Biomedical Text Retrieval System at Korea University

Young-In Song, Kyoung-Soo Han, Hee-Cheol Seo,
Sang-Bum Kim, Hae-Chang Rim

Natural Language Processing Lab., Dept. of CSE,
Korea University, Anam-dong 5-ga, Seongbuk-gu, 136-701, Seoul, Korea
{sprabbit, kshan, hcseo, sbkim, rim}@nlp.korea.ac.kr

Abstract. In this paper, we describe our retrieval system used for the primary task of genomics track at this year. Our primary goal in this task is to find a proper method for the domain-specific retrieval environment. To achieve the goal, we have tested several techniques such as a phrase indexing strategy, two query weighting methods, and two post-processing methods such as a document filtering method and a documents reranking method. According to the experimental results, query weighting methods and document filtering methods can improve the performance of the retrieval system, but there still remain a room for improvement.

1 INTRODUCTION

The primary task of Genomics track is a kind of conventional ad-hoc retrieval task, where the system is expected to retrieve relevant documents in response to a user's query. However, this task has some significant differences to previous ad-hoc tasks, because of its environment. Documents and queries in this task are limited to the biomedical domain.

The document collection used in this task consists of about 520,000 MEDLINE abstracts, which is a database of biomedical literature. Compared to a general news-wire document collection, it has a number of distinguished features such as frequent usage of spelling variants, long length of multi-word terms, and somewhat different lexical phenomena.

Query set in this task is also different in some respects. A typical query in traditional retrieval tasks almost consists of natural language sentences which are weakly structured using a tag such as <desc>, <title> or not structured, and there isn't any restriction about the query. However, the query in this task consists of not sentences but only several terms, formalized as a kind of table structure, and the user information need is limited to find documents relevant to 'basic biology' of gene X[Hersh 2003].

Our primary goal of this experiment, thus, is to explore methods and strategies which can reflect these differences of query and documents to improve retrieval performance of IR system, and especially we focus on following three issues:

1) Keyword extraction strategy for multi-word term.

2) Query weighting methods considering term variants and multi-word terms.

3) Post processing technique such as document reranking and filtering to satisfy restrictions of the structured query.

2. PRELIMANRY EXPERIMENT

In our preliminary experiment with the training data, we have tested basic techniques of information retrieval related to keyword extraction such as stemming, and we found some interesting points.

Table 1 and 2 show the results of our preliminary experiments.

	Avg Precision	R-Precision	Rel-ret
No Stemming	0.2274	0.2056	207
Porter	0.1944	0.1842	252
Lovins	0.2693	0.2408	265

Table 1. Experiments results according to various stemming methods at training data.

	Avg Precision	R-Precision	Rel-ret
Base line	0.2887	0.2680	271
Simple rule	0.3342	0.3112	274

Table 2. Experiments results of keyword extraction with /without simple rule

The retrieval performance of the Porter stemmer, which is one of the most widely used stemmer in retrieval systems, is much worse than the Lovins stemmer, and even worse than the case when any stemmer is not used. That result is contrary to the previous researches which reported that the Porter stemmer yields a similar or better performance than other stemmer including no stemming [Fuller 1998, Namba 2000].

In addition, we tested a simple key word extraction method for a word consisting of two numeric characters or one alphabet letter. They frequently occur in biomedical terms such as gene names, and sometimes they cause to fail in retrieving relevant documents. The simple heuristic rule is described as follows:

> **Simple rule:** if a word w_1 is a short length word and the adjacent word w_2 is not a short length word, w_1 and w_2 words are combined into a keyword as a canonical form.
> In this case, the adjacent word w_2 also is extracted as a keyword, too.
> **E.g.** [G protein, protein G -> protein, G:protein]

The result of adapting the simple rule to retrieval is shown in table 2. That simple method achieves about 15% improvement over the baseline.

3. INDEXING

Based on the observation of the preliminary experiments, keywords were extracted using the Lovins stemmer, and simple rules with case insensitive manner. Our system also did a stopword removal using a stop word list of PubMed [NCBI 2003].

Additionally, a phrase indexing strategy was also used to handle a multiword biomedical term.

3.1 The phrase indexing strategy using term boundary detection

The query and documents in this task have a lot of biomedical terms including multi-word terms, which often prevent a retrieval system from matching between a query and documents, so we tried to index phrases by identifying term boundaries.

Any keyword pair of adjacent non stopwords in order within a term boundary is regarded as a phrase. If a term consists of one word, it is also regarded as a phrase itself. To detect term boundaries in a document, we used a named entity tagger for biomedical domain [lee 2003]. Phrases are weighted with the same scheme as single terms.

4. DOCUMENT RETRIEVAL

In this section, we will describe the basic model of our retrieval system and two query weighting methods.

4.1 Basic retrieval model

All models used in our system are based on the probabilistic model with the BM25 weighting scheme of the Okapi system [Robertson 2000].

Equation (1) is the weighting formula of our basic model. We slightly modified K factor of the weighting function BM25.

$$\sum_{W\in Q} w^{(1)} \times \frac{(k_1+1)tf}{K+tf} \times QW \quad (1)$$

$$w^{(1)} = \log(\frac{N-n+0.5}{n+0.5})$$

$$K = (k_1 + \log(\frac{n}{totallikelihood}) - 4) \times ((1-b) + b(\frac{1+\log(dl)}{1+\log(avdl)})) \quad (2)$$

$$QW = \frac{(k_3+1)qtf}{k_3+qtf} \quad (3)$$

Where

Q is a query, containing word W,
N is the number of documents,
n is the number of documents containing the keyword,
$w^{(1)}$ is the Robertson / Sparck Jones weight[Robertson, et al 1976],
$k_{1,}$ b, $k3$ is the parameters which depend on the nature of queries and document collection. We fixed $k_1 = 1.5$, $b=0.6$, $k3= 1$ experimentally,
tf is the frequency of occurrence of the keyword within a document,
qtf is the frequency of the keyword within query,
dl and $avdl$ are the document length and average document length.

4.2 Query weighting method

We have proposed two query weighting techniques for the genomics-track style queries: normalizing query weight and incorporating inverse query frequency.

4.2.1 Query weight normalization

Genomics-track style query consists of a number of subqueries including an official gene name, its official symbol and aliases, etc. Most of them are equally important to retrieve the relevant documents effectively. However, with the basic model of Equation (1), one critical problem can occur because of the long subqueries. For example, we can have two relevant documents: One contains a long official gene name “cyclin-dependent kinase inhibitor 1A (p21, Cip1)” and the other contains its official symbol “CDKN1A”. In this case, it is obvious that two documents are equally relevant. With the base Okapi model, however, the former document appears at a higher rank since many term weights are added to the score of the former document.

One possible solution to alleviate this problem is ***query weight normalization*** according to the length of the subqueries. To do this, we modified the QW factor defined in Equation (3) as follows:

$$QW^1 = \sum_{q=1}^{|Q|} \frac{(qk+1)qtf}{qk((1-qb)+qb(ql))+qtf} \quad (4)$$

Where

$|Q|$ is the number of subquery within query Q,
qtf is the frequency of the keyword within subquery,
ql is the subquery length, and qk and qb is the parameters which depend on the documents and queries. We fixed $qk = 1.2$, $qb = 0.95$ experimentally.

We define the equation (4), QW^1, with a similar manner to document term weighting scheme of Okapi. The ql factor in equation (4) has an effect to balance weight of different length subqueries in a query.

4.2.2 Inverse query frequency

Each word forming a gene name can have different discriminative power. For example, while some words such as 'inhibitor', 'receptor', and 'kinase' occur within the various gene names, words such as 'p21', 'Cip1' occur only in some specific gene names. In other words, if 'Cip1' and 'receptor' occur in the same query, 'Cip1' is more useful query term than the common word 'receptor'.

Based on this observation, we define a new weight factor, inverse query frequency: the number of every possible query divided by the number of queries containing the specific term. For this task, we regard a set of every possible query as 15,000 gene names list obtained from the various web sites because only the gene names are assumed to be entered into our system.

Thus, the new query weight formula adopting inverse query frequency, QW^2, is represented by:

$$QW^2 = QW^1 \times \frac{QN+0.5}{qn+0.5} \quad (5)$$

Where

QN is the size of gene names list.
qn is the number of queries in the query set containing the keywords

We used equation (5) for submitted runs instead of the equation (3).

5. Reranking and Filtering

In the genomics track, two constraints must be satisfied. First, each retrieved document must be about 'basic biology' of the gene in a query or its protein product. Second, the gene in a document must be from the organism designated in the query. We reranked and filtered the initial retrieved documents to improve the performance of the system. The details are described in the following subsections.

5.1 Reranking using event verbs

We reranked documents using event verbs to increase the score of the documents about "basic biology". The event verb here means a verb widely used to represent interactions among the genes or proteins. We assume that documents containing many event verbs are likely to be about basic biology.

According to this assumption, we reranked documents by using the following new score function:

$$new\ score = inital\ weight + Additional\ weight \qquad \mathbf{(6)}$$

$$Additional\ weight = \sum_{v=1}^{|V|} w^{(1)} \times \frac{(k_1+1)tf}{K+tf} \times \alpha$$

Where
initial weight is the weight between the query and the document, which is calculated at the initial retrieval,
v is the event verb and $|V|$ is the vocabulary size of a event verb list,
$w^{(1)}$,k_1, K, and tf are the same symbol used for equation (1).
α is the parameter depending on the reliability of reranking. We fixed it as 0.2.

The event verb list used in experiments consists of 182 verbs which is chosen by biologists for information extraction [Chun 2003].

5.2 Document filtering using MeSH

Unfortunately, many retrieved documents with the given query may have a lot of irrelevant documents, which focus on the basic biology of the query gene, but from another species.
To filter only the documents about genes from the species designated in given query, we used a simple heuristic using MeSH field in each document [NLM 2003] provided that the query gene from only the four species is given. The heuristic is as follows:

"If a document doesn't have a representative MeSH keyword for the species in the query, but has one of the representative keywords for other three species, remove the document from the list"

We choose four representative keywords for each species: 'human' for the human, 'rats' for the rat, 'mice' for the mice, 'drosophila' for the fruit fly.

6. EXPERIMENT AND EVALUATION

We have submitted two runs for the primary task of genomics track this year. The first run, KUBIO IRRAW, make use of simple rules for keyword extraction, query weighting using length normalization, and inverse query frequency, QW^2, reranking, and document filtering. The second run, KUBIOIRNE, uses one more strategy, phrase indexing method using a term boundary identification. Both of runs performed at or above the median in almost all queries, shown in table 3.

The results of table 4 show that there is little advantage of using phrase indexing strategy for keyword extraction. KUBIOIRNE shows a better performance than KUBIOIRRAW at all evaluation measures, but considering its cost, improvement is tiny.

	Avg Precision				Rel At 10 doc				Rel At 20 doc			
	Best	> Mid	=Mid	< Mid	Best	> Mid	=Mid	< Mid	Best	> Mid	=Mid	< Mid
KUBIOIRRAW	2	42	0	8	4	23	25	2	5	25	21	4
KUBIOIRNE	1	40	1	9	5	24	25	1	3	24	24	2

Table 3. Comparative results

	Avg Precision	R-Precision	Rel-ret	At 10 doc	At 20 doc
KUBIOIRRAW	0.2937	0.2696	541	0.2240	0.1690
KUBIOIRNE	0.2980	0.2837	532	0.2320	0.1710

Table 4. Retrieval results of submitted runs.

	Test Topics		Training Topics	
	Average Precision	Improvement over Baseline	Average Precision	Improvement over Baseline
Baseline	0.1619	+0.00%	0.3342	+0.00%
+ Phrase	0.1649	+1.85%	0.3197	-4.34%
QW^1	0.2011	+24.21%	0.3628	+8.56%
QW^2	0.2100	+29.71%	0.3797	+13.61%
+ Reranking	0.2121	+31.01%	0.3800	+13.70%
+ Filtering	0.2980	+84.06%	0.4201	+25.70%

Table 5. Retrieval performance at each step. Baseline represents the base model for retrieval including simple rules for keyword extractionis.

What is worse, performance of the phrase strategy with the basic model is lower than the baseline as shown in table 5.

Two possible reasons are as follows. One is the risk of a high inverse document frequency of phrase. Especially, some unsuitable phrases with abnormal high idf cause a trouble. Another reason is that when phrase strategy is used, long length terms of the query are more strongly favored. This tendency is proved indirectly in table 5. Improvement by the query weighting using length normalization, QW^1, is much bigger with the phrase indexing than without the phrase indexing.

Table 5 shows the relative improvement of the retrieval performance according to the additional techniques. Almost all our proposed methods for this task yield better results but one negative case, which use the base model with phrase indexing. Relatively, the phrase indexing and the document reranking method produce rather disappointing results, and query weighting methods and document filtering performed well.

The results of query weighting methods, QW^1 and QW^2, are fairly good as shown in table 5, table 6, and table 7. They achieved 17-29% improvement at test and training queries with any indexing methods.

The document reranking method makes just a little improvement. It achieved merely about 1% increase of average precision as shown in table 5, and table 7. We guess the reason is that the value of the parameter αused as 0.2 in experiments is too small to change a document rank, or our method for reranking documents was too heuristic.

The document reranking method makes just a little improvement. It achieved merely about 1% increase of average precision as shown in table 5, and table 7.

	Test (No filtering / Filtering)						**Training (No filtering / Filtering)**					
	No Phrase		Simple Rule		Phrase		No Phrase		Simple Rule		Phrase	
Base model	.1600	.2443	.1619	.2507	.1649	.2528	.2999	.3467	.3342	.3848	.3197	.3651
QW^1			.1822	.2691	.2011	.2843			.3496	.4086	.3628	.4070
QW^2			***.1966***	***.2929***	***.2100***	***.2957***			***.3586***	***.4164***	***.3797***	***.4195***

Table 6. Average Precisoin according to each methods. Bold is the best score.

	Test (No filtering / Filtering)						**Training (No filtering / Filtering)**					
	No Phrase		Simple Rule		Phrase		No Phrase		Simple Rule		Phrase	
Base model	.1312	.2188	.1399	.2200	.1416	.2306	.2773	.3073	.3112	.3569	.2734	.3350
QW^1			.1446	.2275	***.1647***	.2590			.3191	.3723	.3299	***.3767***
QW^2			***.1791***	***.2602***	.1639	***.2680***			***.3303***	***.3926***	***.3414***	.3719

Table 7. Recall-precision according to each methods. Bold is the best score.

	Test				**Training**			
	Average Precision		R-Precision		Average Precision		R-Precision	
	No filter	Filter	No filter	Filter	No filter	Filter	No filter	Filter
before Reranking	0.2100	0.2957	0.1639	0.2680	0.3797	0.4195	0.3414	0.3719
After Reranking	0.2121	0.2980	0.1709	0.2837	0.3800	0.4201	0.3396	0.3701

Table 8. Comparision between before and after reranking.

	Test		**Training**	
	Average Precision	R-Precision	Average Precision	R-Precision
Before filtering	0.2121	0.1709	0.3800	0.3396
After filtering	0.2980	0.2837	0.4201	0.3701

Table 9. Comparison between before and after filtering.

In spite of its simplicity, document filtering achieves the biggest improvement as shown in table 5, and table 9. It means that there are so many documents which are relevant but describe another species. In this task, satisfying the species constraint in a query seems to be important.

7. CONCLUSIONS

We have tried heuristic strategies which can reflect characteristics of this task. The strategies can be classified into three classes.

First one is the indexing strategy. To handle a lot of multi-word terms in biomedical literature, we have tried the phrase indexing method based on term boundary information. Named entity tagger was used for it, but it yields a rather disappointing result. We will have to devise a good phrase extraction method and a reliable phrase weighting scheme. It will be one of our future works.

Second one is the query weighting scheme. The query in this task is quite different from the other ad hoc task. We have developed two heuristic query weighting methods which can reflect the characteristics of the query, and the domain information, and they can increase performance of our system successfully.

Finally, we have used two post-processing methods. Our simple document filtering method works very well, but more analysis is required for documents reranking.

References

1. W. Hersh. http://medir.ohsu.edu/~genomics/
2. M. Fuller and J. Zobel. Conflation-based Comparison of Stemming Algorithms, In *Proceedings of the Third Australian Document Computing Symposium Sydney*, Australia, 21st August, 1998.
3. I. Namba, N Igata, H Horai, K Nittta, and K Matsui. *The Eight Text REtrieval Conference*, 2000.
4. K.J. Lee, Y.S Hwang, H.C. Rim. Two-Phase Biomedical NE Recognition based on SVMs, In *ACL'03 nlbio workshop,* 2003.
5. NCBI. http://www.ncbi.nlm.nih.gov/PubMed/
6. S.E. Robertson, S Walker. Okapi/Keenbow at TREC-8, In *The Eighth Text REtrieval Conference*, 2000.
7. S.E Robertson, and K Sparck Jones. Relevance weighting of search terms. *Jounal of the American Society for information Science 27*, May-June 1976, p129-146.
8. H.W.Chun, Y.S. Hwang, H.C.Rim. Unsupervised Event Extraction from Biomedical Literature using Co-occurrence Information and Basic Pattern, The *2nd annual conference of the Korean Society for Bioinformatics*, (To be appear), 2003.
9. NLM, http://www.nlm.nih.gov/mesh/