A Concept-based Framework for Passage Retrieval in Genomics

Wei Zhou, Clement T. Yu
Department of Computer Science
University of Illinois at Chicago
{wzhou1,yu}@cs.uic.edu

Vetle I. Torvik, Neil R. Smalheiser
Department of Psychiatry and Psychiatric Institute
University of Illinois at Chicago
{vtorvik,neils}@uic.edu

Abstract

The task of TREC 2006 Genomics Track is to retrieve passages (from part to paragraph) from full-text HTML biomedical journal papers to answer the structured questions from real biologists. A system for such task needs to be able to parse the HTML free-texts (convert the HTML free-texts into plain texts) and pinpoint the most relevant passage(s) within documents for the specified question. This task is accomplished in three steps in our system. The first step is to parse the HTML articles and partition them into paragraphs. The second step is to retrieve the relevant paragraphs. The third step is to identify the most relevant passages within paragraphs and finally rank those passages. We are interested in **1**. How does a concept-based IR model perform on structured queries comparing to Okapi? **2**. Will the query expansion based on domain knowledge increase retrieval effectiveness? **3**. Will our abbreviation database from MEDLINE help improve query expansion and will the abbreviation disambiguation help improve the ranking? The experiment results show that our concept-based IR model works better than the Okapi; query expansion based on domain knowledge is important, especially for those queries with very few relevant documents; an abbreviation database for query expansion and disambiguation is helpful for passage retrieval.

1. Step 1: HTML parsing and document partitioning.

The HTML full-text journal articles use special symbols to represent Latin characters. For example, the Latin character β can be written as "&#223" or "&szlig". Since this kind of Latin characters are commonly used for naming genes, it is important to translate them into plain texts (e.g., replace "&#223" with "beta"). We used the ISO 8859-1 (Latin-1) characters list for the translation.

Sometimes GIF files are used to represent the Latin characters. For example, the HTML tag <IMG SRC="/math/12pt/Large/beta.gif" ALIGN=BASELINE ALT="beta"> is used in one of the articles to represent β. Notice that term "beta" is included inside the HTML tag and special processing is needed in the tag handler of the HTML parser. In the end, step 1 will partition each document into paragraphs according to the HTML <p> tag.

2. Step 2: Paragraph retrieval

This step retrieves the top 2,000 most relevant paragraphs, which will be used as the input of step 3 for passage retrieval. Several techniques have been conducted in this step. These techniques are explained in detail in the following sections.

2.1 Conditional Porter stemming

Stemming is employed to recognize variants of the same word, which will also reduce the number of terms indexed. Porter stemmer [1] is widely used in IR community. To determine whether this stemmer is suitable to process biomedical texts, we applied it to both the queries and documents and examined the following two aspects:

1. How are the query words stemmed?

2. What are those words that are equal to the query words after they are stemmed?

We observed that Porter stemmer is suitable in most of cases except the following two:

Case 1: A gene name in the query is changed into a totally different non-gene word after stemming. For example, the stemmer changes gene "Pes" to "Pe". Gene "IDE" is stemmed into "ID".

Case 2: A non-gene word becomes a gene name after stemming. For example, "IDEE", after stemming, becomes the gene "IDE".

Gene names in the queries are very important biomedical concepts. The performance of an IR system could be extremely degraded if the above two cases happen. To utilize the Porter stemmer and avoid the above two cases, we employed a conditional stemming strategy. With this strategy, whether a word should be stemmed or not depends upon both the original word and the potential stemmed word. Given a word *w,* we classify *w* into the following 3 categories:

1. ***G***: gene names. This category is further divided into two sub-categories: $\boldsymbol{G_1}$: gene names ended with numbers (e.g., "HNF4") and $\boldsymbol{G_2}$: gene names not ended with numbers (e.g., "APC").
2. ***E***: regular English words
3. ***N***: neither gene names nor English words.

w is a gene name if it is in the Entrez Gene database. w is an English word if it is in the WordNet dictionary. w could be both a $\boldsymbol{G}$ and a $\boldsymbol{E}$, for example "shot" is a common English word and also a gene name. The conditional Porter stemming strategy is given as follows: suppose a word w is potentially stemmed into w' by the Porter stemmer. Stemming is **skipped** if any of the following conditions is satisfied:

1) $w \in \boldsymbol{G}$
2) $(w \in \boldsymbol{E}) \wedge (w \notin \boldsymbol{G}) \wedge (w' \in \boldsymbol{G_2})$
3) $(w \in \boldsymbol{N}) \wedge (w' \in \boldsymbol{G_2})$

There is an **exception** for condition 3. We observed that the stemming "GSTMs" → "GSTM" is acceptable, in which $w \in \boldsymbol{N}$, $w' \in \boldsymbol{G_2}$, w' is ended with an uppercase alphabetic character, and $w = w' + s$. This exception is only for "s" because usually the plural of a gene name is formed by adding an "s" at the end of the gene name. Condition 1 will guarantee that case 1 be avoided. Condition 2 will make sure case 2 does not happen for English words. Condition 3 will avoid case 2 for words that are neither gene names nor English words.

2.2 Handling lexical variants of gene names

New gene names and their lexical variants are regularly introduced into the biomedical literature [2, 3]. However, many reference databases, such as UMLS and Entrez Gene (formerly named as LocusLink), may not be able to keep track of all this kind of variants. [4, 5] have demonstrated that expanding gene names with their lexical variants will improve the performance of information retrieval of bio-medical literature. In our system, the lexical variants of a gene come from two sources: **1**) automatically generated according to a usual strategy [4, 5] based on the features of the guidelines for human gene nomenclature [6]; **2**) or retrieved from an abbreviation database created from MEDLINE. Here we only explain the 2nd strategy.

ADAM [7] is an abbreviation database which covers frequently used abbreviations and their definitions (or long-forms) within MEDLINE titles and abstracts, including both acronyms and non-acronym abbreviations. An important feature of ADAM is that morphologically similar abbreviations are clustered together. For example, "5HT" is an abbreviation for "5-Hydroxytryptamine". ADAM shows that "5-Hydroxytryptamine" could be abbreviated as "5-HT", "5HT", "5-ht", "5-Ht", and "5-H-T" in the literature. This feature of ADAM could be used to find extra lexical variants of gene names, other than those that are automatically generated.

A gene abbreviation could have a non-gene long-form. For example, gene "APC" could be an abbreviation for "Air Pollution Control". To find the lexical variants of a gene abbreviation, we first need to know what the long-form of the gene is. This is accomplished either by extracting the long-form from the queries themselves or searching the Entrez Gene database.

After the long-form of the abbreviation gene is identified, we search the long-form in ADAM. Suppose the abbreviation gene is "PRNP" and we identified "Prion protein gene" as the long-form from the Entrez Gene database. From ADAM, we find "PRNP" and "Prion protein gene" is an abbreviation/long-form pair and "Prion protein gene" could be shortened as "Prnp", "prn-p", or "prnp" ("PRNP", "Prnp", and "prnp" are the same after tokenization). Notice that "prn-p" will not be automatically generated by the 1st strategy.

2.3 Utilizing domain knowledge

Domain knowledge is critical for query expansion. A biomedical term may have many different ways of saying it. For example, "high blood pressure" and "hypertension" refer to the same vascular disease. "hypocretin-2 receptor" and "orexin B receptor" are two different ways of saying the same receptor. This phenomenon is very common in the biomedical literature. Acquisition and utilization of this kind of domain knowledge will be very useful for retrieving more relevant documents. We define a concept as a biomedical meaning or sense [8]. We consider 1) a gene and its synonym set refer to the same concept. 2) a medical subject heading (MeSH) and its synonym set refer to the same concept.

2.3.1 Identifying query concepts

A query usually contains several concepts. For example, the query "purification of rat IgM" has 3 concepts: "purification", "rat", and "IgM". This section will describe how to automatically identify concepts from a query. A concept, in our system, could be a gene name or a MeSH term. Basically we need to identify gene names and MeSH terms from a query. The gene names, in the queries of the genomics track of TREC 2006, are already specified by the query templates [9]. To extract the MeSH terms from a query, we utilize the PubMed Automatic Term Mapping [10]. Given a query, we submit the whole query to PubMed. PubMed will then return a file in which the MeSH terms in the query are marked.

2.3.2 Retrieving concept information from biomedical thesaurus

For each MeSH term, the MeSH database gives its synonyms, hypernyms (more generic terms), and hyponyms (more specific terms). For each gene name, we retrieve its synonyms from the Entrez Gene database. We also collected 22,446 gene names from the UMLS (they are concepts that map to "Gene or Genome" semantic type). Synonyms of these genes were retrieved from UMLS.

2.3.3 Finding related concepts

Related concepts (not synonyms, hypernyms or hyponyms), in some cases, could be very useful. For example, a query is asking for the information about "the gene HNF4 and COUP-tf I in the suppression in the function of the liver". We observed that some relevant documents in the 2005 document collection are talking about the role of "HNF4" and "COUP-tf I" in regulating "hepatitis B virus" transcription. However, "hepatitis B virus" is known as a virus that could cause serious damage to the function

of liver. The relationships among these three concepts could be described as

$$A \leftrightarrow B \leftrightarrow C$$

where "↔" indicates the two concepts on two sides are related. In the above example, *A* is the "HNF4" and "COUP-tf I". *B* is "hepatitis B virus" and *C* is "liver". The query is asking for *A* and *C*, but the relevant document is about *A* and *B*. Queries from TREC genomics track were collected from real biologists. Some of the queries represent the information needs of their latest research. There may be very few relevant papers exist in the literature. Related concepts could be very useful for this kind of queries. This issue is very related to [11] for identifying implicit relationships between two disjoint literatures.

A **semantic type** is a category assigned to concepts based on their intrinsic and functional properties [12]. We require the related concept *B* be related to *A* and *C* not only in free-texts (i.e., *B* co-occurs frequently with both *A* and *C* in the documents), but also on the semantic level (i.e., the semantic type of *B* interconnects with both the semantic type of *A* and the semantic type of *C*).

For any concept *X*, let *S*(*X*) be the set of semantic types of *X* in the UMLS semantic networks. Given two concepts *A* and *C*, algorithm 1 describes a method of finding the related concepts *B*:

Algorithm 1 Finding related concepts

1. $U \leftarrow$ the semantic types that are related to both $T_1 \in S(A)$ and $T_2 \in S(C)$
2. $B_1 \leftarrow$ the concepts that co-occur with *A* in free-texts
3. $B_2 \leftarrow$ the concepts that co-occur with *C* in free-texts
4. $B \leftarrow B_1 \cap B_2$
5. Filter *B* by removing *X* with $X \in B$ **and** $S(X) \subsetneq U$

The size of *B* could be very big. In practice we only need those concepts that are most related to *A* and *C*. For each $X \in B$, we assign a score which will be used to indicate the association of *X* with *A* and *C*.

Given two concepts C_1 and C_2, we use the mutual information [13] to measure their association:

$$I(C_1, C_2) = \log \frac{p(C_1, C_2)}{p(C_1) \times p(C_2)} \qquad (1)$$

where $p(C_1,C_2)$ is the joint probability of C_1 and C_2, and $p(C_1)$ and $p(C_2)$ are the probabilities of C_1 and C_2, respectively. Given a concept *X*, its probability $p(X)$ is given by:

$$p(X) = \frac{f(X)}{N}$$

where $f(X)$ is the document frequency of concept *X* (i.e., the number of documents that mention *X*). *N* is the size of the document collection.

For each $X \in B$, we have a vector $V(X) = [I(X,A)\ I(X,C)]$. Let $R = \{V(X): X \in B\}$. To rank all the concepts in *B*, we define an operator $\preceq$ to compare two vectors. Given two vectors $V_1 = [x_1, y_1]$ and $V_2 = [x_2, y_2]$, we say $V_1 \preceq V_2$ or V_1 is not greater than V_2 if $x_1 \le x_2$ and $y_1 \le y_2$.

For each $X \in B$, a score is assigned based on its vector:

$$score(X) = \frac{\left|\{X' : X' \in R \text{ and } V(X') \preceq V(X)\}\right|}{\left|\{X'' : X'' \in R \text{ and } V(X) \preceq V(X'')\}\right|} \qquad (2)$$

The numerator is the number of the vectors in *R* that are not greater than *X* and the denominator is the number of the vectors in *R* that are not less than *X*.

After the concepts in *B* are ranked, the top **5** are selected as the most related concepts to *A* and *C* (see section 2.5.1 for how these top 5 selected related concepts are used for query expansion).

2.4 Concept retrieval

Okapi relevance scoring formula [14] is known to embody a good model of relevance based upon term occurrences within text documents and the length of the documents. However, Okapi, in general, does not work well if the query contains multiple concepts. For example, one of the queries in the genomics track is asking for the role of gene "PRNP" in "Mad cow disease". Suppose that the concept "Mad cow disease" occurs several times within a document d_1 and the other concept "PRNP" is not mentioned in this document. In addition, d_1 is short in length (the number of words). Another document d_2 mentions both "Mad cow disease" and "PRNP". Suppose these two concepts occur once in d_2 and d_2 is long. In this case, Okapi is likely to rank d_1 higher than d_2 because of the bigger frequency of "Mad cow disease" and the fewer words in d_1. However, d_2 is more likely to be a relevant document because it covers all the query concepts.

To fix this problem, we use concept retrieval model. Given a query *q* and a document *d*, two similarities are computed. One is their concept similarity $\underset{concept}{sim}(q,d)$ and the other word similarity $\underset{word}{sim}(q,d)$.

$$sim(q,d) = [\underset{concept}{sim}(q,d),\ \underset{word}{sim}(q,d)] \qquad (3)$$

A query *q* is associated with a vector *v*:

$$v = (\vec{v_1},\ \vec{v_2})$$

where $\vec{v_1}$ is a vector of all the concepts in *q* and $\vec{v_2}$ is a vector of all the words in *q*:

$$\vec{v_1} = (c_1, c_2, ..., c_i)$$
$$\vec{v_2} = (w_1, w_2, ..., w_j)$$

$\underset{concept}{sim}(q,d)$ is computed as

$$\underset{concept}{sim}(q,d) = \sum_{c \in v_1} \log \frac{N}{n} \qquad (4)$$

where *N* is the size of the document collection and *n* is the document frequency of the concept *c*.

$\underset{word}{sim}(q,d)$ is computed using Okapi:

$$\underset{word}{sim}(q,d) = \sum_{w \in v_2} \log\left(\frac{N - n + 0.5}{n + 0.5}\right) \frac{(k_1 + 1)tf}{K + tf} \qquad (5)$$

where

N is the size of the document collection

n is the number of documents containing *w*

$K = k_1 \times ((1-b) + b \times \frac{dl}{avdl})$ and k_1=1.2, b=0.75 are constants. dl is the length of the document and $avdl$ is the average document length.
tf is the term frequency within a document.

Given two documents d_1 and d_2, we say

$$sim(q, d_1) > sim(q, d_2)$$

or d_1 will be ranked higher than d_2, with respect to the same query q, if either

1) $sim_{concept}(q, d_1) > sim_{concept}(q, d_2)$ **OR**
2) $sim_{concept}(q, d_1) = sim_{concept}(q, d_2)$, and $sim_{word}(q, d_1) > sim_{word}(q, d_2)$

This concept retrieval model emphasizes the concept similarity more than the word similarity since we believe that a relevant document should first contain the query concepts. [15] has demonstrated the effectiveness of this model in improving the performance of information retrieval in the Robust track of TREC.

2.5 Query expansion

A biomedical concept could have many different ways of saying it. As such, query expansion is critical for improving the performance of IR systems in the biomedical literature. In this section, we describe how the gene lexical variants (section 2.2) and the domain knowledge (section 2.3) are utilized for query expansion and how the query expansion is implemented in the IR model described in section 2.4.

2.5.1 Weighting

Suppose "high blood pressure" is a concept in query q. Document d_1 contains "high blood pressure" and d_2 contains "hypertension", which is a synonym of "high blood pressure". When computing $sim(q, d_1)$ and $sim(q, d_2)$, d_1 and d_2 should receive the same amount of weight on concept "high blood pressure". For different types of related concepts, we use different strategies to assign appropriate weights when they are added into the query, as described in the following:

1. Lexical variants of a gene name (either automatically generated or retrieved from the abbreviation database) are assigned the same weight as the original gene name.
2. Synonyms and hyponyms of a concept are assigned the same weight as the original concept.
3. Hypernyms of a concept are assigned partial weight of the original concept. The ratio α is set at 0.95.
4. For a selected $X \in B$ with $V(X) = [I(X,A)\ I(X,C)]$ (see section 2.3), if $I(X,A) > I(X,C)$, then X is assigned a partial weight of concept A, otherwise X is assigned a partial weight of concept C. The ratio β is given by: $\beta = 0.9 \times \frac{5 - i + 1}{5}$, where 5 is the number of selected top ranked concepts that are most related to concept A and C. i is the position of concept X in the ranking.

2.5.2 Updating the concept similarity

Given a concept c, it could be expanded into a vector u,

$$u = (c, \vec{u_1}, \vec{u_2}, \vec{u_3}) \quad (7)$$

where $\vec{u_1}$ is a vector of terms that have the same weight as concept c, consisting of synonyms, hyponyms, or lexical variants. $\vec{u_2}$ is a vector of hypernyms of c and $\vec{u_3}$ is a vector of indirectly related concepts of c found according to section 2.3. The log inverse document frequency of c is updated as

$$weight(c) = \log \frac{N}{\left| \bigcup_{t \in \vec{u_1}} \{d : t \in d\} \right|} \quad (8)$$

where $\left| \bigcup_{t \in \vec{u_1}} \{d : t \in d\} \right|$ is the number of distinct documents having any of terms in $\vec{u_1}$.

Given a document d and a concept c, the weight $w_{c,d}$ that d achieves from c is given by:

$$w_{c,d} = \max\{w_t : t \in u, \text{ and } t \in d\}$$

where $w_t = weight(c)$ if $t \in \vec{u_1}$; $w_t = \alpha \times weight(c)$ if $t \in \vec{u_2}$; $w_t = \beta \times weight(c)$ if $t \in \vec{u_3}$. α and β are given in section 2.5.1.

Comment 1: we do not implement the query expansion in the $sim_{word}(q, d)$ because the word similarity in our IR model is secondary.

Comment 2: we use window sizes to determine whether a phrase concept occurs within a document. If the phrase concept is a gene name, such as "MMS 2", we require that "MMS" and "2" co-occur exactly adjacent to each other and "MMS" is on the left of "2". If the phrase concept is not a gene name, all the words in that phrase are required to co-occur within $N = (n+(n-1)\times 2-1)$ word distance of the same sentence, where n is the number of words in the phrase concept. This window size will only allow at most two extra words in between two adjacent words in the phrase concept. For example, in the text "…human cells including those from colon, breast, and lung cancer …", the words in the concept "colon cancer" do not appear exactly adjacent to each other. But there are only 2 words in between them (word "and" is a stop word). As such, we say that the phrase concept "colon cancer" occurs in this document.

2.6 Pesudo-feedback

Pseudo-feedback is a technique commonly used to improve retrieval performance assuming that the top-ranked documents are relevant, so that new terms are added into the original query. We used a modified pseudo-feedback strategy described in [16].

First, retrieve all the concepts from the top 15 ranked documents. For each of these 15 documents, it is first partitioned into sentences. For each sentence, the concepts are extracted in the same way the query concepts are extracted (see section 2.3.1)

Second, for each concept c in the top ranked documents, compute the similarity $sim(q,c)$ between the whole query q and the concept c. The computation of $sim(q,c)$ can be found in [16].

Third, the top 5 ranked concepts (according to $sim(q,c)$) are selected.

Fourth, for each selected concept c', compute the global correlation between c' and each concept in query q using equation 1. Associate c' with the concept $c_q \in q$ that has the highest global correlation. If c' and c_q have the same semantic type, add c' as one of alternatives of concept c_q. A document having c' will receive a weight given by: $0.5 \times \frac{5-i+1}{5} \times weight(c_q)$, where i is the position of c' in the ranking of the second step.

2.7 Avoid incorrect match of abbreviations

Some gene names are very short, such as gene "Pes". "Pes" could be the abbreviation for many non-gene long-forms, such as "Pentaspan", "Phenol-extracted slime", or "primary empty sella".

Given an abbreviation X with the long-form L, we scan the top k (k=1000) documents and when a document is found with X, we compare L with all the long-forms of X in that document. If none of these long-forms is equal to L, we cut the concept similarity of X from $sim_{concept}(q,d)$. The program of extracting abbreviations and their long-forms from free-texts is downloaded from [17].

3. Step 3: Passage retrieval

This step takes the output of the step 2, the top 2,000 relevant paragraphs, and 1) identify the optimal passage within each paragraph; 2) rank the extracted optimal passages and output the top 1,000 most relevant passages; and finally 3) find the span of the passages in the original HTML full-text documents.

The criterion for the optimal passage in a paragraph is given by:

> "**The part of the paragraph that has the fewest continuous sentences and covers the maximum number of distinct query concepts**."

The ranking of passages is similar to the ranking of paragraphs. For each passage, we computed the term similarity and the concept similarity. The concept retrieval model described in section 2.4 is then applied for the ranking.

The final step is to link the passages to the source HTML full-text documents. For each passage, its maximum-length legal span given by the official file "legalspans.txt" (a file which includes all of the maximum-length legal spans for the collection) is examined and the actual span of that passage in the maximum-length legal span was identified.

4. Experiment results

We submitted 3 runs. In the first run, the top 1,000 most relevant paragraphs were returned as the passages. In the second run, the passage in a paragraph that has the fewest continuous sentences and covers the maximum number of query concepts is considered as the optimal passage. In the third run, the programs for identifying the optimal passages and finding the actual spans of each passage in the source documents were refined. The result is shown in Table 1. It is interesting to notice that our first run UICgen1 (simply return the whole paragraphs as the passages) achieved better performance than the other two on document level and aspect level. This is probably because a single document might contain multiple relevant passages.

Table 1: Experiment results

Document			
	MAP	# best	# > Median
UICgen1	0.5439	3	25
UICgen2	0.5268	2	25
UICgen3	0.5320	3	25

Passage			
	MAP	# best	# > Median
UICgen1	0.0750	0	25
UICgen2	0.1243	0	25
UICgen3	0.1479	7	25

Aspect			
	MAP	# best	# > Median
UICgen1	0.4411	7	25
UICgen2	0.3478	1	23
UICgen3	0.3492	1	24

A series of new experiments were performed after the conference to examine how each type of domain-specific knowledge contributes to the retrieval performance. A baseline was established using the basic conceptual IR model without incorporating any type of domain-specific knowledge. Then five runs were conducted by adding each individual type of domain-specific knowledge. We also conducted a run by adding all types of domain-specific knowledge. Results of these experiments are shown in Table 2.

We found that any available type of domain-specific knowledge improved the performance in passage retrieval. The biggest improvement comes from the lexical variants, which is consistent with the result reported in [4]. This result also indicates that biologists are likely to use different variants of the same concept according to their own writing preferences and these variants might not be collected in the existing biomedical thesauruses. It also suggests that the biomedical IR systems can benefit from the domain-specific knowledge extracted from the literature by text mining systems.

Table 2 Contribution of different types of domain-specific knowledge

Run	Passage	Aspect	Document
Baseline	0.084	0.233	0.359
Baseline+Synonyms	0.105 (+25%)	0.246 (+5.6%)	0.420 (+17%)
Baseline+Hypernyms	0.088 (+4.8%)	0.225 (-3.4%)	0.390 (+8.6%)
Baseline+Hyponyms	0.087 (+3.6%)	0.217 (-6.9%)	0.389 (+8.4%)
Baseline+Variants [1]	0.150 (+78.6%)	0.348 (+49.4%)	0.495 (+37.9%)
Baseline+Related	0.086 (2.4%)	0.220 (-5.6%)	0.387 (+7.8%)
Baseline+All	**0.174 (107%)**	**0.380 (+63.1%)**	**0.537 (+49.6%)**

[1]. Lexical variants of gene symbols or abbreviations of MeSH terms.

Synonyms provided the second biggest improvement. Hypernyms, hyponyms, and implicitly related concepts provided similar degrees of improvement. The overall performance is an accumulative result of adding different types of domain-specific knowledge and it is better than any individual addition. It is clearly shown that the performance is significantly improved (107% on passage level, 63.1% on aspect level, and 49.6% on document level) when the domain-specific knowledge is appropriately incorporated. Although it is not explicitly shown in Table 4.2.3, different types of domain-specific knowledge affect different subsets of queries. More pacifically, each of these types (with the exception of "the lexical variants" which affects a large number of queries) affects only a few queries. But for those affected queries, their improvement is significant. As a consequence, the accumulative improvement is very significant.

Future work

Five possible improvements will be investigated in the future research:

1. Term alignment. Term variation (i.e., expressing a single concept in a number of different ways), is very common in the biomedical literature. Our second improvement is to handle a certain type of term variation. For example, terms “ImmunoPrecipitation” (appear in 23,775 PubMed citations as of 08/23/2006), “Immune Precipitation” (404), “Immuno Precipitation” (177), and “ImmunePrecipitation” (12) all refer to the same concept. The UMLS only includes “ImmunoPrecipitation” and “Immune Precipitation”. These terms are slightly different morphologically. Fortunately, they all have the same abbreviation “IP”.

2. Google/Wikipedia feedback. This work will investigate whether the feedback of the Web and a widely used encyclopedia could be used to improve the retrieval performance.

3. Automatically retrieve the complex concepts in a query. The queries in the genomics track of TREC 2006 are formulated from templates. In the future research, we want to automatically identify the complex concepts in the queries in general.

4. Utilize the homologs among the genes across different species. A homolog is a gene from one species, for example the mouse, that has a common origin and functions the same as a gene from another species, for example, humans, Drosophila, or yeast. [Source: NHBLI/NCBI Glossary]. In some cases, the role of a gene in a certain disease may be extensively studied in one of the gene’s homologs, but not the gene itself. Retrieving those documents about the homolog could be very useful for the researcher to understand the function of the gene.

Acknowledgement

We gratefully acknowledge Dr. David Featherstone for his helpful discussion. We thank Jie Hong for analyzing the search results of biomedical articles.

Reference

[1] Porter stemmer. Porter MF. An algorithm for suffix stripping. http://www.tartarus.org/~martin/PorterStemmer/Program. 1980;14:130¨C137.

[2] Fukuda K, Tamura A, Tsunoda T, Takagi T. Toward information extraction: identifying protein names from biological papers. *Pac Symp Biocomput*. 1998;:707-18.

[3] Detecting Gene Symbols and Names in Biological Texts: A First Step toward Pertinent Information Extraction. *Genome Inform Ser Workshop Genome Inform*. 1998;9:72-80.

[4] Buttcher S, Clarke CLA, Cormack GV: Domain-specific synonym expansion and validation for biomedical information retrieval (MultiText experiments for TREC 2004). *The Thirteenth Text REtrieval Conference (TREC 2004) Proceedings*, 2004, Gaithersburg, MD.

[5] Huang X, Zhong M, Si L. York University at TREC 2005: Genomics Track. *The Fourteenth Text REtrieval Conference (TREC 2005) Proceedings*, 2005, Gaithersburg, MD.

[6] Guidelines for Human Gene Nomenclature. http://www.gene.ucl.ac.uk/nomenclature/guidelines.html

[7] Zhou W, Torvik VI, Smalheiser NR. ADAM: Another Database of Abbreviations in MEDLINE. *Bioinformatics*. In press. 2006.

[8] McCray AT, Nelson SJ. (1995) The Representation of Meaning in the UMLS. *Methods Inf Med*. 1995 Mar;34(1-2):193-201.

[9] The protocol of the genomic track of TREC 2006. http://ir.ohsu.edu/genomics/2006protocol.html
[10] PubMed Automatic Term Mapping. http://www.ncbi.nlm.nih.gov/entrez/query/static/help/pmhelp.html#AutomaticTermMapping. 2006.
[11] Swanson,D.R., Smalheiser,N.R. (1997) An interactive system for finding complemen-tary literatures: a stimulus to scientific discovery. Artificial Intelligence, 91,183-203.
[12] Schuyler PL, Hole WT, Tuttle MS, Sherertz DD. The UMLS Metathesaurus: representing different views of biomedical concepts. *Bull Med Libr Assoc*. 1993 Apr;81(2):217-22.
[13] Church KW, Hanks P. Word association norms, mutual information and lexicography. *Computational Linguistics*. 1990;16:22, C29.
[14] Robertson SE, Walker S. Okapi/Keenbow at TREC-8. *NIST Special Publication 500-246:The Eighth Text REtrieval Conference (TREC 8)*.
[15] Liu S, Liu F, Yu C, and Meng WY. An Effective Approach to Document Retrieval via Utilizing WordNet and Recognizing Phrases. *Proceedings of the 27th Annual International ACM SIGIR Conference*, pp.266-272, Sheffield, UK, July 2004.
[16] R. Baeza-Yates, B. Ribeiro-Neto. Modern Information Retrieval.. Addison-Wesley, 1999, 129-131.
[17] The Stanford Abbreviation Server. http://abbreviation.stanford.edu/.