14 KMeX: A KNOWLEDGE-BASED DIGITAL LIBRARY FOR RETRIEVING SCENARIO-SPECIFIC MEDICAL TEXT DOCUMENTS

Dr. Wesley W. Chu (PhD), Dr. Zhenyu Liu (PhD), Dr. Wenlei Mao (PhD) and Dr. Qinghua Zou (PhD)
Computer Science Department, University of California, Los Angeles

Medical records, such as patient records, lab reports, literature articles, newsletters, etc. are in free-text form and often time, medical practioners wish to retrieve these scenario-specific documents. A scenario typically refers to a specific health care task, such as, searching for treatment methods for a specific disease. Although, traditional information retrieval systems are useful for retrieving general documents, these systems cannot support scenario-specific information retrieval because:

1. The terms in the query posed by the user may not use a standardized medical vocabulary.
2. The lack of an effective technique to represent synonyms, phrases and similar concepts in free-text.
3. The mismatch between the terms used in a query and those used in a document for representing the same topic.

In this chapter, we present a new knowledge-based (e.g. UMLS) approach to mitigate these problems. More specifically, we propose to use the metathesaurus and semantic structure in the UMLS to extract key concepts from a free-text for: 1) indexing; 2) phrase-based indexing for representing similar concepts, and 3) query expansion to improve the probability of matching query terms with the terms in the document. To do so, the system formulates the query based on the user's input, and selects scenario templates such as "disease, treatment" or "disease, diagnosis." Thus, the system is able to retrieve relevant documents for a specific scenario. Furthermore, a topic (A group of co-occuring) oriented directory is proposed which is generated based on query template, frequently occurring relevant topics in a document. Such a directory system not only selects a set of relevant documents in respect to the query template but also provides cross-reference among related topics. These techniques have been implemented in a test bed at UCLA. Using the standard OSHMED corpus, our empirical results validate the effectiveness of this new approach over the traditional text retrieval techniques.

14.1 INTRODUCTION

Medical information knowledge and clinical data are 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 entries per month. Today, the volume of bibliographic citations is growing at 1,000 per day in Medline alone [1]. Hospitals also generate large amounts of healthcare data that are stored on computers. Hence, the delivery of quality healthcare to consumers requires the availability and accurate information retrieval from this large information sources. The demand for the use of evidence-based practices to help improve the quality of care also adds great amounts of pressure on healthcare professionals to regularly access the highest quality information during the healthcare planning, decision and delivery. Today, computer-assisted information retrieval and processing are necessary to support quality decision-making and to help overcome human cognitive constraints [2].

A Medical Digital Library consists of three types of data: 1) structure data, such as patient lab data and demographic data; 2) multi-media images, such as MRIs; and 3) 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 [3-4]. However, many medical records are in free-text form and usually require scenario-specific retrieval. 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).

From the above scenario, specific queries cannot be effectively supported by traditional information retrieval systems because the lack of indexing for free text, ranking the similarity of the content within the document with the query term and a method to resolve the mismatch of the term in the query with that in the document. We developed the following knowledge-based techniques to ameliorate the above problems.

Extracting key concepts from free-text
We have developed a new technique to automatically extract key concepts from a free-text and to permute the set of words in the input free-text, thereby generating all valid concepts defined by the controlled vocabulary in a knowledge base (e.g. UMLS). Since the generated valid concept may not be relevant to the query, syntactic and semantic filters are then used to filter out the irrelevant concept. Thus, retrieval efficiency is improved because key concept terms can be used as indices in a free-text directory system, as well as transforms the ad hoc terms in the query into a controlled vocabulary.

Phrase-based vector space model (VSM)
Vector Space Models (VSM) are commonly used to measure the similarity between a query and a document. Traditional stem-based VSM cannot match terms in the query with those used in the documents that have a similar meaning but different expressions. We developed a knowledge-based/phrase-based VSM [5], which identifies terms with similar meanings, and represents them based on both concepts and stems. As a result, phrase-based VSM yields significantly better retrieval performance than the stem-based VSM.

Knowledge-based query expansion
Queries can be appended with related terms to increase the probability of matching the terms in the query with those of relevant documents. Traditional expansion techniques append all statistically co-occurring terms into the original query, many of the expanded terms 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.

14.2 EXTRACTING KEY CONCEPTS FROM DOCUMENTS

14.2.1 The UMLS knowledge source

Since our approach is leveraged on knowledge bases, we shall first briefly describe the Unified Medical Language System's (UMLS) [6] knowledge sources, 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 and 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 the Metathesaurus as the controlled vocabulary to detect concepts and to derive the conceptual relations using the hyponym 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 14.6).

14.2.2 Indexing for Free-text Documents

Indexing free-texts is a difficult task since the writing in the free-text does not use a controlled vocabulary. Further, similar concept terms, synonyms, and etc. in the free-text add an additional level of difficulty to such a task. This also applies to ad hoc queries which can also be viewed as documents. Unlike medical literature, where the author(s) provide key words, 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 a topic oriented directory system for free-text where the document can be obtained 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 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 etc.

Therefore, we developed indexing systems based on 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 been dedicated at developing effective methods for mapping free-text into UMLS concepts. Examples of such efforts include SENSE [7], MicroMeSH [8], Metaphrase [9], KnowledgeMap [10], PhraseX [11], *MetaMap* [12]. 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. This approach achieves some success, however, there are two major weaknesses to this general technique:

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

Table 14.1. Problems with mapping noun phrases individually.

First, some important concepts can never be discovered through the identification of noun phrases. Table 14.1 provides examples of texts that reveal the shortcomings of the use of noun phrases.

- Example 1: A word from the first line with a word from the second line forms the key concept, "prostate hyperplasia," which corresponds to concept ID 33577 in the UMLS Metathesaurus.
- Example 2: 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.

Second, NLP requires significant computing resources. As a result, most of the NLP systems work in an offline mode, and 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.

14.2.3 IndexFinder [13]

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 and concept terms may contain negative sense and may not be relevant to the query, negation detection is used to identity negative concepts. Further, syntactic and semantic filters, which are based on a specific scenario are used to filter out irrelevant concepts.

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.

Negation Detection

Negation detection is an important task in medical document processing since whether or not a medical symptom presented can make totally different diagnoses for a disease. If a doctor searches for the concept "no cough", retuning the concept "cough" is considered to be irrelevant. To handle the negation problem in IndexFinder, we first define a list of terms and negation hues, which carry negative sense for a concept. Then, we identify the UMLS semantic types which can be negated. Finally, for the concepts of the above defined semantic types, we combine them with possible negation hues according to certain defined rules. More specifically, IndexFinder relays on the three parts for negation detection:

- *Negation hues list*: specifies the list of words which tends to negate a concept in a sentence. For example, in medical reports, the words, *no*, *not*, *isn't*, etc are frequently used for negation.
- *UMLS semantic types qualified for negation*: specifies the list of UMLS semantic types which can be negated. For example, the semantic type T191 (*disease/cancer*) is qualified for negation since the concepts related to T191 can appear in patient records in negation form.
- *Rules for negating concepts in a sentence*: specifies the rules to negate UMLS concepts when negation hues are presented in the same sentence where the concepts are extracted. For example, "*no*" tends to negate the concept immediately followed; when multiple concepts are qualified for negation, the concept closest to the negation hue is selected for negation.

Figure 14.1 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.

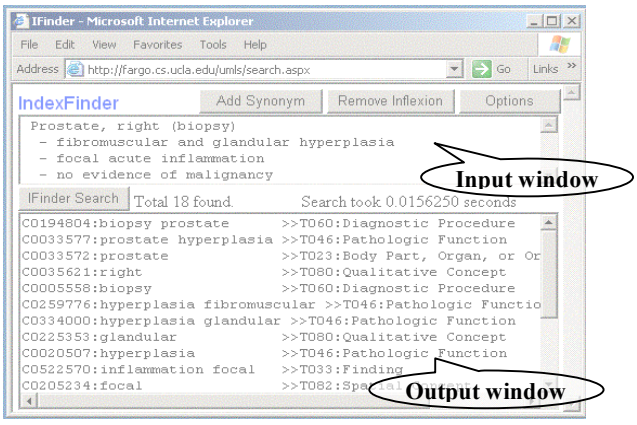


Figure 14.1. *IndexFinder* web interface.

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 knowledge source, and syntactic or semantic information from the original sentence to filter out irrelevant concepts. For example, if a physician wishes to know what kind of diseases a patient suffers, it is more desirable to return disease related UMLS phrases rather than returning all concepts to the physician. We consider six types of filters as shown in Figure 14.2.

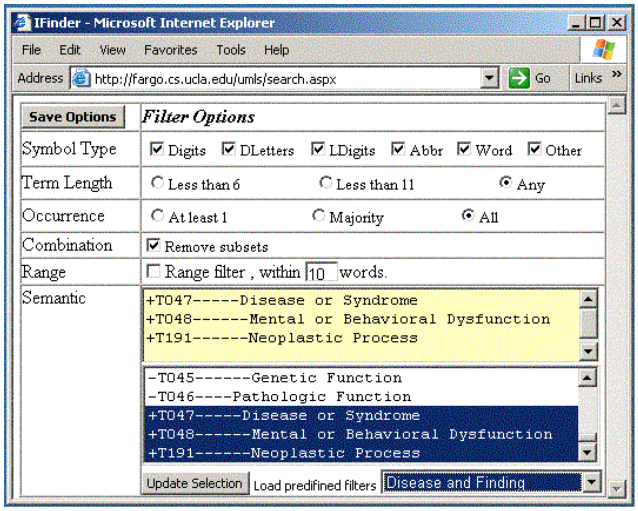


Figure 14.2. Filter Selection

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, he can simply not check the Digits box as in Figure 14.2.
- *Term Length filter*: specifies the length limitation of candidate phrases.
- *Coverage filter*: specifies the coverage condition for a candidate phrase. It has three options, *at least one*, *majority*, and *all*. By default, the option *all* is 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 other phrases. For example, if the 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*: to remove 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, as shown in Figure 14.2, the user can select *Disease or Syndrome* and its two sub types, so that the resulting phrases will be of these two types. As a result, the filter also eliminates those irrelevant phrases from the set of

phrase candidates. Note that UMLS ISA relationship may also be used to filter out more general phrases.

Figure 14.3 shows the filtering result for the sample input in Figure 14.1, (also depicted at the top of Figure 14.3). 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 wishes to know about body parts or spatial characteristics. Prostate biopsy is the only diagnostic procedure used.

Input:	Prostate, right (biopsy) - fibromuscular and glandular hyperplasia - focal acute inflammation - no evidence of malignancy
Filtering	***Results***
Subset	C0194804:biopsy prostate C0033577:prostate hyperplasia C0035621:right C0259776:hyperplasia fibromuscular C0334000:hyperplasia glandular C0522570:inflammation focal C0333361:inflammation acute C0391857:no malignancy evidence
Pathologic Function (T046)	C0033577:prostate hyperplasia C0259776:hyperplasia fibromuscular C0334000:hyperplasia glandular C0333361:inflammation acute
Body parts & Spatial (T023, T082)	C0033572:prostate C0205234:focal
Diagnostic Procedure (T60)	C0194804:biopsy prostate

Figure 14.3 Key concepts after filtering

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 reports. There are total 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 14.4.

We are currently in the process of further evaluating the accuracy of our method. We plan to generate a test dataset by randomly selecting a set of topic sentences from the above 5,783 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 [14].

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

Input: *A small mass was found in the left hilum of the lung.*

IndexFinder Results:

C0024873:a mass >>T190:Anatomical Abnormality
C0700321:small >>T080:Qualitative Concept
C0746117:mass lung left >>T033:Finding
C0332285:found >>T082:Spatial Concept
C0225733:lung left hilum >>T029:Body Location or Region

MetaMap Results:

Phrase: "A small mass"
861 Mass, NOS [Anatomical Abnormality]
694 Small [Qualitative Concept]
Phrase: "was" Meta Mappings: <none>
Phrase: "found" Meta Mappings: <none>
Phrase: "in the left hilum"
1000 Left hilum [Body Part, Organ, or Organ Component]
Phrase: "of the lung"
1000 Lung [Body Part, Organ, or Organ Component]
1000 Lung <3> (Lung diseases) [Disease or Syndrome]

Figure 14.4. Comparing results generated by *IndexFinder* and *MetaMap*.

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 documents consist of primary care clinical notes, specialist clinical notes, pathology reports, laboratory results, radiology reports, and surgical notes. Figure 14.5 shows an excerpt from a radiology report.

"The right frontal convexity meningioma is slightly larger now than on the prior examination. The left frontal meningioma is unchanged. There are three other small enhancing nodules seen along the frontal convexities bilaterally, as described above. There are no new lesions seen. There is no mass effect caused by these lesions. There is bifrontal encephalomalacia."

Figure 14.5. Free-text 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 14.2.

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

Table 14.2. Using UMLS semantic type to define interests

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 14.3. The two concepts, in fact, are important in the excerpt and thus are good terms for indexing.

Semantic Descriptor	UMLS code
T191:Neoplastic Process	C0025286:meningioma
T047:Disease or Syndrome	C0014068:encephalomalacia

Table 14.3. Output from *IndexFinder* for the text in Figure 14.5

14.3 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 [15-18] 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. This motivates us to introduce the notion of a *query template*. 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 focuses 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 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 concept and scenario concepts (the structural information) and transform 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:

a) 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.

b) 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.

14.4 TOPIC-ORIENTED DIRECTORY

To improve the efficiency of free-text document-retrieval in terms of precision and recall of the request documents and to provide cross reference among related topics, we shall propose to develop a directory system that is based on user queries and topic/sub-topic hierarchies derived from key features of the documents.

Using our IndexFinder, we are able to automatically extract a set of key features to represent a document. Next we will use data-mining techniques to identify frequently co-occurring key features. Each group of frequent features can be viewed

as a directory topic. Since these topics are directly derived from the document content without any generalization, we consider them to be the most specific ones in the directory. Therefore they are placed at the leaf-level of the topic hierarchy. Starting from the most specific topics, we merge these sub-topics into more general topics. By continuing this process, eventually a topic hierarchy can be constructed. In order for the merging process to be semantically meaningful, it will be guided by the Semantic Network in the knowledge base (e.g., UMLS). For the topic hierarchy to be sensitive to directory users, we should reorganize the hierarchy also based on the user querying patterns. One way to achieve this is to adjust the hierarchy so that it corresponds to the frequent browsing patterns from general to specific topics. This can be accomplished by modifying the knowledge hierarchies in the semantic network in accordance with the query granularity to form directory paths and by concatenating these directory paths in the drill-down browsing patterns. As a result, the topics and subtopics in the directory hierarchy are derived based on key features in the documents, as well as on user query patterns.

Such a directory design differs from existing document clustering techniques in the following ways. First, our directory topics are derived from the documents and represented by control vocabulary from a knowledge source. In conventional document clustering, each tree node only represents a subgroup of documents without any semantic meaning. Second, our directory topics are generated from mining the document key features as well as the user queries (query templates). As a result, our directory system can adapt to different types of queries and is user-sensitive. Existing document clustering techniques do not consider information related to frequent query patterns and user type. Third, the directory topic hierarchy is organized by the guidance of the semantic network in the knowledge source, e.g., UMLS, and therefore is well defined. The resulting directory structure has more semantic meaning than the statistical approaches and thus is able to provide scenario-specific indexing and improve document retrieval performance.

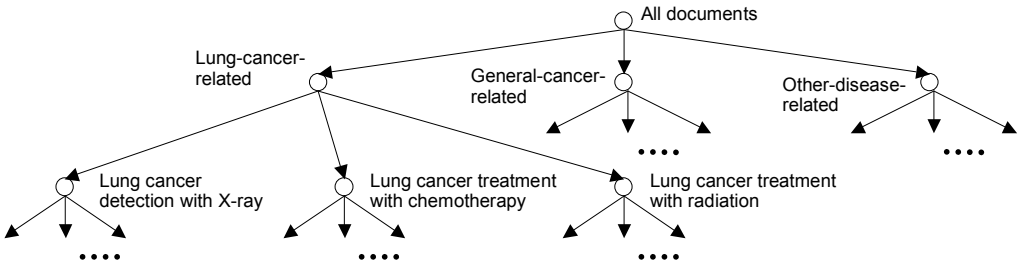


Figure 14.6. A sample directory system for a lung cancer physician

Let us illustrate the process of organizing a topic-oriented directory system by the following example. Given a large corpus of documents related to disease, we will design a directory system for lung cancer physicians. Based on their interests, most of the query will be related to lung cancer; that is, the diagnosis, treatment, risk

factors etc., of lung cancer. As a result, the document collection for these particular physicians can be divided into three topics: lung cancer-related, general cancer-related, and other disease-related documents, as shown in Figure 14.6. Through data-mining of the key features of these documents, we are able to derive the following list of topics from the above broader topics: lung cancer, diagnosis, treatment, risks of cancer, chemotherapy, surgery, radiation, etc. Such topics and subtopics can be organized with the guidance of the semantic network of UMLS. For example, the topic "lung cancer" can be further divided into the various subtopics such as "diagnosis," "treatment," and "risk factors of cancer." Then, based on the knowledge source, the subtopic "treatments" can organized into the sub-subtopics: "chemotherapy," "surgery," and "radiation." Since the topics are derived from the key features of the documents, such topics and subtopics can be indexed to represent scenario-specific topics.

Note that the directory is organized based on of a given user query (query template), as well as topics and subtopics that derive from the key features of documents. Thus, the directory system not only can provide scenario-specific document retrieval, but it can also improve document retrieval performance. Likewise, we can organize the directory system for different user query templates. These different directory systems can be linked and formed into a general directory system for the set of query templates. Nodes in the directory of a query may overlap with nodes in the directories of some other queries. Such overlap provides cross-references of topics and increases the search scope of the nodes (topics). For a given query, the system will navigate according to its directory to retrieve the documents. The overlap nodes may provide cross-references to different scenarios in other directory systems. In order to restrict the cross-reference topics, the user can provide a certain range of topics of interest. As a result, the directory navigator will only branch to these topics. Such focused cross-referencing can increase the search scope while providing focused expansion of topics and improving retrieval performance.

14.4.1 Deriving Frequent Directory Topics via Data-Mining

Using IndexFinder, we can extract a set of key features from each document. Each feature is a concept defined in the controlled vocabulary of the knowledge source (e.g., the Metathesaurus in UMLS). In this section, we will present a data-mining technique to discover topics from document features for directory construction.

A topic can be viewed as a condensed synopsis of a sub-collection of documents. For example, "lung cancer and chemotherapy" is a topic that covers all the documents on the treatment of "lung cancer" with "chemotherapy." To capture the meaning of a subset of documents, we typically need multiple concepts, e.g., "lung cancer" and "chemotherapy." Therefore, a specific topic should consist of multiple concepts. Further, the concepts that belong to one topic should frequently co-occur in the documents. For example, it is meaningless to combine "back pain" and "heart surgery" within a topic, because very few medical documents mention both concepts.

Since a topic is a group of concepts that frequently co-occur in documents, we propose to use frequent item-set mining techniques [19-21] for topic discovery. To map topic discovery into a frequent item-set mining problem, we shall view each

document as a market basket, and the concept features extracted from that document as the items in the basket. To use the data-mining techniques for topic discovery, we need to specify a minimum support number. In the topic-discovery context, this minimum support is the minimum number of documents that we want to group under each topic. For example, if any topics in our directory cover at least five documents, then we should set the support level at "5." Table 14.4 further illustrates the mapping between topic discovery and data-mining.

If we discover that each topic is a group of frequently co-occurring concepts, any sub-portion of that group must also be frequent. That is, any sub-portion of a topic is also a valid topic. For example, if we discovered a topic {"lung cancer," "detection," "biopsy"} as a group of frequent concepts, then sub-groups such as {"detection," "biopsy"} must also be valid topics. Super-groups of concepts have more specific meanings than sub-groups, e.g., {"lung cancer," "biopsy," "detection"} is more specific than {"biopsy," "detection"}. Therefore, it would be desirable to keep only the topics that are super-groups instead of those of the sub-groups. To efficiently discover these super-groups, we need a specialized data-mining technique called "maximum frequent item-sets (MFI)" mining. We have developed a general-purpose MFI mining algorithm, SmartMiner, which can handle extremely large datasets [21]. We plan to apply this technique to discover topics that have the longest and the most specialized form and use this to construct a more accurate directory system.

Document	Market basket
Concepts in a document	Items in a basket
Topic as a group of frequently co-occurring concepts	Frequent item-set
Minimum number of documents under each leaf	Minimum support for frequent item-sets

14.4.2 Organizing Topics into a Hierarchical Directory Structure

By mining frequent co-occurring features in the document collection, we obtain a list of topics as well as the corresponding set of documents covered by that topic. We shall build a hierarchical structure from these topics for efficient retrieval of relevant documents. Since topics derived from mining frequent document features are the most specific ones in the hierarchy, they are placed at the leaf level. Starting from these most specific topics, we can construct a topic hierarchy by iteratively merging sub-topics into more general ones.

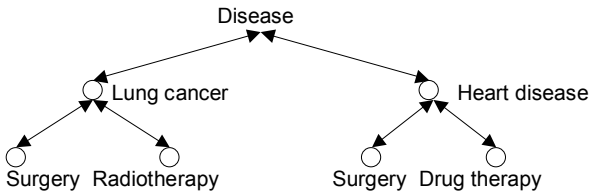
To construct a scenario-specific and query-sensitive hierarchical directory, we will leverage using knowledge source, UMLS, and the query templates. The knowledge source organizes its concepts in a general-to-specific fashion. For example, "lung neoplasm" is a more general term than "lung cancer," and "lung cancer" is more general than "non-small-cell lung cancer." This provides useful guidance to determine the general-to-specific relationships among directory topics.

For example, "lung cancer with chemotherapy" will be considered more general than "non-small-cell lung cancer with chemotherapy."

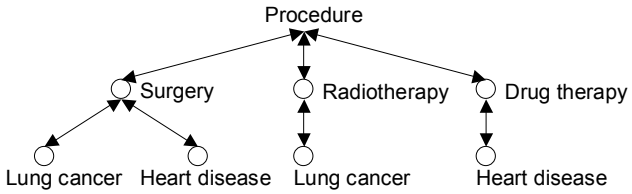
UMLS defines multiple hierarchies of concepts. Each UMLS concept hierarchy focuses on one semantic type of concept. For example, the disease-concept hierarchy represents the general-to-specific relationships among all the "Disease and Syndrome" concepts. Similarly, the procedure-concept hierarchy focuses on all "Therapeutic and Preventive Procedure" concepts. The information in query templates can be used to select the appropriate candidate hierarchy.

Let us consider the following example. Suppose that we have discovered four specific topics by mining the key features of the documents:

1. "lung cancer, surgery"
2. "lung cancer, radiotherapy"
3. "heart disease, surgery"
4. "heart disease, drug therapy"



(a) Directory structure derived from the disease-concept hierarchy



(b) Directory structure derived from the procedure-concept hierarchy

Figure 14.7. Different directory structures derived from different set of query templates

Following UMLS's *disease-concepts hierarchy*, entries 1 and 2, and 3 and 4 are two pairs of similar topics. At a higher level, both of these two topics fall under a general topic called "disease." The resulting directory structure is shown in Figure 14.7(a). If we use UMLS's *procedure-concept hierarchy*, the resulting directory structure is shown in Figure 14.7(b). We shall leverage the query templates information to select the appropriate candidate structure.

Recall that a query template consists of two parts: a key concept and a set of scenario concepts. For a particular user type, we can identify the set of frequently-used query templates. Suppose the key concepts in these frequent templates are

"<Disease and syndrome>," i.e., many templates are seen as "<Disease and syndrome>, treatment," or "<Disease and syndrome>, diagnosis," etc. Now consider a sample query constructed by the template, "lung cancer, treatment." In the initial step, we want the directory to guide us to a single branch that is all about "lung cancer." Underneath that single branch, we want to further focus on the treatment subtopic. Clearly the first structure (Figure 14.7(a)) serves this need better than the second one (Figure 14.7(b)). On the other hand, if the key concept in the query templates is of type "<Therapeutic and Preventive Procedure>," then the second structure will be more preferable than the first one.

14.4.3 Navigating the Topic-Oriented Directory

The topic-oriented directory is constructed by the set of topics that are generated by data-mining, query templates of a particular user type, and the semantic structure of the knowledge source. With the directory system, identifying a set of relevant documents for a given query is equivalent to selecting a path in the hierarchy to navigate to a leaf node. The path selection should be based on user type and query templates. Further, the directory enables us to easily navigate to broader topics related to the query. For example, if we use the directory in Figure 14.7(a) to answer the query "lung cancer treatment with surgery," we first select the path "disease" →"lung cancer" → "surgery" to reach a subset of documents. Thereafter, we can suggest further reading in the closest path "disease" →"lung cancer" → "radiotherapy."

Multiple directory structures are constructed from the query templates for multiple user types. The commonality in query templates results in overlapping nodes of various directory structures. Such overlapping nodes provide cross-referencing points among multiple directories and enlarge the search scope. For example, the leaf node for "disease" →"lung cancer" → "surgery" in Figure 14.7(a) overlaps with the leaf node "procedure" → "surgery" → "lung cancer" in Figure 14.7(b). Depending on the user's preference, the system may decide the direction and the scope of cross-referencing. For example, a lung cancer oncologist may also be interested in the topics of the treatment procedure and in the etiology and development of the patient's disease, but not other diseases such as mental illness. Note that for the most general cases, the navigation path may include generalization (going upward). We propose to use query, user type, and topic hierarchy in the directory to generate and control the navigation path that provides scenario-specific document retrieval.

14.4.4 An Example

We should use the 5000 UCLA Medical reports to construct the knowledge hierarchies and a sample topic directory. Directly following the *Parent of* relationships in UMLS, we extract all the possible knowledge hierarchies (or knowledge paths). Such paths cannot be directly used in our directory system for two reasons. First, using all the knowledge paths for our directory system design is infeasible since the number of knowledge paths in UMLS for a concept can be large. Second, the granularity can be too detail for a set of documents and thus, we need to simplify the knowledge hierarchies as follows:

- Select a proper source for a knowledge type. UMLS Semantic Network defines about 200 knowledge types (or semantic types) such as *disease*, *treatment*, *body part*, etc. For each knowledge type, a domain expert can identify the best knowledge source. For example, ICD-9 can be a good source for *disease* knowledge hierarchies. By applying a source selector for a knowledge type, we significantly reduce the number of knowledge paths for a concept.
- Combine nodes in knowledge paths that contain synonyms. Patent reports may possesses synonym concepts. To reduce the number of knowledge paths we combine the parents of synonym concepts to a synonym group and assign a concept for the synonym group in the knowledge paths.
- Reduce the number of knowledge paths. Remove the nodes in the knowledge paths that contain only a single child node to the topic concepts. For a specific document set, topics can be derived by data mining the MFIs. The set of concepts that contains the topics are called topic concepts. For each topic concept, we can extract knowledge paths from UMLS, and we compute the number of descendant nodes in the path. All path nodes with a single child will be removed for the simplicity of the topic directory.

Using the three techniques, we can extract knowledge hierarchies from UMLS and simplify them for our directory system design. For example, Table 14.5 shows a portion of the body part knowledge hierarchies we have extracted from UMLS.

Depth	*Disease*	*CUI*
1	Disease	C0012634
	…	
2	. cancer	C0006826
	…	
3	. . respiratory system cancer	C0814136
4	. . . bronchus cancer	C0345950
4	. . . lung cancer	C0242379
5	. . . . small lung cell cancer	C0149925
5	. . . . non small lung cell cancer	C0220601
4	. . . mediastinum cancer	C0153504
4	. . . pleural tumor	C0345966
	…	

Table 14.5 Sample disease knowledge hierarchy extracted from UMLS for the UCLA document set

There are 875,255 concepts in UMLS, 2003AA edition. In a real dataset, the number of concepts appearing in the topics of a document set can be much less. Table 14.6 shows the number of concepts for some knowledge types in the set of about 5000 UCLA medical reports.

TUI	*Knowledge Type*	*Number of Concepts*
T191	Disease	181
T184	Finding	171
T061	Treatment	242
T060	Diagnosis	155
T023	Body Organ	482

Table 14.6 Number of concepts for some knowledge types for the UCLA document set

We obtain knowledge hierarchies for the five types of knowledge as shown in Table 14.6. A portion of the knowledge paths used in our experiment are shown in Table 14.7.

CUI	*TUI*	*Concept Name*	*Knowledge Path*
C0000735	T191	abdomen tumor	disease/cancer/abdomen
C0001418	T191	adenocarcinoma	disease/cancer/epithelial/adenocarcinoma
C0001624	T191	adrenal tumor	disease/cancer/urological/kidney/adrenal
C0005967	T191	Bone cancer	disease/cancer/bone
C0006118	T191	Brain tumor	disease/cancer/neurologic/brain
C0006142	T191	breast cancer	disease/cancer/breast
C0006264	T191	bronchus tumor	disease/cancer/respiratory/bronchus
…	…	…	…

Table 14.7 Example of the simplified directory paths for some disease concepts

We constructed the topic directory systems by using a set of 50 patient reports from the UCLA Medical Center. Figure 14.8 shows a user giving a usage pattern [disease], after which the system creates a directory for the usage pattern.

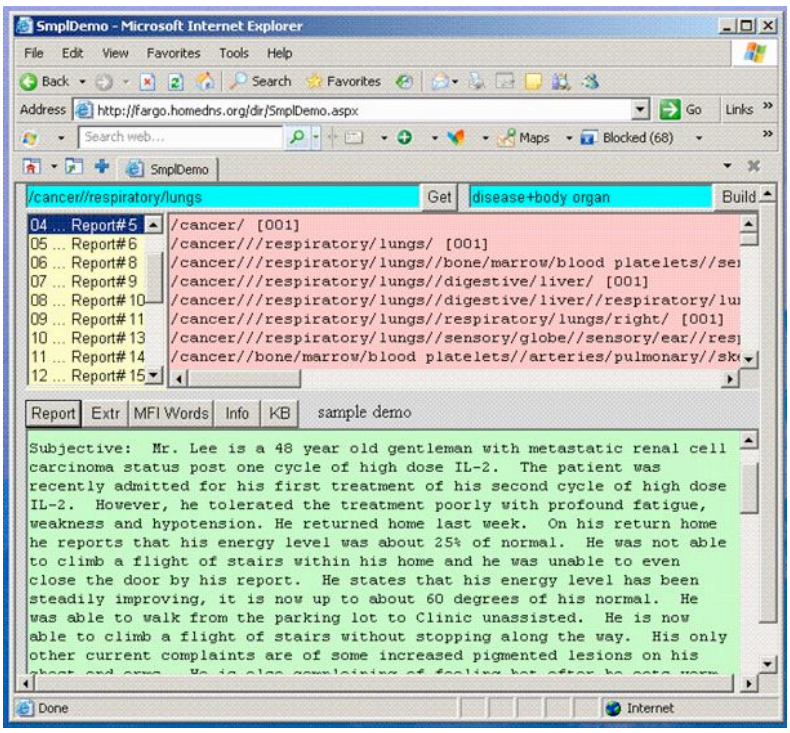


Figure 14.8: Topic directory system experiment

Using such a directory, a user is able to obtain patient reports that are organized by disease type + body organ. For example, if a user wants to find reports on cancer/respiratory system/lung, the system returns 33 reports as illustrated in the upper-left corner of Figure 14.8. When a user clicks on a report, the system will bring the document to the user as shown in the bottom of the figure.

Such a system provides the user with the capability to generate a topic oriented directory, which enables the user to navigate the information that best satisfies the query goals. Such scenario-speific directories generate a set of relevant clinical free text documents which can then be inputted for ranking.

14.5 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 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) [22] 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.

14.5.1 The Problem

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

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

1. The component words of a phrase sometimes have only 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 document indexing. On the other hand, through manual indexing, [23] showed the potential of concept-based indexing to produce significant improvements over the stem-based scheme. 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.

14.5.2 Vector Space Models

Stem-based Vector Space Model

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," and "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.

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

Concept-based Vector Space Model

Using word stems to represent documents 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 mimics human thought processes, 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." 3) Some concepts are related to one another.

Assuming that we can partition the documents into phrases, and ignoring the polysemy, our example query becomes (C0015967), (C0023518), and (C0151740) etc., representing "hyperthermia," "leukocytosis," and "increased intracranial pressure," etc., respectively, where the three strings in the parentheses are concept unique identifiers (CUIs) in UMLS [24].

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, $w_{c,x} = \tau_{c,x}\iota_c = \tau_{c,x}(\log_2(N/n_c)+1)$, 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

$$\langle x,y\rangle^c = \sum_{c\in C}\sum_{d\in C} \iota_c \tau_{c,x} \iota_d \tau_{d,y} s^c(c,d) \qquad (14\text{-}1)$$

where we take $s^c(c,d)$, the conceptual similarity between concepts c and d, into consideration. The similarity between documents x and y is defined to be the cosine of the angle between their respective document vectors, $sim^c(x,y) = \frac{\langle x,y\rangle^c}{\sqrt{\langle x,x\rangle^c \langle y,y\rangle^c}}$.

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. Noticing the words "infiltrative" and "process" that match no concepts and the common component word "cerebral" in phrases "cerebral edema" and "cerebral lesion," we propose a phrase-based VSM to remedy the incompleteness of the controlled vocabularies and the knowledge sources.

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.

We use an ordered pair of two sets to represent a *phrase* p = ({(s, $\pi_{s,p}$)}s∈S, {(c, $\pi_{c,p}$)}c∈C). The first set, {(s, $\pi_{s,p}$)}s∈S, consists of ordered pairs that indicate the stems and their occurrence counts, $\pi_{s,p}$, in the phrase. The second set {(c, $\pi_{c,p}$)}c∈C indicates the concepts and their occurrence counts, $\pi_{c,p}$, in the phrase. 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∈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 ι_s, ι_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 *cp* is the concept that phrase *p* expresses. Then the phrase similarity is reduced to

$$s^p(p, q) = \max\left(\left(f^s \sum_{s \in S} \iota_s^2 \pi_{s,p} \pi_{s,q}\right), \left(f^c \iota_{c_p} \iota_{d_q} s^c(c_p, d_q)\right)\right) \qquad (14\text{-}2)$$

where *cp* is the concept phrase *p* expresses, and *dq* 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*. We can achieve this goal by applying indexing methods such as IndexFinder or the Aho-Corasick algorithm.

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.

Conceptual Similarity Evaluation

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 [25]. 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

$$s^c(c,d) = \frac{1}{l(c,d)\log_2(D(c) + D(d) + 1)}, \qquad (14\text{-}3)$$

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.

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 [26]. 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.

14.5.3 Experimental Evaluation of The Phrase-Based VSM

Phrase Detection and Conceptual Similarity Derivation via UMLS

In our experiments, we used UMLS as the controlled vocabulary for phrase detection. We also apply the conceptual relations in the Metathesaurus to derive conceptual similarities. 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 [25]. 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 Eq. (14.3) to compute the conceptual similarities.

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 [16] 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 have 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 [27] 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 are provided by "a medical school student."

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

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 OHSUMED collection in later sections. Table 14.8 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).

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 [28-29] based on the user's judgment of whether each document is relevant to a query q. When a certain number of documents are returned, we define *precision* to be the proportion of the retrieved documents that are relevant; and define *recall* 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*.

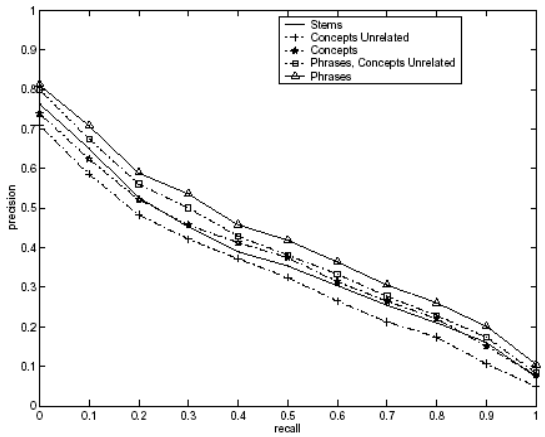


Fig.14.9 Comparison of the average recall-precision curves over 105 OHSUMED queries

Comparison of the Recall-Precision Curves

Figures 14.9 and 14.10 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. For the OHSUMED results,

1. "Stems" is the baseline generated by the stem-based VSM. Its average 11-point average precision is $G^{s}P11 = 0.376$.
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 (Eq. 14-1). The average 11-point average precision is $G^{cu}P11 = 0.336$, an 11% decrease from the baseline.

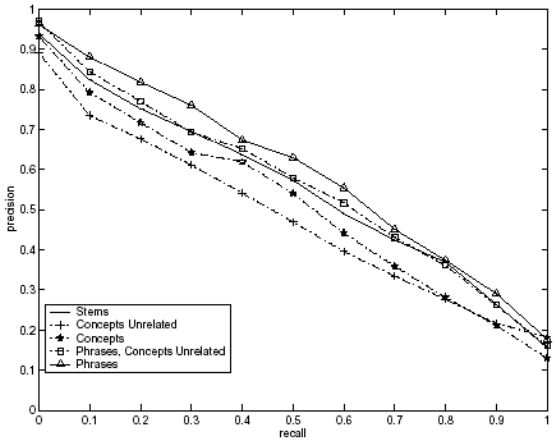


Fig.14.10 Comparison of the average recall-precision curves over 30 Medlars queries.

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 Eq. (14.2) 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.

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 14.9 and Figure 14.10, 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 Eq. (14.4), (14.5) and (14.6), 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 14.11 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.

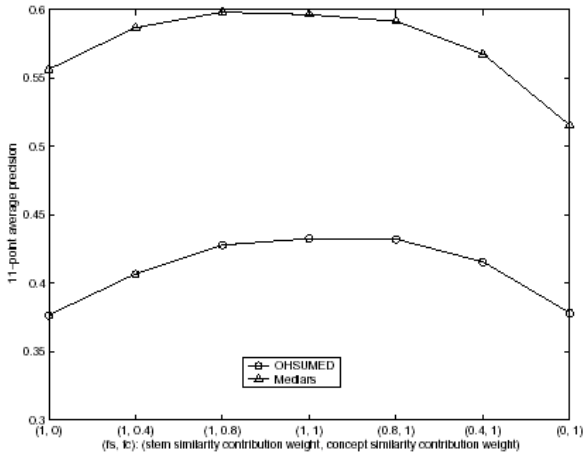


Fig.14.11 Sensitivity of *GP*11 to f^s, f^c changes in OHSUMED and Medlars.

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 [30-31]. The resulting document clusters are searched using top-down and bottom-up searching strategies.

Figure 14.12. Retrieval effectiveness comparison of the cluster-based retrieval in OHSUMED.

Figure 14.12 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 [23]. In our case, there is a 20% improvement average over 100 queries, representing a significant improvement.

14.5.4 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.

14.6 KNOWLEDGE-BASED SCENARIO-SPECIFIC QUERY EXPANSION

14.6.1 A Framework for Knowledge-Based Query Expansion

A knowledge-based query expansion and retrieval framework is shown in Figure 14.13. For a given query, *Statistical Query Expansion* (whose scope is marked by the inner dotted rectangle) derives *candidate expansion concepts*[1] that are statistically co-occurring with the given query concepts (Section 14.6.2) and assign weights to each candidate concept according to the statistical co-occurrence. Such weights will be carried through the framework. Based on the candidate concepts derived by statistical expansion, *Knowledge-based Query Expansion* (whose scope is marked by the outer rectangle) further derives the scenario-specific expansion concepts, with the aid of a domain knowledge source such as UMLS [32] (Section 14.6.2). Such knowledge may be incomplete and fail to include all possible query scenarios. Therefore, in an off-line process, we apply a *Knowledge Acquisition and Supplementation* module to supplement the incomplete knowledge (Section 14.6.5. After the query is expanded with scenario-specific concepts, we employ a *Vector Space Model* (VSM) to compare the similarity between the expanded query with each document. Top-ranked documents with the highest similarity measures are output to the user.

[1] In the rest of this paper, a concept is referred to as a word or a word phrase that has a concrete meaning in a particular application domain. In the medical domain, concepts in free text can be extracted using existing tools, e.g. MetaMap [Aro01], IndexFinder [ZCM03], etc.

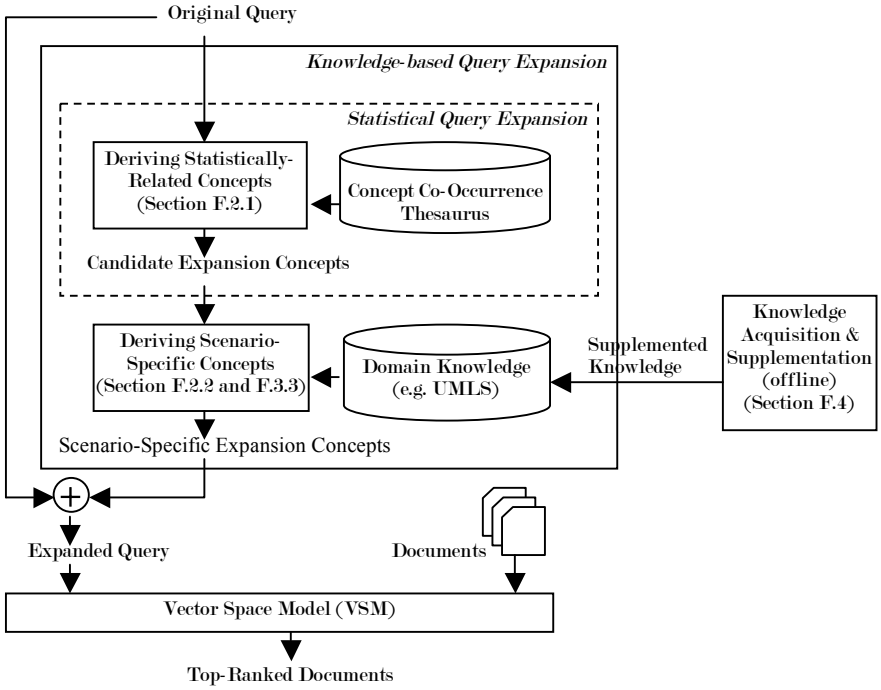


Figure 14.13: A knowledge-based query expansion and retrieval framework

14.6.2 Method

Formally, the problem for knowledge-based query expansion can be stated as follows: Given a scenario-specific query with a key concept denoted as c_{key} (e.g., `lung cancer` or `keratoconus`[2]) and a set of scenario concepts denoted as c_s (e.g., `treatment` or `diagnosis`), we need to derive specialized concepts that are related to c_{key} and the relations should be specific to the scenarios defined by c_s. In this section, we describe how to derive such scenario-specific concepts by presenting existing statistical query expansion methods which generate candidate concepts. We then propose a method that selects scenario-specific concepts from this candidate set with the aid of a domain knowledge source.

Deriving Statistically-Related Expansion Concepts

Statistical expansion is also referred to as *automatic query expansion* [33-34]. The basic idea is to derive concepts that are statistically related to the given query concepts, where the statistical correlation is derived from a document collection (e.g., OHSUMED [16]). Appending such concepts to the original query makes the query expression more specialized and thus match relevant documents better. Depending on how such statistically-related concepts are derived, statistical expansion methods fall into two major categories:

[2] An eye disease

- *Co-occurrence-thesaurus-based expansion* [35-37]. In this method, a *concept cooccurrence thesaurus* is first constructed automatically offline. Given a vocabulary of M concepts, the thesaurus is an $M \times M$ matrix, where the $<i, j>$ element quantifies the co-occurrence between concept i and concept j. When a query is posed, we look up the thesaurus to find all concepts that statistically co-occur with concepts in the given query and assign weights to those co-occurring concepts according to the values in the co-occurrence matrix. A detailed procedure for computing the co-occurrence matrix and for assigning weights to expansion concepts can be found in [35].
- *Pseudo-relevance-feedback-based expansion* [34, 38-41]. In pseudo relevance feedback, the original query is used to perform an initial retrieval. Concepts extracted from top-ranked documents in the initial retrieval are considered statistically related and are appended to the original query. This approach resembles the well-known *relevance feedback* approach except that, instead of asking users to identify relevant documents as feedback, top-ranked (e.g. top-10) documents are automatically treated as "pseudo" relevant documents and are inserted into the feedback loop. Weight assignment in pseudo relevance feedback [39] typically follows the same weighting scheme for conventional relevance feedback techniques [38].

We note that the choice of statistical expansion method is orthogonal to the design of the knowledge-based expansion framework (Figure 14.13). In our current experimental evaluation, we used the co-occurrence-thesaurus-based method to derive statistically related concepts. For convenience of discussion, we use $co(c_i, c_j)$ to denote the co-occurrence between concept c_i and c_j, a value that appears as the $<i, j>$ element in the $M \times M$ co-occurrence matrix. Table 14.9 lists the top-15 concepts that are statistically related to `keratoconus` using the co-occurrence measure. Here, the co-occurrence measure is computed from the OHSUMED corpus.

#	Concepts that statistically correlate to `keratoconus`
1	fuchs dystrophy
2	penetrating keratoplasty
3	Epikeratoplasty
4	corneal ectasia
5	acute hydrops
6	Keratometry
7	corneal topography
8	Corneal
9	aphakic corneal edema
10	Epikeratophakia
11	granular dystrophy corneal
12	Keratoplasty
13	central cornea
14	contact lens
15	ghost vessels

Table 14.9: Concepts that statistically correlate to `keratoconus`

Deriving Scenario-Specific Expansion Concepts

Using a statistical expansion method, we can derive a set of concepts that are statistically-related to the key concept, c_{key}, of the given query. Only a subset of these concepts are relevant to the given query's scenario, e.g., `treatment`. For example, the 5th and 8th concepts in

Table 14.9, which are `acute hydrops` and `corneal`, are not related to the treatment of `keratoconus`. Therefore, in terms of deriving expansion concepts for query `keratoconus treatment`, these concepts should be filtered out. In this section, we will first describe the type of knowledge structure that enables us to perform this filtering and then present the filtering procedure.

In previous sections, we have introduced UMLS and how to apply its subsystems, i.e,, the Metathesaurus and the SPECIALIST lexicon, for implementing the IndexFinder and the Phrase-base VSM. For the task of knowledge-based query expansion, we apply the subsystem of the Semantic Network.

The Semantic Network defines about one hundred *semantic types* such as `Disease or Syndrome`, `Body Part`, etc. Each semantic type corresponds to a class/category of concepts. The semantic type `Disease or Syndrome`, for instance, corresponds to 44,000 concepts in the Metathesaurus such as `keratoconus`, `lung cancer`, `diabetes`, etc. Besides the list of semantic types, the Semantic Network also defines the relations among various semantic types, such as `treats` and `diagnoses`. Such relations link isolated semantic types into a graph/network structure. The top half of Figure 14.14 presents a fragment of this network, which includes all semantic types that have a `treats` relation with the semantic type `Disease or Syndrome`. Relations such as `treats` in Figure 14.14 should be interpreted as follows: Any concepts that belong to semantic type `Therapeutic or Preventive Procedure`, e.g., `penetrating keratoplasty` or `chemotherapy`, have the potential to `treat` concepts that belong to the semantic type `Disease or Syndrome`, e.g., `keratoconus` or `lung cancer`. However, it is not indicated whether such relations concretely exist between two concepts, e.g., a `treats` relation between `penetrating keratoplasty` and `lung cancer`.

Given the knowledge structure in the Semantic Network, the basic idea in identifying scenario-specific expansion concept is to use this knowledge structure to filter out statistically-correlated concepts which do not belong to the specific semantic types. Let us illustrate this idea through Figure 14.14, using the `treatment` scenario as an example: We start with the set of concepts that are statistically related to `keratoconus`. Our goal in applying the knowledge structure is to identify that: 1) concepts such as `penetrating keratoplasty`, `contact lens` and `griffonia` have the scenario-specific relation, i.e., `treats`, with `keratoconus` and should be kept during expansion; 2) concepts such as `acute hydrops` and `corneal` which do not have the scenario-specific relation with `keratoconus` are filtered out.

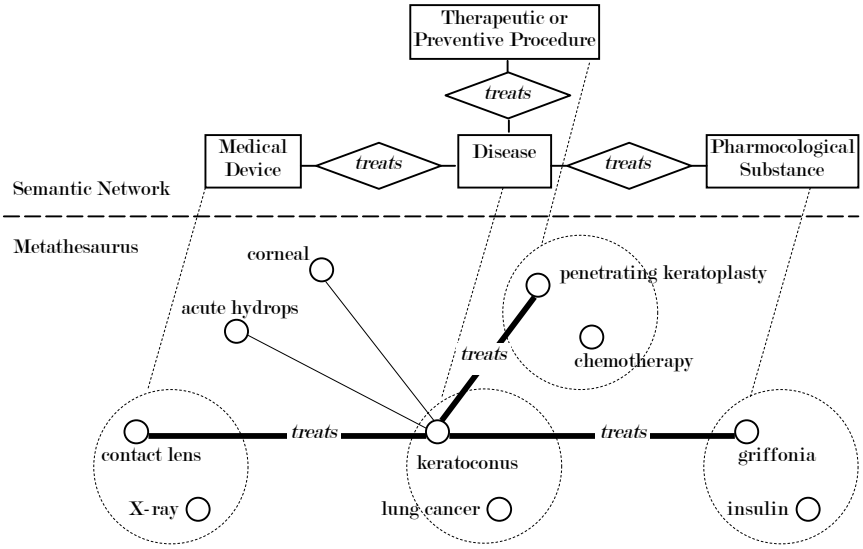


Figure 14.14: Using knowledge to identify scenario-specific concept relationships

Each solid circle in Figure 14.14 represents a single concept, and the solid lines connecting these solid circles indicate strong statistical correlations computed for a pair of concepts, e.g., the solid line between `keratoconus` and `contact lens`. A dotted circle represents a class of concepts, and a dotted line links that class of concepts to a corresponding semantic type. For example, concepts `keratoconus` and `lung cancer` are in the class that links to `Disease or Syndrome`. We identified scenario-specific expansion concepts using the following process: Given a key concept c_{key} of the given query, we first identified the semantic type that c_{key} belongs to. For example, we identified `Disease or Syndrome` given the key concept `keratoconus`. Starting from that semantic type, we further followed the relations marked by the query's scenario and reached a set of relevant semantic types. For the previous example, given the query's scenario, `treatment`, we followed the `treats` relation to reach the three other semantic types as shown in Figure 14.14. Finally, we identified those statistically-related concepts that belong to the relevant semantic types as scenario specific. We further filtered out other statistically-related concepts which do not satisfy this criteria. From the previous example, this final step identified `penetrating keratoplasty`, `contact lens` and `griffonia` as scenario-specific expansion concepts and filtered out non-scenario-specific ones such as `acute hydrops` and `corneal`.

#	Concepts that *treat* keratoconus
1	penetrating keratoplasty
2	epikeratoplasty
3	epikeratophakia
4	keratoplasty
5	contact lens
6	thermokeratoplasty
7	button
8	secondary lens implant
9	fittings adapters
10	esthesiometer
11	Griffonia
12	Trephine
13	slit lamps
14	fistulization
15	soft contact lens

(a)

Concepts that *diagnose* keratoconus
keratometry
corneal topography
slit lamp examination
topical corticosteroid
echocardiography 2 d
Tem
Interferon
Alferon
Analysis
Microscopy
Bleb
tetanus toxoid
Antineoplastic
heart auscultation
Chlorbutin

(b)

Table 14.10: Concepts that `treat` or `diagnose keratoconus`

The lists of the concepts for `treating` and `diagnosing keratoconus` are shown in Table 14.10(a) and Table 14.10(b). These concepts were derived based on the process described above and show the top-15 concepts in terms of their correlation with `keratoconus`. To highlight the effectiveness of the knowledge-based filtering process, we can compare the concepts in Table 14.10 with those in Table 14.9 that are statistically correlated with `keratoconus`. 5 out of these 15 statistically-correlated concepts are kept in Table 14.10(a), whereas 2 are kept in Table 14.10(b). This comparison reveals that the knowledge structure is effective in filtering out concepts that are not closely related to the scenarios of `treatment` or `diagnosis`.

The goal of knowledge-based query expansion is to append specialized terms that appear in relevant documents but not in the original query. Scenario-specific concepts derived from the previous subsection represent a subset of such specialized terms. Another set of highly relevant terms contains hypernym/hyponyms of the key concept c_{key}.[3] For example, `corneal estasia`, a hypernym of `keratoconus`, is frequently mentioned by documents regarding `keratoconus treatment`. Therefore, we need also expand those concepts that are close to c_{key} in the hypernym/hyponym hierarchy.

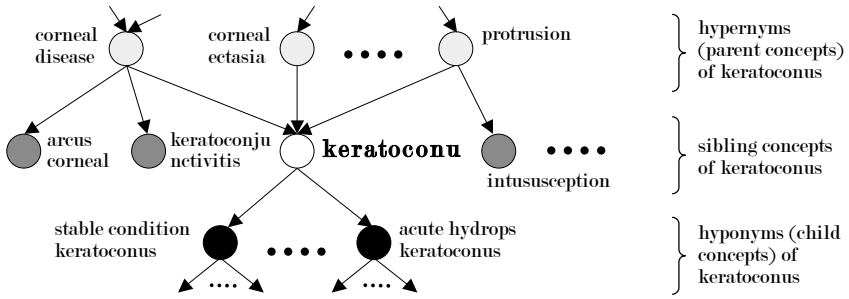


Figure 14.15: The direct parents, direct children and siblings for `keratoconus`

[3] A hypernym of concept *c* is a concept with a broader meaning than *c*, whereas a hyponym is one with a narrower meaning.

To expand hypernyms/hyponyms of the key concept to the original query, we again refer to the UMLS knowledge source. The Metathesaurus subsystem defines not only the concepts but also the hypernym/hyponym relationships among these concepts. For example, Figure 14.15 shows the hypernyms (parents), hyponyms (children) and siblings of concept `keratoconus`. Here we define a concept's siblings as those concepts that share the same parents with the given concept. Through empirical study (which will be discussed later), we have found that expanding the direct parents, direct children and siblings to the original query generates the best retrieval performance. This is in comparison to expanding parents/children that are two or more levels away from the key concept. Therefore, in the rest of our discussion, we will focus on expanding only the direct parents/children and siblings.

Concepts that statistically correlate to `keratoconus`	Weight
fuchs dystrophy	0.289
penetrating keratoplasty	0.247
Epikeratoplasty	0.230
corneal ectasia	0.168
acute hydrops	0.165
Keratometry	0.133
corneal topography	0.132
Corneal	0.130
aphakic corneal edema	0.122
Epikeratophakia	0.119
granular dystrophy corneal	0.109
Keratoplasty	0.103
central cornea	0.103
contact lens	0.101
ghost vessels	0.095

(a)

Concepts that treat `keratoconus`	Weight
penetrating keratoplasty	0.247
Epikeratoplasty	0.230
Epikeratophakia	0.119
Keratoplasty	0.103
contact lens	0.101
Thermokeratoplasty	0.092
Button	0.067
secondary lens implant	0.057
fittings adapters	0.048
Esthesiometer	0.043
Griffonia	0.035
Trephine	0.033
slit lamps	0.032
Fistulization	0.030
soft contact lens	0.026

(b)

Table 14.11: Weights for sample expansion concepts

Weight Adjustment for Expansion Terms

To match a query and a document using the Vector Space Model (VSM), we represent both the query and the document as vectors. Each term in the query becomes a dimension in the query vector, and receives a weight that quantifies the importance of this term in the entire query. Under this model, any additional term appended to the original query needs to be assigned a weight. An appropriate weight scheme for these additional terms is important because "under-weighting" will make the additional terms insignificant compared to the original query and lead to unnoticeable changes in the ranking of the retrieval results. On the other hand, "over-weighting" will make the additional terms overly significant and cause a "topic drift" for the original query.

In the past, researchers have proposed weighting schemes for these additional terms based on the following intuition: The weight for an additional term c_a should be proportional to its correlation with the original query terms. Thus, the weight for c_a, w_a, is proportional to its correlation with the key concept c_{key}, i.e.:

$$w_a = co(c_a, c_{key}) \cdot w_{key} \qquad (14\text{-}4)$$

In Eq.(14-4), the correlation between c_a and c_{key}, $co(c_a, c_{key})$, is derived using methods described in Section 14.6.2. w_{key} denotes the weight assigned to the key concept c_{key}. In Section 14.6.3 we will further explain how w_{key} is decided according to a common weighting scheme. Given that $co(c_a, c_{key})$ lies in [0, 1], the weight that c_a receives will not exceed that of c_{key}. Using this equation, we compute the weights for the terms that statistically correlate with keratoconus (Table 14.9) and the weights for those that treat keratoconus (Table 14.10(a)). We list the weights for these terms in Table 14.11(a) and Table 14.11(b), respectively. These weights are computed by assuming the weight of the key concept (i.e., w_{key}) keratoconus is 1.

We will compare the retrieval effectiveness of knowledge-based query expansion with that of statistical expansion. Since the knowledge-based method applies a filtering step to derive a subset of all statistically-related terms, the impact created by this subset on retrieval effectiveness will be less than the entire set of statistically-related terms. Therefore, weight adjustments are needed to compensate for the filtering. For instance, in our example of keratoconus, treatment, the "cumulative weight" for all terms in Table 14.11(b) is obviously smaller than the "cumulative weight" of those in Table 14.11(a). To increase the impact of the terms derived by the knowledge-based method, we can "boost" their weights by multiplying a linear factor β, so that the cumulative weight of those terms is comparable to those of the statistical-related terms. We refer to β as the *boosting factor*. With this factor, we alter Eq.(14-4) which assigns the weight for any additional term c_a as follows:

$$w_a = \beta \cdot c_o(c_a, c_{key}) \cdot w_{key} \tag{14-5}$$

We quantify the cumulative weight for both the statistical expansion terms (e.g., those in Table 14.11(a)) and the knowledge-based expansion terms (e.g., those in Table 14.11(b)). The former cumulative weight will be larger than the latter. We define β to be the former divided by the latter. In this way, the cumulative weight for the knowledge-based expansion terms equals to that of the statistical expansion terms after boosting.

More specifically, we quantify the cumulative weight of a set of expansion terms using the length of the "expansion vector" composed by these terms. Here we define the vector length according to the standard vector space notation: Let $V^{KB} = <w_1^{KB}, ..., w_k^{KB}>$ be the augmenting vector consisting solely of terms derived by the knowledge-based method, where w_i^{KB} $(1 \le i \le k)$ denotes the weight for the i_{th} term in knowledge-based expansion (Eq.(14-4)). Likewise, let $V^{stat} = <w_1^{stat}, ..., w_l^{stat}>$ be the augmenting vector consisting of all statistically related terms. The process of deriving $\{w_1^{KB}, ..., w_k^{KB}\}$ yields $k < l$. Consequently, $\{w_1^{KB}, ..., w_k^{KB}\} \subset \{w_1^{stat}, ..., w_l^{stat}\}$. Let $|V^{KB}|$ be the length of the vector V^{KB}, i.e.,

$$|V^{KB}| = \sqrt{\left(w_1^{KB}\right)^2 + \left(w_1^{KB}\right)^2 + \cdots + \left(w_k^{KB}\right)^2} \tag{14.6}$$

Likewise, let $|V^{stat}|$ represent the length of vector V^{stat} which can be computed similarly as Eq.(14-6). Thus, the *boosting factor* for V^{KB} is:

$$\beta = \frac{|V^{stat}|}{|V^{KB}|} \tag{14-7}$$

To study the effects of different levels of boosting, a *boosting-level-controlling factor* α is introduced to refine Eq.(14-7):

$$\beta_r = 1 + \alpha \cdot \left(\frac{|V^{stat}|}{|V^{KB}|} - 1 \right) \quad (14\text{-}8)$$

where β_r is the refined boosting factor. The parameter α, ranging within [0, 1], can be used to control the boosting scale. From Eq.(14-8), we note that $\beta_r = 1$ when we set $\alpha = 0$, which represents no boosting. β_r increases as α increases. As α increases to 1, β_r reduces to $\frac{|V^{stat}|}{|V^{KB}|}$. Thus, α can be used to experimentally study the boosting sensitivity. (We have experimentally evaluated cases of setting $\alpha > 1$. We noted that the retrieval effectiveness in those cases is usually sub-optimal compared to cases with α within [0, 1].)

14.6.3 Retrieval Performance

In this section, we compare the retrieval performance of the knowledge-based query expansion with that of statistical expansion using two standard medical corpuses. We start with the experiment setup and then present the results under selective settings.

Testbeds

A testbed for a retrieval experiment consists of three components: 1) a corpus (or a document collection), 2) a set of benchmark queries and 3) relevance judgments indicating which documents are relevant for each query. Our experiment is based on the following two testbeds:

OHSUMED [HBL94]. This testbed has been introduced in Section 14.5.3. In the task of evaluating knowledge-based query expansion, we are interested in a subset of the OHSUMED queries which are scenario-specific. Among the 106 queries, we have identified a total number of 57 such queries. In Table 14.12, we categorize these 57 queries based on the scenario(s) each query mentions. The corresponding ID of each query is listed in this table. (The full text of each query is shown in [42]). Note that a query mentioning multiple distinct scenarios will appear multiple times in this table corresponding to its scenarios.

Scenario	Query ID
treatment of a disease	2, 13, 15, 16, 27, 29, 30, 31, 32, 35, 37, 38, 39, 40, 42, 43, 45, 53, 56, 57, 58, 62, 67, 69, 72, 74, 75, 76, 77, 79, 81, 85, 93, 98, 102
diagnosis of a disease	15, 21, 37, 53, 57, 58, 72, 80, 81, 82, 97
prevention of a disease	64, 85
differential diagnosis of a symptom/disease	14, 23, 41, 43, 47, 51, 65, 69, 70, 74, 76, 103
pathophysiology of a disease	2, 3, 26, 64, 77
complications of a disease/medication	3, 30, 52, 61, 62, 66, 79
etiology of a disease	14, 26, 29
risk factors of a disease	35, 64, 85
prognosis of a disease	45
Epidemiology of a disease	3
research of a disease	75
organisms of a disease	81
criteria of medication	49, 52, 94
when to perform a medication	33
preventive health care for a type of patients	96

Table 14.12: OHSUMED queries categorized based on their scenarios

The McMaster Clinical HEDGES Database [43-46]. This testbed was originally constructed for the task of medical document classification instead of free-text query

answering. As a result, adaptation is needed for retrieval performance study. We first describe the original dataset, and then explain how we adapted it to make it a usable testbed for retrieval performance evaluation.

- Original dataset. The McMaster Clinical HEDGES Database contains 48,000 PubMed articles published in 2000. Each article was classified into the following scenario categories: *treatment*, *diagnosis*, *etiology*, *prognosis*, *clinical prediction guide* of a disease, *economics* of a healthcare issue, or *review* of a healthcare topic. Consensus about the classification was drawn among six human experts [WMH01]. When the experts classified each article, they had access to the hardcopies of the full text. However, to construct a testbed for our retrieval system, we were only able to download the title and abstract of each article from the PubMed system. (The full text of each article is typically unavailable through PubMed.)
- Construction of Scenario-Specific Queries. Since the McMaster Clinical HEDGES Database is constructed to test document classification, it does not contain a query set. Using the following procedure, we constructed a set of 55 scenario-specific queries, and determined the relevance judgements for these queries based on the document classification that can be adapted for these queries:

  Step 1. We identified all the disease/symptom concepts in the OHSUMED query set. We identified such concepts based on their semantic type information (defined by UMLS). We used these concepts as the key concepts in constructing the scenario-specific queries for the McMaster testbed. In selecting these concepts, we manually filtered out eight concepts (out of an original number of 90 concepts) that we considered as too general to make a scenario-specific query, e.g., infection, lesion and carcinoma. After this step, we obtained 82 such key concepts.

  Step 2. For each key concept identified in Step 1, four scenario-specific queries are constructed, namely the treatment, diagnosis, etiology and prognosis of a disease/symptom. For example, for the concept breast cancer, we constructed the queries breast cancer treatment, breast cancer diagnosis, breast cancer etiology, and breast cancer prognosis. Our study was restricted to these four scenarios because UMLS only covers these four scenarios.

  Step 3. For each query generated in Step 2, we generated its relevance judgments by applying the following simple criterion: A document is considered to be relevant to a given query if 1) experts have classified the document to the category of the query's scenario and 2) the document mentions the query's key concept. This criterion has been our best choice to automate the process of generating relevance judgments on a relatively large scale; however, it may misidentify irrelevant documents as relevant. After we identified the relevant documents for each query, certain queries are filtered out based on the intuition that a query with too few relevance judgments will lead to less reliable retrieval results (especially in terms of precision/recall). For example, for a query with only one relevant document, two similar retrieval systems may obtain completely different precision/recall results if one ranks the relevant document on top, and another accidentally ranks it out of top-10. Following this intuition, queries that have less than 5 relevant documents are filtered out. After this filtering step, we were left with 55 queries. These queries together with the scenarios identified for each query are presented in [42].

VSM and Indexing

In Information Retrieval studies, *indexing* refers to the step of converting free-text documents and queries to their respective vector representations [29]. The query and document vectors are then matched based on a Vector Space Model (VSM). In experimental evaluation of the knowledge-based query expansion method, we focus on results generated by the following two VSMs:

- Stem-based VSM [29]. Using a stem-based VSM, both a query and a document are represented as vectors of word stems. Given a piece of free text, we first removed common stop words such as "a," "the," etc., and then derived word stems from the text using the Lovins stemmer [47]. We further applied the $tf \cdot idf$ weighting scheme (more specifically the $atc \cdot atc$ scheme [48]) to assign weights to stems in documents and the query before expansion. (This weighting process yields the weight for the key concept in Eq.(14.1). Under the stem-based VSM, all terms expanded to a given query need to be in the word-stem format. Thus, for expansion concepts derived from procedures in Section 14.6.2, we applied the following procedure to identify the corresponding word stems: For each expansion concept, we first looked up its string forms in UMLS. We further removed stop words and used the Lovins stemmer to convert the string forms into word stems. Lastly, we assigned weights to these expansion word stems using the method described in Section 14.6.2.
- Phrase-based VSM [5]. Using a phrase-based VSM, both a query and a document are represented as vectors of phrases. We first used the concept extraction method presented in Section 14.2 to identify the concepts appearing in a given query and a set of documents. We further formulated phrase representations of the query and the documents based on the definition of phrases in Section 14.5.2. We applied the weighting method in Section 14.5.2 to assign weights to phrases in the query and the documents. For expansion concepts appended to the original query, we converted them into their corresponding phrase representation, and assigned the weights for both concepts and word stems appearing in a phrase using the method described in Section 14.6.2.

Evaluation Metrics

We measure the retrieval performance using the following three different metrics:

- *avgp* - 11-point precision average (precision averaged over the 11 standard recall points [29])
- *p@10* - precision in top-10 retrieved documents
- *p@20* - precision in top-20 retrieved documents

Retrieval Performance Using The Stem-Based VSM

In the following, we study the performance improvement of knowledge-based expansion as compared to that of statistical expansion.

We use s to denote an expansion size. For a given s, we used both knowledge-based expansion and statistical expansion to expand the top-s stems that have the heaviest weights. For knowledge-based expansion, no weight boosting was applied. We compute the three metrics for both methods on the OHSUMED and McMaster testbeds. We further

average the results over the queries in these two testbeds. Table 14.13 shows the performance comparison of the two methods on both testbeds, which is under the above metrics. The first row in each sub-table shows the performance of statistical expansion, whereas the second row shows the performance of knowledge-based expansion and its percentage of improvement over statistical expansion.

In these tables, "*s*=All" means appending all possible expansion terms that have a non-zero weight (Eq.(14.5)) into the original query. Using the knowledge-based method, setting "*s*=All" led to expanding an average of 1717 terms to each query on average, with the standard deviation of 1755; using the statistical method, it led to an average of 50317 terms with the standard deviation of 15243.

From these experimental results, we observe the following: The performance for knowledge-based expansion generally increases as *s* increases and usually reaches the peak when *s*=All. (The only exception is in the case of using the *avgp* metric on the McMaster testbed, in which the performance of the knowledge-based method roughly remains constant as *s* increases.) On the other hand, the performance of the statistical method degrades as *s* increases. On the OHSUMED testbed, its performance degrades after *s* reaches a certain level, e.g., *s*=100 (Table 14.13(a)) and *s*=200 (Table 14.13(b) and Table 14.13(c)); on the McMaster testbed, the performance starts degrading almost immediately after *s* exceeds 20. This is due to the fact that statistical expansion does not distinguish whether an expansion term is scenario-specific. As a result, as more terms are appended to the original query, the negative effect of including those nonscenario-specific terms begins to accumulate and the performance drops after a certain point. In contrast, the knowledge-based method appends scenario-specific terms, and consequently, the performance keeps increasing as more "useful" terms are appended.

Our experimental results also revealed that both statistical expansion and knowledge-based expansion consistently outperform the no expansion method by more than 5%. On the OHSUMED testbed, for example, the *avgp* of no expansion is 0.382, which is outperformed by the peak performance of statistical expansion at 0.432 and by the peak performance of knowledge-based expansion at 0.452 (Table 14.13(a)). Similarly, the *p@10* and *p@20* of no expansion are 0.532 and 0.470, which are outperformed by the peak performance of statistical expansion at 0.581 and 0.497, and by the peak performance of knowledge-based expansion at 0.600 and 0.514 (Table 14.13(b) and 14.13(c)).

S	10	20	30	40	50	100	200	300	All
Statistical Expansion	0.417	0.424	0.428	0.43	0.429	0.432	0.429	0.43	0.425
Knowledge-Based Expansion without weight boosting	0.422	0.431	0.430	0.432	0.434	0.438	0.442	0.443	0.445
Knowledge-Based Expansion with weight boosting	0.428	0.436	0.437	0.437	0.439	0.443	0.446	0.450	0.452

(a) Performance comparison using the *avgp* metric for the OHSUMED testbed

S	10	20	30	40	50	100	200	300	All
Statistical Expansion	0.535	0.546	0.549	0.553	0.551	0.567	0.581	0.574	0.567
Knowledge-Based Expansion without weight boosting	0.544	0.547	0.554	0.551	0.553	0.572	0.572	0.577	0.588
Knowledge-Based Expansion with weight boosting	0.552	0.567	0.568	0.577	0.577	0.595	0.586	0.595	0.600

(b) Performance comparison using the *p@10* metric for the OHSUMED testbed

S	10	20	30	40	50	100	200	300	All
Statistical Expansion	0.482	0.491	0.493	0.491	0.492	0.496	0.497	0.493	0.496
Knowledge-Based Expansion without weight boosting	0.483	0.491	0.494	0.496	0.493	0.498	0.496	0.497	0.498
Knowledge-Based Expansion with weight boosting	0.482	0.496	0.498	0.510	0.509	0.514	0.514	0.513	0.511

(c) Performance comparison using the *p@20* metric for the OHSUMED testbed

S	10	20	30	40	50	100	200	300	All
Statistical Expansion	0.326	0.328	0.325	0.324	0.323	0.319	0.311	0.309	0.295
Knowledge-Based Expansion without weight boosting	0.325	0.328	0.324	0.326	0.325	0.324	0.321	0.32	0.321
Knowledge-Based Expansion with weight boosting	0.325	0.326	0.324	0.325	0.323	0.322	0.320	0.315	0.318

(d) Performance comparison using the *avgp* metric for the McMaster testbed

S	10	20	30	40	50	100	200	300	All
Statistical Expansion	0.316	0.324	0.324	0.318	0.324	0.311	0.295	0.3	0.293
Knowledge-Based Expansion without weight boosting	0.322	0.324	0.322	0.325	0.322	0.318	0.315	0.32	0.335
Knowledge-Based Expansion with weight boosting	0.320	0.322	0.318	0.322	0.320	0.315	0.316	0.313	0.324

(e) Performance comparison using the *p@10* metric for the McMaster testbed

S	10	20	30	40	50	100	200	300	All
Statistical Expansion	0.285	0.285	0.285	0.283	0.283	0.281	0.279	0.278	0.279
Knowledge-Based Expansion without weight boosting	0.285	0.287	0.287	0.291	0.29	0.293	0.286	0.291	0.292
Knowledge-Based Expansion with weight boosting	0.285	0.289	0.287	0.287	0.289	0.289	0.285	0.287	0.289

(f) Performance comparison using the *p@20* metric for the McMaster testbed

Table 14.13: Performance comparison of the two methods under selected expansion sizes using the stem-based VSM

We evaluated the effectiveness of weight boosting and its impact on retrieval performance. The boosting factor β was computed using Eq.(14.8), under the different settings of α = 0.25, 0.5, 0.75, 1, 1.25, 1.5. We present the peak performance of weight boosting in the third row of each sub-table of Table 14.13. For the OHSUMED testbed, boosting helped improve the performance, and the best performance occurred in the range from α = 0.5 to α = 1.25. We note that setting α = 0.5 or = 0.75 generally yields the best boosting effect for the *avgp* metric; setting α = 1 to 1.25 yields better performances for the *p@10* and *p@20* metrics. For the McMaster testbed, weight boosting failed to yield improvements. Further discussion of the weight boosting are presented in [42,49].

We further studied how knowledge-based expansion perform for different query scenarios and experimental results show that the performance varied depending on the query scenario [42,49]. More specifically, the method yields more improvements in scenarios such as `treatment`, `differential diagnosis` and `diagnosis`, whereas it yields less improvements in such scenarios as `complication`, `pathophysiology`, `etiology` and `prognosis`. An explanation of this lies in the different quality of the knowledge structures for these scenarios. The knowledge structures (i.e., the fragments of UMLS Semantic Network such as Figure 14.14) for the latter four scenarios were originally missing in UMLS and were acquired by ourselves from experts. (see the knowledge acquisition process in Section 14.6.5) These acquired structures have more semantic types marked as relevant than those for the former three scenarios. As a result, when handling queries with the latter four scenarios, the knowledge-based method keeps more concepts during the filtering step. Thus, the expansion result for the knowledge-based method resembles that of the statistical expansion method, leading to almost equivalent performance between the two methods and less improvements. Further refinement on the clustering and ranking of the knowledge structures for the four scenarios (i.e., `complication`, `pathophysiology`, `etiology` and `prognosis`) will increase the improvements in retrieval performance.

Choice of α for weight boosting. Experimental results revealed that weight boosting is helpful in improving retrieval performance. Further, the performance of weight boosting is sensitive to the query scenario. Certain query scenarios such as `treatment` and `diagnosis` are associated with more mature knowledge

structures, which requires less expansion concepts. In these scenarios, setting α in between 0.75 and 1.25, which represents more aggressive weight boosting, achieves noticeable improvements. In other scenarios associated with less mature knowledge structures, e.g., complication, the difference is insignificant between the set of expansion concepts by our method and those by statistical expansion. As a result, the cumulative weights of the two set of expansion concepts are close to each other. For such scenarios, our experimental data suggests a more conservative weight boosting with α in the range of 0 to 0.5.

Comparison with previous knowledge-based query expansion studies. In past studies [50-52], researched compared their knowledge-based expansion methods against a baseline generated without expansion. Such studies reported an insignificant improvement [51-52] or even degrading performance [50] compared to the no-expansion method. In contrast, our study compares against a baseline generated by statistical expansion. In our experimental setup, this baseline has an observed improvement over the no-expansion method by 5% to 10%.

In Aronson and Rindflesch's study [53], the researchers applied the UMLS Metathesaurus to automatically expand synonyms to the original query. In one particular case, their approach achieved a 5% improvement over a previous study [54] that applied statistical expansion on the same testbed. This result indicates the value of knowledge-based query expansion. However, their approach is limited to expanding only synonyms instead of scenario-specific terms. Thus the improvement is limited.

Retrieval Performance Using The Phrase-Based VSM

In this section, we compare the performance of knowledge-based query expansion with that of statistical expansion by using the phrase-based VSM for query-document matching. The experiments were performed on the 57 scenario-specific queries in OHSUMED. (Similar results were observed on the McMaster testbed and are excluded from this discussion due to space limits.) The results are shown in Table 14.14, under the three metrics, *avgp*, *p@10* and *p@20*. We present the performance of both knowledge-based query expansion and statistical expansion under selected expansion sizes s. We have also provided the retrieval results for the original queries without expansion, as shown in each row and listed under $s = 0$. From these results, we made the following two major observations:

- With phrase-based VSM, query expansion (both methods) still brings significant improvements for about 10%. For example, both expansion methods yield a peak *avgp* of 0.49 compared to the *avgp* of the no-expansion method which is 0.44.
- Both expansion methods achieve the peak performance when expanding 10 to 20 concepts. This makes it desirable to combine query expansion with the phrase-based VSM, since appending 10 to 20 concepts to the original query incurs a small amount of computation overhead. We note that this is in contrast to the case of using the stem-based VSM in which we need to expand hundreds or thousands of word stems to reach peak performance.

We also noted that the peak performance of the two expansion methods is comparable. That is, expanding 10 to 20 statistically-related concepts is almost as

good as expanding 10 to 20 scenario-specific concepts identified by the knowledge-based method. This is in contrast to the comparison obtained by using the stem-based VSM, where there is significant difference between the two methods. This is mainly due to the ability of the phrase-based VSM in approximately matching distinct concepts. Recall the fact that expanding all statistically-related terms introduces certain heavily-weighted terms which are non-scenario-specific. Using the stem-based VSM which performs strict matching among terms, the existence of such non-scenario-specific terms promotes the ranking of certain non-scenario-specific documents while demoting the ranking of other scenario-specific documents. The phrase-based VSM, however, is able to partially match a non-scenario-specific phrase with a scenario-specific one appearing in a relevant document. Subsequently, the existence of certain non-scenario-specific phrases generated by the statistical expansion no longer negatively impacts the retrieval result.

s	0	10	20	30	40	50	100
Statistical Expansion	0.440	0.486	0.489	0.483	0.479	0.479	0.460
Knowledge-Based Expansion	0.440	0.486	0.490	0.487	0.482	0.485	0.475

(a) Performance comparison using the *avgp* metric

s	0	10	20	30	40	50	100
Statistical Expansion	0.584	0.612	0.604	0.581	0.579	0.567	0.544
Knowledge-Based Expansion	0.584	0.612	0.616	0.604	0.600	0.595	0.586

(b) Performance comparison using the *p@10* metric

s	0	10	20	30	40	50	100
Statistical Expansion	0.504	0.546	0.540	0.532	0.528	0.525	0.496
Knowledge-Based Expansion	0.504	0.538	0.546	0.554	0.543	0.542	0.535

(c) Performance comparison using the *p@20* metric

Table 14.14: Performance comparison of the two methods under various expansion sizes using the phrase-based VSM

We also note that the precision of using the phrase-based VSM without expansion (the first cell in each row of Table 14.14) is significantly higher than that of using the stem-based VSM (the first cell in each row of Table 14.13). Since the phrase-based VSM replies on UMLS, these improvements can be viewed as the results of a first step in applying human knowledge. On top of this, statistical expansion takes another step and applies statistical knowledge derived from a sample corpus to append statistically-correlated concepts. The 5%-10% improvement in precision (e.g., an *avgp* of 0.489 for statistical expansion under s = 20 compared to an *avgp* of 0.440 for no expansion, Table 14.14(a)) suggests that the statistical knowledge is "additive" on top of human knowledge to achieve better retrieval results. Knowledge-based query expansion uses statistical expansion as a starting point, and attempts to further apply UMLS to refine the query expansion results. Nonetheless, since the same knowledge source has already been applied in the form of the phrase-based VSM, this refinement step yields only a small amount (1-2%) of performance improvements.

14.6.4 Computation Complexity Comparison

The computation complexity of knowledge-based expansion is comparable to that of statistical expansion. In the step of deriving expansion terms, the knowledge-based method requires an additional step of going through all statistically-related terms and selecting those that are scenario-specific. This step incurs a complexity that is linear to the number of statistically-related terms. Since the complexity of identifying all statistically-related terms by the statistical method is at least linear to the number of these terms, the additional step in the knowledge-based method does not significantly increase complexity.

In the step of matching an expanded query with documents, the complexity of the knowledge-based method is less than that of the statistical method. The complexity in this step is directly proportional to the number of terms in the expanded query. As revealed by our experiments, knowledge-based expansion requires significantly less expansion terms, which reduce the computation complexity. As a result, the knowledge-based expansion yields comparable retrieval performance with that of statistical expansion.

14.6.5 Knowledge Acquisition

The quality of our knowledge-based method largely depends upon the quality and completeness of the domain-specific knowledge source. The knowledge structure in the UMLS knowledge base is not specifically designed for scenario-specific retrieval. As a result, some frequently asked scenarios (e.g., `etiology` or `complications` of a disease) are either undefined in UMLS, or defined but with incomplete knowledge. Therefore, we present a methodology that consists of the following two steps:

1. Acquire knowledge for undefined scenarios to supplement the UMLS knowledge source.
2. Refine the knowledge of the scenarios defined in the UMLS knowledge source (including the knowledge supplemented by Step 1).

Knowledge Acquisition Methodology

Knowledge Acquisition for Undefined Scenarios. For an undefined scenario, an incomplete relationship graph as shown in Figure 14.16 is presented to medical experts. Edges in this relationship graph are labeled with one of the undefined scenarios, e.g., "etiology." The experts will fill in the question marks with existing UMLS semantic types that fit the relationship. For example, because viruses are related to the etiology of a wide variety of diseases, the semantic type "Virus" will replace one of the question marks in Figure 14.16. This new relationship graph (etiology of diseases) will be appended to the UMLS Semantic Network, and can be used for queries with the "etiology" scenario.

Knowledge Refinement Through Relevance Judgments. A relationship graph for a given scenario (either previously defined by UMLS or newly acquired from Step 1) may be incomplete in including all relevant Semantic Types. A hypothetical example of this incompleteness would be the missing relationship `treats` between `Therapeutic or Preventive Procedure` and `Disease or Syndrome`.

The basic idea in amending this incompleteness is to explore the "implicit" knowledge embedded in the relevance judgments of a IR testbed. Such a testbed typically provides a set of benchmark queries and for each query, a pre-specified set of relevant documents. To amend the knowledge structure for a certain scenario, e.g., `treatment`, we focus on sample queries that are specific to this scenario, e.g., `keratoconus treatment`. We then study the content of documents that are marked as relevant to these queries. From the content, we can identify concepts that are directly relevant to the query's scenario, e.g., `treatment`. If the semantic type for those concepts are missing in the knowledge structure, we can then refine the knowledge structure by adding the corresponding semantic types. For example, let us consider a hypothetical case where the type `Therapeutic or Preventive Procedure` is missing in the knowledge structure of Figure 14.16. If by studying the sample query `keratoconus treatment`, we identify quite a few "`Therapeutic or Preventive Procedure`" concepts appearing in relevant documents such as `penetrating keratoplasty` and `epikeratoplasty`, we are then able to identify `Therapeutic or Preventive Procedure` as a relevant semantic type and append it to Figure 14.16.

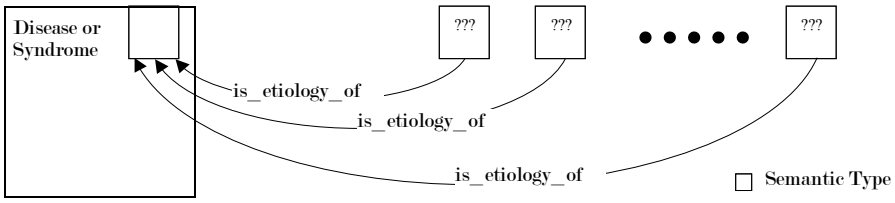


Figure 14.16: A sample template to acquire knowledge for previously undefined scenarios

Given that a typical benchmark query has a long list of relevant documents, it is labor-intensive to study the content of every relevant document. One way to accelerate this process is to first apply an incomplete knowledge structure to perform knowledge-based query expansion and conduct retrieval tests based on such expansion. An incomplete knowledge structure leads to an "imperfect" query expansion, which in turn, fails to retrieve certain relevant documents to the top of the ranked list. Comparing this ranked list with the "gold standard" and identifying the missing relevant documents will give us pointers to determine the incomplete knowledge. For example, failure to include `Therapeutic or Preventive Procedure` in the knowledge structure in Figure 14.14 prevents us from expanding concepts such as `penetrating keratoplasty` to the sample query of `keratoconus, treatment`. As a result, documents with a focus on `penetrating keratoplasty` will be ranked unfavorably low. After we identify such documents, we can discover the missing expansion concepts that are contributing to the low rankings and refine the knowledge structure as we have just described.

Scenarios	# of semantic types defined in UMLS	# of semantic types acquired from experts	# of additional semantic types through knowledge refinement	Total # of semantic types after knowledge acquisition
treatment of a disease	3	N/A	1	4
diagnosis of a disease	5	N/A	2	7
prevention of a disease	3	N/A	0	3
differential diagnosis of a symptom/disease	N/A	10	4	14
Etiology of a disease	N/A	40	1	41
risk factors of a disease	N/A	40	2	42
complications of a disease/medication	N/A	15	0	15
pathophysiology of a disease	N/A	56	0	56
prognosis of a disease	N/A	15	2	17
epidemiology of a disease	N/A	13	0	13
research of a disease	N/A	28	0	28
organisms of a disease	N/A	7	0	7
criteria of medication	N/A	26	0	26
when to perform a medication	N/A	5	6	11
preventive health care for a type of patients	N/A	10	2	12

Table 14.15: Knowledge acquisition results

Knowledge Acquisition Process

The 57 scenario-specific queries (Table 14.12) in the OHSUMED testbed are chosen to apply our proposed knowledge-acquisition method because of the following considerations:

- The OHSUMED queries are collected from physicians patients in a clinical setting. Therefore, the OHSUMED query scenarios should be representative in healthcare, and the knowledge acquired from these scenarios should be broadly applicable.
- The knowledge-acquisition methodology also requires the explorion of relevance judgments for a set of benchmark queries. OHSUMED is the largest testbed for medical free-text retrieval that has relevance judgments for knowledge refinement.

We have identified 12 OHSUMED scenarios whose knowledge structures are missing in UMLS. We applied the two-step knowledge-acquisition method to acquire the knowledge structures for these 12 undefined scenarios and to refine the knowledge structures for all scenarios. During the first step of the acquisition process, we interviewed two intern physicians at the UCLA School of Medicine. During the interview, we first described the meaning of the relationship graphs as shown in Figure 14.16. Next, we presented the entire list of UMLS semantic types to the experts so that appropriate semantic types were filled into the question marks. We communicated the results from one expert to another until they reached a consensus for each scenario. For the second step of knowledge acquisition, we performed retrieval tests on the OHSUMED testbed using both queries expanded by the knowledge-based method and the method of expanding all statistically-related concepts. We focused on 12 queries where the statistical method outperforms the knowledge-based method in terms of the precision in top-10 results. We further applied the method presented in the previous section to study the content of these top-ranked documents and augmented the knowledge structure for the corresponding scenario with appropriate semantic types.

Knowledge Acquisition Results

The acquisition results are shown in Table 14.15. Due to space constraints, we only provide a statistical summary of the results. The scenarios in the first three rows, i.e., `treatment`, `diagnosis` and `prevention`, are defined in UMLS. The first column in these rows shows the number of semantic types marked as relevant for each scenario (i.e., the number of semantic types that experts have filled into the blank rectangles of Figure 14.16). The second column for these rows is "N/A" because there was no need to acquire knowledge structure from domain experts for these scenarios. The third column shows the number of semantic types added during knowledge refinement (the second step of knowledge acquisition). For example, for the `diagnosis` scenario two additional semantic types, `Laboratory or Test Result` and `Biologically Active Substance` were added because of the study on `Query #97: Iron deficiency anemia, which test is best`. These two semantic types were added because the absence of these two types has prevented the knowledge-based method from expanding two critical concepts into the original query: `serum ferritin` and `fe iron`, each belonging to one of the two semantic types. From the relevance judgment set, we noted that missing these two concepts leads to the low ranking of three relevant documents that heavily use these two concepts.

Starting from the fourth row, we list the scenarios for which we need to acquire knowledge structure from domain experts. The first column for these scenarios is "N/A" because these scenarios are originally undefined in UMLS. The second column shows the number of semantic types that experts have filled into the structure template of Figure 14.16. The third column shows the number of additional semantic types from knowledge refinement (the second step of knowledge acquisition), and the last column shows the total number of semantic types after knowledge acquisition.

The proposed knowledge-acquisition method on the OHSUMED testbed has shown to be efficient and effective. We finished communicating with domain experts and acquiring the knowledge structures for the 12 scenarios in less than 20 hours, and spent an additional 20 hours to refine the knowledge structure by exploring the relevance judgments. The augmented knowledge was applied in our experiments presented in Section 14.6.3 and was effective in improving the retrieval performance of the knowledge-based method over the statistical expansion method.

14.6.6 Study of The Relevancy of Expansion Concepts by Domain Experts

Through experiments on the two standard medical text retrieval testbeds, we have observed that under most retrieval settings, knowledge-based query expansion outperforms statistical expansion. Our conjecture is that knowledge-based query expansion selects more specific expansion concepts to the original query's scenario than statistical expansion does. To verify this conjecture, we have requested domain experts to manually evaluate the relevancy of expansion concepts.

The basic idea for this study is the following: For each query in a given retrieval testbed, we apply two query expansion methods to generate two sets of expansion

concepts. We then prepare an evaluation form which inquires about the relevancy of each expansion concept to the original query. In this form, we present the query and ask domain experts to judge the relevancy based on the query's scenario(s). For each concept, we provide four scales of relevancy: *relevant*, *somewhat relevant*, *irrelevant*, or *do not know*. We blind the method used to generate each concept and in doing so, we reduce bias that an expert might have towards a particular method.

To implement this idea, we chose the 57 scenario-specific queries in the OHSUMED testbed. We applied the two expansion methods and derived 40 expansion concepts from each method with the highest weights. We presented the evaluation form consisting of these concepts to three medical experts who are intern doctors at the UCLA School of Medicine. We asked them to make judgments only on those queries that belong to their area of expertise, e.g., oncology, urology, etc. On average, each expert judged the expansion concepts for 15 queries. Thus, for each expansion method, we obtained 1,600 expansion concepts classified into one of the four categories.

Figure 14.17 and Figure 14.18 present a summary of the results from this human subject study. For the expansion concepts derived from each method, we summarized the results into a histogram. The bins of this histogram are the four scales of relevancy. We note that 56.9% of the expansion concepts derived by the knowledge-based method are judged as either *relevant* or *somewhat relevant*, whereas only 38.8% of expansion concepts by statistical expansion are judged similarly. This represents a 46.6% improvement. The results validate that knowledge-based query expansion derives more relevant expansion concepts to the original query scenario(s) than those by statistical expansion, and thus yields improved retrieval performance for scenario-specific queries.

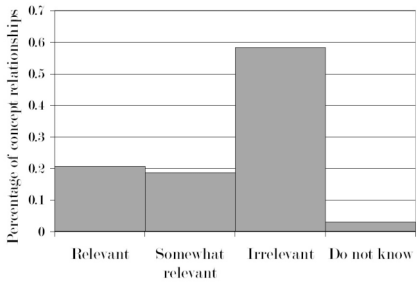


Figure 14.17 Relevancy of statistical expansion concepts

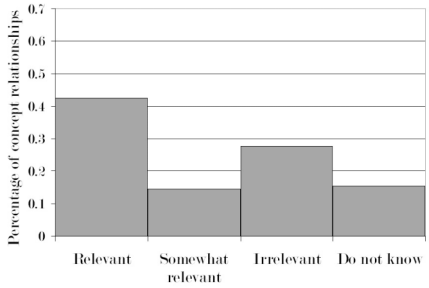


Figure 14.18 Relevancy of knowledge-based expansion concepts

14.7 A SYSTEM ARCHITECTURE FOR RETRIEVING SCENARIO-SPECIFIC FREE TEXT DOCUMENTS

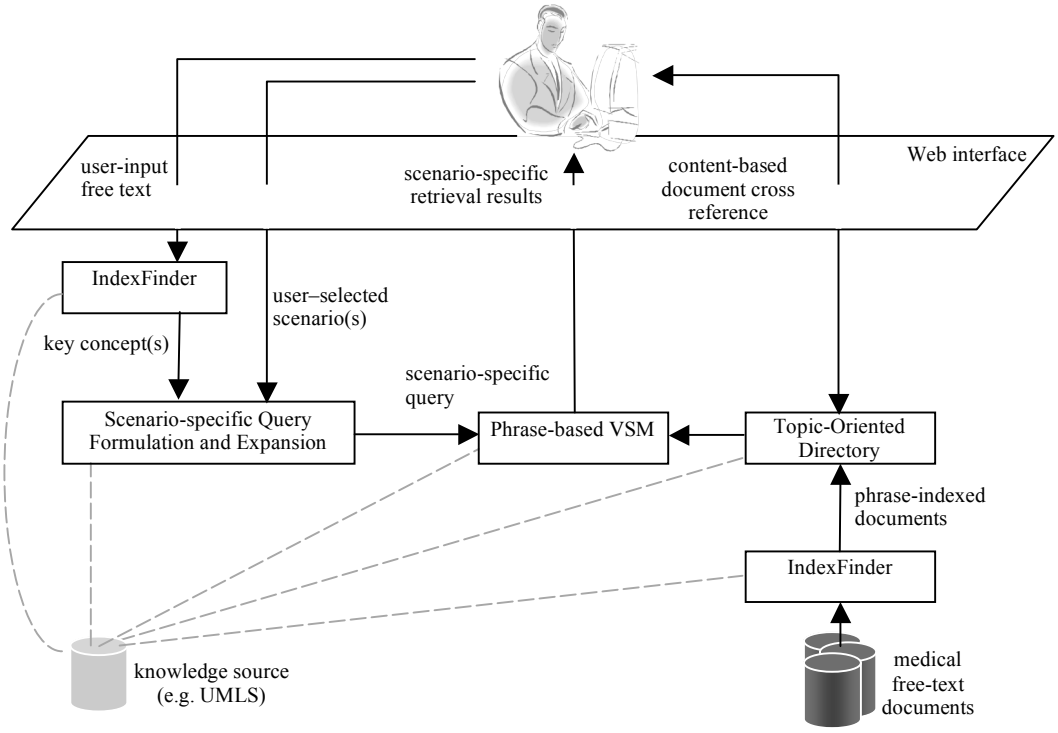


Figure 14.19. The KMeX system architecture

We have implemented and integrated the three proposed techniques in a test bed to provide scenario-specific free-text retrieval (Figure 14.19). 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). Topics and subtopics are then derived by mining the frequently co-occurring features extracted from the documents. With the aid of the knowledge source and the user's query patterns, a topic-oriented directory system can be constructed.

During the retrieval phase, the query expansion module appends the user query with scenario-specific terms. The directory system selects the most relevant topics that match the expanded query. Documents that belong to those topics are submitted to the module which ranks the documents based on their similarity to the query via the phrase-based Vector Space Model (VSM) and return to the users.

14.8 SUMMARY

We have developed a new knowledge-based approach for 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 the traditional technique based on NLP. 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 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 for directory systems, 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 yields 5% - 10% improvements in precision and recall as compared to the statistical query expansion case. Knowledge-based query expansion can be applied together with the Phrase-based VSM. In that case, the peak performance occurred with very few expansion terms (10 to 20) which is a desirable property.

Topics can be generated from mining document features. Based on query templates, and knowledge type hierarchies, free text documents can be organized into a set of scenario specific topic oriented directory systems. In each such directory, the documents are indexed and linked based on the topics. Such topic organization not only improves the retrieval performance for ranking relevant documents but also provides cross-referencing among related topics.

We have implemented a test bed with the above three technologies. Using the UCLA patient reports as a test set, we have shown that *IndexFinder* is able to extract features from free text documents, and data mining algorithms can be used to organize features into topics and are feasible to construct topic oriented directory systems. Our knowledge based query expansion techniques as well as the phase based vector space model can be used in conjunction to significantly improve precission and recall. The scenario specific topic oriented directory systems further improves the retrieval effectiveness as well as to perform content correlation of medical documents.

14.9 EXERCISES

1. Explain why IndexFinder currently limits word combination within a sentence. Discuss the tradeoffs of using other methods of word combination such as phrase, paragraph or word properties (e.g. part of speech).
2. Discuss why semantic filtering is important in improving the retrieval quality for IndexFinder.
3. Discuss how to handle negation concept in the IndexFinder.
4. List the reasons why the knowledge-based query expansion technique performs better than the statistical expansion.
5. Discuss what type of queries the knowledge-based query expansion method in this chapter may not yield significant retrieval performance improvements over that of the statistical expansion cases; Suggest ways to improve such queries. (Hint: non-scenario-specific queries.)
6. Discuss why the retrieval performance of statistical query expansion improves as the number of expansion terms increases and then the performance degrades with expansion after it reaches a certain size, while the knowledge based expansion does not have such a behavior.
7. Discuss why the phrase-based vector space model alone (without applying query expansion) yields similar performance as that of the combination of knowledge-based query expansion and the stem-based vector space model.
8. Explain why the expansion size required to reach optimal performance using phrase-based vector space model is much smaller than using the stem-based vector space model.
9. What is the computation complexity of phrase vector space model? Suggest methods to reduce the computation complexity.
10. Describe the concept of topics directory. How does topic directory compliment search technique to improve document retrieval performance.
11. Discuss what are the additional tasks and research issues needed to extend the knowledge-based document retrieval methods used in this chapter (i.e., IndexFinder, the phrase-based VSM, knowledge-based query expansion, and topic-oriented directory) to application domains other than medicine and healthcare.

14.10 ACKNOWLEDGEMENT

This research is supported in part by NIC/NIH Grant #4442511-33780. We thank Dr. Hooshang Kangarloo, Dr. Denise Aberle, Dr. Suzie El-Saden, Dr. Craig Morioka, Dr.

Andrew Chen and Dr. Blaine Kristo from the UCLA School of Medicine for stimulating discussions and insightful comments. We also thank Nancy Wilczynski and Dr. Brian Haynes from the Health Information Research Unit (HIRU) at McMaster University for sharing their valuable dataset for our experimental evaluation.

14.11 BIBLIOGRAPHY AND REFERENCES

1. http://www.nlm.nih/gov/pubs/factsheets/
2. S. Chu. Yearbook of Medical Informatics, 2002
3. W.W. Chu. Cooperative Information Systems. Encyclopedia of Electrical and Electronic Engineering, edited by J. G. Webster, John Wiley & Son, Inc., 1998
4. 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
5. 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
6. National Library of Medicine, UMLS Knowledge Sources, 14th edition, 2003
7. 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
8. 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.
9. 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.
10. 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.
11. 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
12. Alan R. Aronson, Effective Mapping of Biomedical Text to the UMLS Metathesaurus: The MetaMap Program. In Proc. of AMIA Annual Symp 2001, 2001.
13. 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
14. C. Friedman and G. Hripcsak. Evaluating natural language processors in the clinical domain. Methods of Information in Medicine, 37(4/5): 334-344, 1998.
15. 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
16. 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
17. 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
18. 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
19. R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In Proceedings of the 20th VLDB Conference, Santiago, Chile, 1994
20. J. Han, J. Pei, and Y. Yin. Mining Frequent Patterns without Candidate Generation, Proc. 2000 ACM SIGMOD Int. Conf. on Management of Data (SIGMOD'00), Dallas, TX, May 2000.
21. Qinghua Zou, Wesley Chu, Baojing Lu. SmartMiner: A Depth First Algorithm Guided by Tail Information for Mining Maximal Frequent Itemsets. In *Proc. of the IEEE International Conference on Data Mining*, Japan, Dec 2002
22. G. Salton, A. Wang, and C. S. Yang. A Vector Space Model for Automatic Indexing Communication of the ACM, 18(11): 613-620, 1975
23. Julio Gonzalo, Felisa Verdejo, Irina Chugur, Juan M.Cigarran. Indexing with WordNet synsets can improve Text Retrieval, In *Proceedings of the COLING/ACL'98 Workshop on Usage of WordNet for NLP*,

Montreal, 1998
24. National Library of Medicine, UMLS Knowledge Sources, The Metathesaurus, http://www.nlm.nih.gov/research/umls/meta2.html
25. John Lyons. Semantics, Cambridge University Press, 1977
26. Nancy Ide and Jean Veronis, Word Sense Disambiguation: The State of Art, Computational Linguistics, 24(1): 1-40, 1998
27. 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
28. C.J. van Rijsbergen Information Retrieval, Butterworths, 1979
29. G. Salton and M. J. McGill, Introduction to Modern Information Retrieval, McGraw-Hill Computer Science Series, McGraw-Hill, Inc, 1983
30. M. Steinbach, G. Karypis, and V. Kumar, A comparision of document Clustering Techniques, In Proc of the KDD Work Shop on Text Mining, 2000
31. Ying Zhao and George Karypis Evaluation of Hierarchical Clustering Algorithms for Document Datasets, TR 02-022, Dept. of Computer Science, U. of Minnesota, 2002
32. National Library of Medicine. UMLS Knowledge Sources, 12th edition, 2001
33. E.N. Efthimiadis. Query expansion. Annual Review of Information Science and Technology, 31:121-187, 1996.
34. M. Mitra, A. Singhal, and C. Buckley. Improving automatic query expansion. In *Proceedings of ACM SIGIR '98*, 1998.
35. Y. Qiu and H.P. Frei. Concept-based query expansion. In Proc. 16th ACM-SIGIR, pages 160-169, 1993
36. Y. Jing and W.B. Croft. An association thesaurus for information retrieval. In Proc. RIAO'94, pages 146-160, 1994
37. J. Xu and W.B. Croft. Query expansion using local and global document analysis. In Proc. 19th ACM-SIGIR, pages 4-11, 1996
38. E.N. Efthimiadis and P. Biron. UCLA-okapi at TREC-2: Query expansion experiments. In *Proceedings of the Second Text REtrieval Conference (TREC-2)*, 1993.
39. C. Buckley, G. Salton, J. Allan, and A. Singhal. Automatic query expansion using SMART: TREC-3. In *Proceedings of the Third Text REtrieval Conference (TREC-3)*, 1994.
40. S.E. Robertson, S. Walker, S. Jones, M.M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC-3. In *Proceedings of the Third Text REtrieval Conference (TREC-3)*, 1994
41. C. Buckley, A. Singhal, M. Mitra, and G. Salton. New retrieval approaches using SMART: TREC-4. In *Proceedings of the Fourth Text REtrieval Conference (TREC-4)*, 1995.
42. Z.L. Liu and W.W. Chu. Knowledge-Based Query Expansion to Support Scenario-Specific Retrieval of Medical Free Text. Technical Report #060019, Computer Science Department, UCLA, ftp://ftp.cs.ucla.edu/tech-report/2006-reports/060019.pdf, 2006.
43. N.L. Wilczynski, K.A. McKibbon, and R.B. Haynes. Enhancing retrieval of best evidence for health care from bibliographic databases: calibration of the hand search of the literature. *International Journal of Medical Informatics*, 10(1):390–393, 2001.
44. S.-L. Wong, N.L.Wilczyski, R.B. Haynes, and R. Ramkissoonsingh. Developing optimal search strategies for detecting sound clinical prediction studies in MEDLINE. In *Proceedings of AMIA Annual Symp 2003*, 2003.
45. N.L. Wilczynski and R.B. Haynes. Developing optimal search strategies for detecting sound clinically sound causation studies in MEDLINE. In *Proceedings of AMIA Annual Symp 2003*, 2003.
46. V.M. Montori, N.L. Wilczynski, D. Morgan, and R.B. Haynes. Systematic reviews: A cross-sectional study of location and citation counts. *BMC Medicine*, 1(2), 2003.
47. J.B. Lovins. Development of a stemming algorithm. Mechanical Translation and Computational Linguistics, 11 1-2):22-31, 1968
48. G. Salton and C. Buckley. Term weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523, 1988.
49. Z.L. Liu and W.W. Chu. Knowledge-Based Query Expansion to Support Scenario-Specific Retrieval of Medical Free Text. In *Proceedings of ACM SAC 2005*.
50. W.H. Hersh, S. Price, and L. Donohoe. Assessing thesaurus-based query expansion using the UMLS metathesaurus. In *Proceedings of AMIA Annual Symp 2000*, 2000.

51. R.M. Plovnick and Q.T. Zeng. Reformulation of consumer health queries with professional terminology: a pilot study. *Journal of Medical Internet Research*, 6(3), 2004
52, Y. Guo, H. Harkema, and R. Gaizauskas. Sheffield university and the trec 2004 genomics track: Query expansion using synonymous terms. In *Proceedings of the Thirteenth Text Retrieval Conference (TREC-13)*, 2004.
53. A.R. Aronson and T.C. Rindflesch. Query expansion using the UMLS. In *Proceedings of AMIA Annual Symp 1997*, 1997.
54. P. Srinivasan. Query expansion and MEDLINE. *Information Processing and Management*, 32(4):431–443, 1996.