}$$

measures the stem overlaps between phrases *p* and *q*, and the concept contribution

$$f^c \iota_{c_p}\iota_{d_q} s^c(c_p,d_q)$$

takes the concept interrelation into consideration. Conceptually, when combining the stem contribution and the concept contribution this way, we use stem overlaps to compensate for the incompleteness of the controlled vocabularies in encoding all necessary concepts, and the incompleteness of the knowledge sources in describing all necessary concept interrelations. Once again, we define the *phrase-based document similarity* between documents *x* and *y* to be the cosine of the angle between their respective phrase vectors,

$$\mathrm{sim}^p(x,y) = \frac{\langle x,y\rangle^p}{\sqrt{\langle x,x\rangle^p \langle y,y\rangle^p}}$$

Phrase Detection

The building blocks of the concept-based VSM and the phrase-based VSM are phrases. A phrase usually consists of multiple words. Given a controlled vocabulary containing a set of phrases, P, and a set of documents, X, we need to efficiently detect the occurrences of the phrases in P in each of the documents in X.

A naive algorithm (see [Gus77]) requires $O(N_x N_p)$ word comparisons in the worst case, where N_x is the total number of words in the document set X and N_p is the total number of words in all the phrases in P. There are N_p = 6.7M words in the 1.3M English phrases in UMLS. Using the statistics of the larger OHSUMED collection shown in Table 4, we see that on average there are 112 x 1.25 x 14K = 2.0M words in the test documents. The naïve algorithm described above is too time consuming, and thus unacceptable for phrase detection. On the other hand, the Aho-Corasick algorithm [AC75] detects all the occurrences of the phrases in P from the documents in X using $O(N_x + N_p)$ word comparisons. Therefore, we adapt the Aho-Corasick algorithm for phrase detection:

1. The Aho-Corasick algorithm detects all occurrences of any phrase in a document. However, we only keep the longest, most specific phrase. For example, although both "edema" and "cerebral edema" are detected in the sample query, we keep only the latter, the more specific concept, and ignore the former, the more general concept.
2. To detect multi-word phrases, we match stems instead of words in a document with the UMLS phrases. To avoid conflating different abbreviations into a single stem, we define the stem for a word shorter than four characters to be the original word.
3. In English, about 250 common words such as "a" and "the" appear frequently.

It is a standard practice to include them in a stop list and remove them from document representations [SM83]. In our phrase detection, we remove the stop words in the stop list *after* multi-word phrase detection. In this way, we correctly detect "secondary to" and "infection" from "cerebral

edema secondary to infection." We would incorrectly detect "secondary infection" if the stop words ("to" in this case) were removed before the phrase detection.

Primitive Word Sense Disambiguation

Polysemy is one of the difficulties people encounter when using concepts. A polysemous phrase can express multiple meanings. As a result, it is necessary to disambiguate polysemous phrases in document retrieval. For example, seeing "hyperthermia," it is necessary to figure out whether it means "fever" or a type of "treatment" using word sense disambiguation [IV98]. The current accuracy and efficiency of word sense disambiguation algorithms are low. We perform a very primitive word sense disambiguation based on the following observation. UMLS tends to assign a smaller CUI to the more popular sense of a phrase. For example, the CUI for the "fever" sense of "hyperthermia" is C0015967, while the CUI for its "treatment" sense is C0203597. Therefore, we use the concept corresponding to the smallest CUI in the concept-based VSM and the phrase-based VSM.

C.3 Retrieval Effectiveness Evaluation

C.3.1 The Knowledge Source, UMLS

UMLS [NLM01] is a medical lexical knowledge source and a set of associated lexical programs. The knowledge source consists of the UMLS Metathesaurus, the SPECIALIST lexicon, and the UMLS semantic network. Particularly of interest to us is its central vocabulary component -- the Metathesaurus. It contains 1.6M biomedical phrases representing over 800K concepts from more than 60 vocabularies and classifications.

A concept unique identifier (CUI) identifies each concept. Because of synonymy, multiple phrases can be associated with one CUI. For example, 71 phrases in 15 languages are associated with CUI C0015967. Some examples of English phrases for that CUI include "fever," "high body

temperature," "temperature, high," and "hyperthermia." On the other hand, a phrase can express multiple meanings. For example, "hyperthermia" can be associated with both C0015967 (the "fever" sense) and C0203597 (the "treatment" sense).

The Metathesaurus encodes many conceptual relations. We are particularly interested in the hypernym/hyponym relations. Two pairs of relations in UMLS roughly correspond to the hypernym/hyponym relations: the RB/RN (broader than/narrower than) and the PAR/CHD (parent/child) relations. For example, C0015967 (fever) has a parent concept C0005904 (body temperature change). RB and RN are redundant -- for two concepts *c* and *d*, if (*c, d*) is in the RB relations, then (*d, c*) is in the RN relations, and vice versa. Similarly, PAR and CHD are redundant. As a result, we combine RB and PAR into a single hypernym hierarchy. Hypernymy is transitive [Lyo77]. For example, "sign and symptom" is a hypernym of "body temperature change," and "body temperature change" is a hypernym of "hyperthermia," so "sign and symptom" is also a hypernym of "hyperthermia." However, the UMLS Metathesaurus encodes only the direct hypernym relations but not the transitive closure. We derive the transitive closure of the hypernym relation and use Formula (3) to compute the conceptual similarities.

UMLS plays two important roles in the concept-based VSM and the phrase-based VSM. First, we use its Metathesaurus as a controlled vocabulary in phrase detection. Second, we use the hypernym relations encoded in RB and PAR in conceptual similarity derivation.

C.3.2 The Test Collections

To compare the effectiveness of different vector space models in document retrieval, we need a test collection that provides 1) a set of queries, 2) a set of documents, and 3) the judgments indicating if a document is relevant to a query.

OHSUMED [HBL94] is a test collection widely used in recent information retrieval tests. OHSUMED contains 106 queries. Each query contains a patient description and an information

need. Our example query is query 57 in the collection. The document collection is a subset of 348K MEDLINE references from 1987 to 1991. Seventy-five percent of the references contain titles and abstracts, while the remainder has only titles. Each reference also contains human-assigned subject headings from the Medical Subject Headings. 14,430 references in the document collection are judged by "physicians who were clinically active and were current fellows in general medicine or medical informatics or senior medical residents" to be definitely relevant, possibly relevant, or non-relevant to each of the 105 [1] queries. The standard recall and precision evaluation that we shall discuss later requires a binary relevance judgment -- relevant or non-relevant. This can be easily achieved by merging the definitely relevant and the possibly relevant documents into a single relevant category.

Another test collection known as Medlars [Sal75] is based on MEDLINE reference collections from 1964 to 1966. It has been used extensively in document retrieval system comparisons. There are 30 queries and 1,033 references in the collection. The judgments provided with the Medlars collections were made by a medical school student.

We use both test collections to compare the retrieval effectiveness of different methods. However, based on the qualification of the human experts, the extent, and the up-to-dateness of these collections, we believe that OHSUMED reflects expert judgment better. As such, we direct the attention of the reader to the results obtained from the OHSUMED collection in later sections. Table 4 compares some statistics of the two collections. Besides the collection size difference discussed above, other noticeable differences include: OHSUMED queries are slightly shorter than those in Medlars; OHSUMED documents on average contain more long phrases (those with more than one stem); and Medlars contains slightly more polysemous phrases (those with multiple senses).

C.3.3. Retrieval Effectiveness Measures

The goal of document retrieval is to return documents relevant to a user query before non-relevant ones. The effectiveness of a document retrieval system is measured by the recall and precision [Rij79,SM83] based on the user's judgment of whether each document is relevant to a query *q*. When a certain number of documents are returned, *precision* is defined to be the proportion of the retrieved documents that are relevant and *recall* is defined to be the proportion of the relevant documents retrieved so far. More specifically, if we use *Rq* to represent the set of documents relevant to *q*, and *A* to represent the set of retrieved documents, then we define

$$\text{precision} = \frac{|R_q \cap A|}{|A|} \text{ and recall} = \frac{|R_q \cap A|}{|R_q|}$$

There are several ways to evaluate the retrieval effectiveness using recall and precision.

To visually display the change in the precision values as documents are retrieved, we interpolate the precision values to a set of eleven recall points 0*,* 0.1*,* 0.2*, . . . ,* 1. Averaging the precision values over a set of queries at these recall points illustrates the behavior of a system. Further averaging the eleven average precision values, we arrive at the *average 11-point average precision*, denoted by *GP*11. Instead of interpolating the precision values to a set of standard recall points, we could also compute the average precision values after each relevant document is retrieved. The average of such a value over a set of queries is called the *average precision*, denoted by *GP*.

The two retrieval effectiveness measures described above, *GP*11 and *GP*, measure the average retrieval effectiveness of a system when different amounts of documents are retrieved. Sometimes, it is important to know the performance of a system after a certain number of documents are retrieved. We use the *average precision at cutoff level*, *GPÂ=n*, to measure the average of the precision values over a set of queries when *n* documents are retrieved. Similarly, we use the *average recall at cutoff level*, *GRÂ=n*, to measure the average of the recall values when *n* documents are

retrieved. By varying the cutoff level n, we can study the effectiveness of a system using two families of such measures.

$GP\hat{A}=n$ and $GR\hat{A}=n$ describe the performance of a system when a fixed number of documents are retrieved. We could also study the performance of a system when some query-specific condition is satisfied. Let us use Rq to denote the set of documents relevant to query q, and $jRqj$ denote the number of documents relevant to query q. The *average precision at* $|Rq|$, $GPjRqj$, measures the average of the precision values when $|Rq|$ documents are retrieved over a set of queries. The average precision at half recall, GP.5, on the other hand, measures the average precision values when half of the relevant documents have been retrieved.

C.3.4. Comparison of the Recall-Precision Curves

Figures 6 and 7 depict the average precision values of 105 OHSUMED queries and 30 Medlars queries, respectively, at the eleven standard recall points 0, 0.1, 0.2, . . . , 1 for five different vector space models. The results for OHSUMED show that,

1. "Stems" is the baseline generated by the stem-based VSM. Its average 11-point average precision is $G^sP11 = 0.376$.
2. "Concepts Unrelated" is generated by using the concepts as the terms, and treating different concepts as unrelated. More specifically, we use $s^c(c, d) = \delta c,d$ in the inner product calculation (Formula (1)). The average 11-point average precision is $G^{cu}P11 = 0.336$, an 11% decrease from the baseline.
3. "Concepts" is similar to case 2, but taking the concept interrelations into consideration, we achieve a significant improvement over case 2. The average effectiveness is approximately equal to that of the baseline.
4. "Phrases, Concepts Unrelated" refers to considering contributions from both the concepts and the word stems in a phrase, but once again, treating different concepts as unrelated. By setting

$s^c(cp, dq)$ in Formula (2) to $\delta_{cp,dq}$, we achieve significant improvement over the "Concept Unrelated" case. In fact, ts average 11-point average $G^{cu}P11$, 7.1% better than the baseline.

5. "Phrases" is similar to case 4, but considering the concept interrelations, we achieve an average 11-point average precision of $G^pP11 = 0.433$, which is a significant 15% improvement over the baseline. In both cases 4 and 5, we used equal weight for the stem and the concept contributions, $f^s = f^c = 1$.

Our experimental results reveal that using only concepts to represent documents and treating different concepts as unrelated can cause the retrieval effectiveness to deteriorate (case 2). Considering the concept interrelations (case 3) or relating different phrases by their shared word stems (case 4) can both improve retrieval effectiveness. Measuring the similarity between two phrases using their stem overlaps and the relation between the concepts they represent, the phrase-based VSM (case 5) is significantly more effective than the stem-based VSM.

C.3.5 Sensitivity of Retrieval Effectiveness to f^s and f^c

To generate the two sets of recall-precision curves "Phrase, Concept Unrelated" and "Phrase" in Figure 6 and Figure 7, we used equal weight, $f^s = f^c = 1$. To study the relative importance of the stem contribution and the concept contribution in the inner product calculation, we vary the weights f^s and f^c and study the change of the average11-point average precision value $GP11$. From Formulae (4), (5) and (6), it is clear that the document similarity value depends on the ratio between f^s and f^c, not their absolute values, therefore, we vary the (f^s, f^c) from the stem-only case (1, 0), to the equal-weight phrase case (1, 1), to the concept-only case (0, 1), and study the change of the average 11-point average precision values.

Figure 8 depicts the changes of the average 11-point average precision values as the result of the change of f^s and f^c. We observe that the retrieval effectiveness measured by $GP11$ is maximized

when f^c is about the same as f^s, and, in this region, the retrieval effectiveness is not sensitive to the change of the relative importance of the stem contribution and the concept contribution.

C.3.6 Retrieval Effectiveness Comparison in Cluster-based Document Retrieval

In the previous section, we showed that the phrase-based VSM is more effective than the stem-based VSM in document retrieval using an exhaustive search. Let us consider a set of N documents. In an exhaustive search system, the similarity values between an incoming query and all the N documents need to be computed *online* before the documents can be returned to the user. Because of the relatively large computation complexity of the vector space models, such an exhaustive search scheme is not feasible for large document collections. Using hierarchical clustering algorithms, we can first construct a document hierarchy using $O(N \log N)$ *offline* document similarity computations, and return a ranked list of documents using only $O(N \log N)$ online comparisons.

We compare the stem-based VSM and the phrase-based VSM using a $O(N \log N)$ spherical k-means algorithm that has been shown to produce good clusters in document clustering [SKK00, ZK02]. The resulting document clusters are searched using top-down and bottom-up searching strategies.

Figure 9 contains the recall-precision curves of six different searching strategies on the OHSUMED data. They are the result of an exhaustive search on the 14K documents in OHSUMED. Their average 11-point average precision values are $G_{11}^s = 0.376$ and $G_{11}^p = 0.433$. The other four curves depict the retrieval effectiveness of systems when the document hierarchies are searched. Clearly, the retrieval effectiveness of the cluster-based approaches is lower than that of the exhaustive-search-based approaches. That is, by using cluster-based document retrieval, we sacrifice the retrieval effectiveness for more efficient retrieval. More importantly, using the same

searching strategy, we see that the retrieval effectiveness of the phrase-based VSM is always much better than that of the stem-based VSM. For the top-down search, $G_{11}^{s,td} = 0.235$ and $G_{11}^{p,td} = 0.283$, and for the bottom-up search, $G_{11}^{s,bu} = 0.251$ and $G_{11}^{p,bu} = 0.299$. In each case, the phrase-based VSM is about 20% more effective than the stem-based VSM. In information retrieval, if the performance improvement for a new retrial model exceeds 5% evaluated from 50 queries over an existing model, then it is considered significant enough to warrant using the new retrieval model [SM 83]. In our case, there is a 20% improvement averaged over 100 queries, representing a significant improvement.

C.4 Computation Complexity

The document similarity calculation in the phrase-based VSM is more complex than that in the stem-based VSM. Let us use *L* to represent the average length of a document. In the stem-based VSM, different word stems are considered unrelated. As a result, by building indexes on the word stems in the documents, an efficient algorithm computes the stem-based similarity between two documents using $O(L \log L)$ time. The time complexity of a straightforward implementation of the phrase-based document similarity calculation is $O(L^2)$. Different phrases in the phrase-based VSM can be related to one another not only because they may share common word stems, but also because the concepts they represent can be related. Therefore, indexing the phrases in the documents does not reduce the time complexity of the phrase-based document similarity calculation to $O(L \log L)$. To reduce the computation complexity, we need to build separate indexes on the concepts and the stems in the documents, keep track of where each stem or concept occurs, and modify the conceptual similarity storage structure. The phrase-based document similarity calculation utilizes such data structure modifications has a $O(L \log L)$ time complexity. For the OHSUMED documents, the improved phrase-based document similarity calculation is about 10

times slower than the stem-based calculation, while the straightforward implementation is over 250 times slower than the stem-based calculation.

Preliminary experimental results show that the number of related concept pairs decreases drastically as the pairwise conceptual similarity value increases. Therefore, we can further reduce the phrase-based computation complexity by treating related concepts with low conceptual similarity values as unrelated. We are currently investigating the tradeoff between the retrieval effectiveness and the computation time complexity when related concepts are treated as unrelated in the phrase-based document similarity calculations.

D. Transforming similar queries into query templates

Recent studies reveal that users' information requests in a specific domain typically follow a limited number of patterns. In the medical domain [HMW90, HBL94, EOE99, EOG00] for example, more than 60% of all the physicians' clinical questions can be classified into ten frequent categories. We can summarize the frequently asked similar queries and tailor our retrieval system according to the summarized queries. A *query template* defines the structure of a group of similar queries which consist of a key concept and scenario concept(s). Filling in the key concept values in a query template results in a specific free-text query.

To find out how to define a query template, we shall investigate a few medical queries presented in [HBL94].

Q_1: **LACTASE DEFICIENCY**, *therapy options*

Q_2: **IRON DEFICIENCY ANEMIA**, *which test is best*

Q_3: **THROMBOCYTOSIS**, *treatment and diagnosis*

By inspecting these queries, we note that each focuses on a particular disease concept, e.g., "lactase deficiency," "iron deficiency anemia," or "throbocytosis." Such disease concepts provide the

focus of each query. Further, each query asks about a specific scenario related to the disease concept. For example, Q_1 asks about the "treatment" scenario of a disease, Q_2 asks about the "diagnosis" scenario, and Q_3 asks about both. We highlight the disease concept of each query in bold, and the scenario concepts in italic.

To generalize the above sample queries, we can extract the key concepts and scenario concepts (the structural information) and transform the queries into the following templates. Note that in the templates we unify the representation of scenario concepts, e.g. mapping "therapy options" to "treatment."

T_1: <Disease and syndrome>, treatment

T_2: <Disease and syndrome>, diagnosis

T_3: <Disease and syndrome>, treatment and diagnosis

Thus, in general, each query template has two essential components:

1. The key concept. In the template, we only specify the semantic type of this concept, e.g., "Disease and syndrome." The user needs to fill in the concept value to generate a concrete query. For example, filling "lung cancer" into template T_1 results in a real query of "lung cancer, treatment." Further, the concept must belong to the semantic type defined in the template, e.g., "lung cancer" must be a "Disease and syndrome" concept.
2. One or more scenario concepts. For example, "treatment," "diagnosis," and/or "complication" of some disease concept.

In the following sections, we shall illustrate how we use the structural information in query templates to organize the key document features into a topic-oriented directory. Further, the structural information in query templates enables us to expand more scenario-specific terms to the original query and significantly improve the retrieval performance.

E. Knowledge-based scenario-specific query expansion

In this section, we will present a knowledge-based query-expansion technique to rewrite the original query into a more scenario-specific query, thus improving the retrieval performance.

As indicated in Figure 10, a class of concepts in the Metathesaurus (the lower half of the graph) is abstracted into one semantic type in the Semantic Network (the upper half of the graph). Although UMLS does not specify the potential relationships among the Metathesaurus concepts, it indicates the relationships between semantic types in the Semantic Network level. For example, UMLS does not indicate that "radiotherapy" "treats" "lung cancer." Nevertheless, "radiotherapy" belongs to "Therapeutic or Preventive Procedure" which "treats" "Disease or Syndrome," which in turn is the semantic type for "lung cancer." Using this knowledge structure, our knowledge-based query expansion automatically expands scenario concepts c_s to the original query. Let us illustrate the expansion steps using the query example "lung cancer, treatment options."

1. Navigate the key concept ck to its semantic type (e.g. from "lung cancer" to its semantic type: "Disease or Syndrome").
2. Starting from c_k's semantic type, traverse through the relationships as indicated by the original scenario concept c_s to reach a set of relevant semantic types (e.g. starting from "Disease or Syndrome," traverse through the "treats" link because the original c_s is "treatment options," and reach "Therapeutic or Preventive Procedure," "Medical Device," and "Pharmacologic Substance").
3. Append all concepts belonging to the relevant semantic types to the original query (e.g. appending all the concepts in the shaded circular areas in Figure 8).
4. Assign weights to each appended c_s based on how frequently it co-occurs with c_k in a sample corpus. A scenario concept c_s receives a higher weight if it co-occurs with c_k more often. The weights distinguish c_s that are truly semantically related to c_k (since they co-occur more often)

from those that are only marginally related. For example, for two "Therapeutic or Preventive Procedure" concepts, "radiotherapy" co-occurs with "lung cancer" more often than "heart surgery." As a result, "radiotherapy" receives a much higher weight than "heart surgery" when appended to the query "lung cancer, treatment." Details of weight computation can be found in [CL03].

Following the above procedure, we automatically derive the scenario concepts c_s to expand the queries "lung cancer, treatment options" and "lung cancer, diagnosis options," respectively. We list the fifteen of such c_s with the highest weights in Table 6. The preliminary results show that our automatic procedure is able to generate scenario-specific expansion. For example, we expand "chemotherapy" and "radiotherapy" to the query with the "treatment" scenario, and expand "brochoscropy" and "brochoscropy with biopsy" to the query with the "diagnosis" scenario.

In our preliminary experiment, we focus on five types of scenarios: "treatment," "diagnosis," "prevention," "cause" and "indication." In the standard test set OHSUMED [He94], there are 40 queries that belong to these five scenarios. We evaluate the effectiveness of the knowledge-based expansion method together with other retrieval methods for all of these 40 queries. For each retrieval method, we compute the average 11-point precision-recall curve for the 40 queries as shown in Figure 11. The curve at the bottom is for the stem VSM without query expansion. The solid curve just above the bottom line represents the statistical expansion that expands on word stems. This curve represents the best performance that traditional techniques can achieve, without using any knowledge source. The dashed line immediately above the stem expansion is produced by the phrase VSM without expansion, and the top curve of the four is the knowledge-based expansion method that uses the phrase VSM. Compared to stem VSM without expansion, the knowledge-based expansion method has achieved 33% improvement in retrieval effectiveness; compared to

the statistical stem expansion, the knowledge-based method has achieved 19% improvement. These results show that knowledge-based expansion provides significantly greater improvements for supporting scenario-specific queries than the traditional methods.

F. Test bed for evaluating the effectiveness of scenario-specific retrieval

We have implemented and integrated the three proposed techniques in a test bed to provide scenario-specific free-text retrieval (Figure 12). This system provides the capability to retrieve many types of medical free-text documents, e.g., patient clinical reports, medical literature articles, etc. IndexFinder will first extract key concepts and normalize them into standard terms as defined in the knowledge source (e.g., UMLS)..

During the retrieval phase, the query expansion module appends the user query with scenario-specific terms. Documents are ranked based on their similarity to the query via the phrase-based Vector Space Model (VSM) and returns them to the users.

G. Summary

We have developed a new knowledge-based approach to retrieving scenario-specific free-text documents, which consists of three integrated components: *IndexFinder*, phrase-based VSM and knowledge-based query expansion. *IndexFinder* can extract key terms from free-text, generating conceptual terms by permuting words in a sentence rather than through the traditional NLP-based technique. Although the generated concepts are matched with the controlled vocabulary in the ULMS and are valid terms, they might not be relevant to the document. Thus, syntactic and semantic filters are used to eliminate the irrelevant candidates. Preliminary evaluation shows that filtering is effective in eliminating irrelevant concepts and the semantics in the section headings in a docu-

ment are useful for guiding semantic filter selection. Our experimental results show that *IndexFinder* can process free-texts at a speed of about 43K bytes of text per second on a PC with Pentium 4. As a result, it is able to extract key UMLS concepts from clinical texts in real time. The extracted concepts can be used for content correlation, document indexing , and transforming ad hoc terms in the queries into controlled vocabulary to improve retrieval effectiveness.

A new vector space model, the phrase-based VSM, has been developed for document retrieval. In the phrase-based VSM, we divided each document into a set of phrases. Each phrase is represented by both a concept defined in the controlled vocabulary and the corresponding word stems. The similarity between concepts is based on the interrelationships of concepts in the knowledge base. The similarity between two phrases is measured by their stem overlaps as well as the similarity between the concepts they represented. The similarity between two documents is defined as the cosine of the angle between their respective phrase vectors.

Using UMLS as both the controlled vocabulary and the knowledge base to derive the conceptual similarities, we demonstrated from different perspectives that the retrieval effectiveness of the phrase-based VSM was significantly higher than that of the current gold standard – the stem-based VSM. This is because in phrase VSM, the stem similarity compensates for the incompleteness of knowledge sources, while the concept similarity compensates for the lack of semantic meaning in the stem similarity. Such a significant increase in retrieval effectiveness was achieved without sacrificing excessive computation efficiency. Knowledge-based query expansion expands terms related to the scenario and is able to provide scenario-specific query answering and retrieve content correlated medical documents.

We have implemented a test bed with the above three technologies. Using the UCLA patient reports as a test set, we are currently investigating the methodology for generating the topic oriented directory system from document features as well as evaluating the effectiveness of our approach

for retrieving scenario-specific ad hoc queries and performing content correlation of medical documents.

Acknowledgements

This research is supported by NIC/NIH Grant #4442511-33780 and NSF Grant # IIS-0097438

References

[Aro01] Alan R. Aronson, Effective Mapping of Biomedical Text to the UMLS Metathesaurus: The MetaMap Program. In Proc. of AMIA Annual Symp 2001, 2001.

[AC75] A.V. Aho and M.J. Corasick. Efficient String Matching: an Aid to Bibliographic Search. In *CACM*, 18(6),330-340, 1975

[BP98] S. Brin, L. Page, The Anatomy of a Large-Scale Hypertextual Web Search Engine, In *Proceedings of WWW '98*, 1998

[BWB01] Bui, A., Weigner, G. S., Barretta, S. J., Dionisio, J. D., McCoy, M. J. An XML gateway to patient data for medical research applications. In Proceedings of the 2001 International Conf. on Mathematics and Engineering Techniques in Medicine and the Biological Sciences (METMBS '00), Las Vegas, NV, 2001

[Chu98a] W.W. Chu. Cooperative Information Systems. Encyclopedia of Electrical and Electronic Engineering, edited by J. G. Webster, John Wiley & Son, Inc., 1998

[Chu02] S. Chu. Yearbook of Medical Informatics, 2002

[CHC98b] W.W. Chu, C. Hsu, A.F. Cárdenas, and R.K. Taira. Knowledge-based image retrieval with spatial and temporal constructs. IEEE Transactions on Knowledge and Data Engineering, 10(6): 872-888, 1998

[CL03] W.W. Chu, and Z. Liu. A knowledge-based approach for scenario-specific content correlation in a Medical Digital Library. UCLA Computer Science Technical Report, # 030039, 2003

[DSS02] Joshua C. Denny, Jeffrey D. Smithers, Anderson Spickard, III, Randolph A. Miller. A New Tool to Identify Key Biomedical Concepts in Text Documents. In Proc. of AMIA Annual Symp 2002, 2002.

[Eft96] E.N. Efthimiadis. Query expansion. Annual Review of Information Science and Technology, 31:121-187, 1996.

[ECA88] Elkin PL, Cimino JJ, Lowe HJ, Aronow DB, Payne TH, Pincetl PS and Barnett GO. Mapping to MeSH: The art of trapping MeSH equivalence from within narrative text. In Proc. 12th SCAMC, 185-190, 1988.

[EOE99] J.W. Ely, J.A. Osheroff, M.H. Ebell, G.R. Bergus, et al. Analysis of questions asked by family doctors regarding patient care. British Medical Journal, 319:358-361, 1999

[EOG00] J.W. Ely, J.A. Osheroff, P.N. Gorman, M.H. Ebell, et al. A taxonomy of generic clinical questions: classification study. British Medical Journal, 321:429-432, 2000

[FH98] C. Friedman and G. Hripcsak. Evaluating natural language processors in the clinical domain. Methods of Information in Medicine, 37(4/5): 334-344, 1998.

[Gus93] Dan Gusfield, Algorithms on Strings,Trees, and sequences: Computer Science and Computational Biology, Cambridge University Press, 1997

[HBL94] W. Hersh, C. Buckley, T.J. Leone and D. Hickam. OHSUMED: an Interactive Retrieval Evaluation and New Large Test Collection for Research. In Proc. 17th ACM-SIGIR, pages 191-197, 1994

[HMW90] R. Haynes, K. McKibbon, C. Walker, N. Ryan, D. Fitzgerald, and M. Ramsden. Online access to MEDLINE in clinical settings. Ann Intern Med, 112:78-84, 1990

[IOM01] Crossing the Quality Chasm: A New Health System for the 21st Century, Institute of Medicine, 2001

[IV98] Nancy Ide and Jean Veronis, Word Sense Disambiguation: The State of Art, Computational Linguistics, 24(1): 1-40, 1998

[JC94] Y. Jing and W.B. Croft. An association thesaurus for information retrieval. In Proc. RIAO'94, pages 146-160, 1994

[JCH92] J.P. Callan, W.B. Croft, S.M. Harding, The INQUIRY Retrieval System, In *Proceedings of DEXA '92*, 1992.

[Lov68] J.B. Lovins. Development of a stemming algorithm. Mechanical Translation and Computational Linguistics, 11 1-2):22-31, 1968

[Lyo77] John Lyons. Semantics, Cambridge University Press, 1977

[Man99] R. Mandala, T. Tokunaga, H. Tanaka. Combining Multiple Evidence from Different Types of Thesaurus for Query Expansion, In Proc. 22nd ACM-SIGIR, pages 191-197, 1999

[MC02] W. Mao and W.W. Chu. Free-text medical document retrieval via phrase-based vector space model. In Proc. of AMIA Annual Symp 2002, 2002

[NLM01] National Library of Medicine. UMLS Knowledge Sources, 12th edition, 2001

[NLM02] http://www.nlm.nih/gov/pubs/factsheets/

[NLM03] National Library of Medicine, UMLS Knowledge Sources, 14th edition, 2003

[QF93] Y. Qiu and H.P. Frei. Concept-based query expansion. In Proc. 16th ACM-SIGIR, pages 160-169, 1993

[Rij79] C.J. van Rijsbergen Information Retrieval, Butterworths, 1979

[Sal 75] G. Salton A new comparison between conventional indexing(MEDLARS) and automatic text processing (SMART). J. of the American Society of Information science, 23(2):74-84, March-April 1975

[SKK00] M. Steinbach, G. Karypis, and V. Kumar, A comparision of document Clustering Techniques, In Proc of the KDD Work Shop on Text Mining, 2000

[SM83] G. Salton and M. J. McGill, Introduction to Modern Information Retrieval, McGraw-Hill Computer Science Series, McGraw-Hill, Inc, 1983

[SO94] A. Stuart and J.K. Ord. Kendall's Advanced Theory of Statistics, Sixth Edition, London, Edward Arnold, 1994

[SRH02] Suresh Srinivasan, Thomas C. Rindflesch, William T. Hole, Alan R. Aronson, and James G. Mork. Finding UMLS Metathesaurus Concepts in MEDLINE. In Proc. of AMIA Annual Symp 2002, 2002

[SWY75] G. Salton, A. Wang, and C. S. Yang, A Vector Space Model for Automatic Indexing Communication of the ACM, 18(11): 613-620, 1975

[TOK98] Tuttle MS, Olson NE, Keck KD, Cole WG, Erlbaum MS, Sherertz DD et al. Metaphrase: an aid to the clinical conceptualization and formalization of patient problems in healthcare enterprises. Methods Inf Med. 1998 Nov, 37(4-5): 373-83.

[Voo93] E.M. Voorhees. On expanding query vectors with lexically related words. In Proc. TREC-2, pages 223-232, 1993

[XC96] J. Xu and W.B. Croft. Query expansion using local and global document analysis. In Proc. 19th ACM-SIGIR, pages 4-11, 1996

[ZB97] Yuri L. Zieman and Howard L. Bleich. Conceptual Mapping of User's Queries to Medical Subject Headings. In Proc. of AMIA Annual Symp 1997, 1997

[ZC03] Q. Zou, W.W. Chu, Craig Morioka, Gregory H. Leazer, and Hooshang Kangarloo, *IndexFinder*: A Knowledge-based Method for Indexing Clinical Texts. AMIA Annual Symp 2003

[ZK02] Ying Zhao and George Karypis Evaluation of Hierarchical Clustering Algorithms for Document Datasets, TR 02-022, Dept. of Computer Science, U. of Minnesota, 2002

Table 1. Problems with mapping noun phrases individually.

Example	Text
1	**Prostate**, right (biopsy) - fibromuscular and glandular **hyperplasia**
2	A small **mass** was found in the **left** hilum of the **lung**.

Table 2. Using UMLS semantic type to define interests

Brain Tumor Characteristics	**Relevant UMLS semantic types**
Specific Cancer	Neoplastic Process
Medical Intervention	Therapeutic Procedure
Anatomical location	Body Part, Organ or Organ Component

Table 3. Output from *IndexFinder* for the text in Figure 5

Semantic Descriptor	**ULMS Concept**
T191:Neoplastic Process	C0025286:meningioma
T047:Disease or Syndrome	C0014068:encephalomalacia

Table 4. Comparison of OHSUMED and Medlars statistics. Noticeable differences are shown in italic fonts

	OHSUMED		Medlars	
	Query	Document	Query	Document
Number of Documents	*105*	*14,430*	*30*	*1,033*
Phrases per Document	*7.5*	112	*11*	90
Stems per Phrase	1.34	1.25	1.25	1.14
Concepts per Phrase	1.21	1.18	1.27	1.21
Multi-stem Phrases per Document	1.96	*21.3*	2.6	*10.8*
Multi-sense Phrases per Document	*1.2*	11.3	*2*	9.8

Table 5. Comparison of the expansion terms derived by the statistically co-occurring and knowledge-based query expansion method

Stem
patient
Stud
Tumor
pulmon
Diseas
Result
Therap
Treat
carcinom
Effect
Year
Breast
increas

(a) Expansion terms for "lung cancer" derived from co-occurrence

Concept ID	Concept String
C0015133	Etoposide
C0032284	Pneumonectomy
C0013216	Drug Therapy / Chemotherapy
C0008838	Cisplatin
C0039991	Thoracotomy
C0025065	Mediastinoscopy
C0038903	Surgery, lung
C0034618	Radiotherapy
C0281477	Lung cancer screening
C0079172	Cranial irradiation
C0048420	4-ipomeanol
C0042682	Vindesine
C0023928	Lobectomy

(b) Expansion terms for "treatment of lung cancer."

Concept ID	Concept String
C0015133	Etoposide
C0008838	Cisplatin
C0039991	Thoracotomy
C0025065	Mediastinoscopy
C0027646	Neoplasm staging
C0048420	4-ipomeanol
C0042682	Vindesine
C0013089	Doxorubicin
C0010583	Cyclophosphamide
C0051733	Amonafide
C006290	Bronchoscopy
C0063067	Hydrazine sulfate
C00189396	Bronchoscopy with biopsy

(c) Expansion terms for "diagnosis of lung cancer."