

Retrieval of Biomedical Documents by Prioritizing Key Phrases

Kevin Hsin-Yih Lin, Wen-Juan Hou and Hsin-Hsi Chen

*Department of Computer Science and Information Engineering,
National Taiwan University
Taipei, Taiwan, 106*

E-mail: {hylin, wjhou }@nlg.csie.ntu.edu.tw; hhchen@csie.ntu.edu.tw

Abstract

In this paper, we presented an approach to retrieving relevant articles from the biomedical corpus. Our first run considered four kinds of operators as query expansion. The operators are phrase, mandatory, optional and synonym set. The second run lowered the ranking of documents which contained query terms only in their MeSH fields. The results of the official runs were listed.

1. Introduction

For the retrieval task of the Genomics Track, we implemented an information retrieval system which supports phrase searching and BM25 scoring. Our system first extracted key terms from topic narratives by pattern matching. The Entrez Gene database and MeSH database were used for query expansion. We did not rank documents directly by BM25 score. Instead, we devised a method to identify more important phrases in a query and gave those terms higher priority over the less important phrases, regardless of their BM25 scores. We indexed only the TI, AB, MH, RN and GS fields of the Medline entries. The stop word list of PubMed was used to filter out stop words. We also used the Porter stemmer to stem all words.

The rest of this paper about genomics track on ad hoc task is organized as follows. Section 2 presents the overview of our system. The methods in our system are explained in details in Section 3. Finally, the results of our official runs are shown in Section 4.

2. System Description

Our retrieval system is an extension of the BM25 document scoring scheme. The BM25 formula is defined as

$$\text{BM25}(q, d) = \sum_{t \in Q} q_t \times \ln \left(\frac{D - df_t + 0.5}{df_t + 0.5} \right) \times \frac{\text{freq}_{t,d} \times (1 + k_1)}{\text{freq}_{t,d} + k_1 \left((1 - b) + b \times \frac{dl_d}{\text{avdl}} \right)}$$

where $\text{BM25}(q, d)$ is the BM25 score of document d for the query q , q_t is the frequency of the term t in q , D is the total number of documents, df_t is the document frequency of t , $\text{freq}_{t,d}$ is the term frequency of t in d , k_1 is a parameter, b is a parameter, dl_d is the document length of d and avdl is the average document length of all the documents [4]. We used the same k_1 and b values as those used by Büttcher *et al.* in TREC 2004 Genomic Tracks [3]. The parameters were set to be $k_1 = 1.2$ and $b = 0.75$.

In our retrieval system, we added phrase, mandatory, optional and synonym set operators. We evaluated an expanded query by creating different permutations of a query out of the query term synonyms, instead of adding the synonyms to the original query. To evaluate a query, our system went through three iterations, each time relaxing some constraints of the query.

2.1 Operators

We used four kinds of operators: phrase, mandatory, optional and synonym set. The optional, mandatory and phrase operators contained one or more query terms. The synonym set operator contained one or more of the three other operators.

In order for a document to satisfy a phrase operator, the exact phrase contained in the phrase operator must appear in the document. If a document contained such a phrase, the weight of the phrase was computed by summing up the BM25 score of each term in the phrase. If a query contained a phrase operator, then a document which did not satisfy the phrase operator was considered as irrelevant regardless of what the rest of query was.

The mandatory operator behaved very much like the phrase operator, except that terms inside a mandatory operator did not have to occur next to each other or in any particular order in a document. A document satisfied a mandatory operator if the document contained all the terms inside the operator. The weight of a mandatory operator was the sum of the BM25 scores of all the terms inside the operator. Again, if a query contained a mandatory operator, then a document which did not satisfy the mandatory operator was seen as irrelevant regardless of what the rest of query was.

The optional operator was the least restrictive operator. For each document, the weight of an optional operator was the sum of the BM25 scores of all the terms that were both inside the operator and appeared in the document. The final score of a document was the sum of BM25 scores of all the operators.

2.2 Query Expansion

The synonym set operator was used to expand a query. We did not put all synonyms into a single query. Instead, we constructed different permutations of a query, each containing a different member of the synonym set. If a query contained more than one synonym set, then the number of queries generated was the product of the sizes of the synonym sets. The relevance score of a document was its maximum relevance score over all the query permutations.

2.3 Three Iterations

Since the phrase and mandatory operators were highly restrictive, the number of documents retrieved could be very small when a query contained too many phrase and mandatory operators. To increase the number of documents retrieved, we evaluated a query in three iterations, each time reducing some constraints of the query. The first iteration used the original set of expanded query permutations. In the second iteration, we converted phrase operators to mandatory operators. In the last iteration, we converted all operators to optional operators.

During query evaluation, the documents proposed (i.e., documents with BM25 score greater than 0) in the preceding iteration always have higher ranks than the documents proposed by the succeeding iteration, regardless of the relevance weight of the documents. Within each iteration, documents are ranked by the BM25 scores.

3. Query Formulation

To formulate a query that can be accepted by the retrieval model in Section 2, we took the topic narratives and went through the steps of template type identification, gene name synonym expansion, query phrase identification and query phrase synonym expansion.

3.1 Template Type Identification

Each topic narrative follows a specific format depending on its corresponding template. We used pattern matching to convert the templates back to their template form. The templates have the same format as the ones that are defined in the task description [5].

3.2 Gene Name Synonym Expansion

We used the *gene_info* file provided by the Entrez Gene database to find gene synonyms [2]. The Symbol, LocusTag, Synonyms, Symbol From Nomenclature Authority, and Full Name From Nomenclature Authority fields of the *gene_info* file were used to form synonym sets. From the *gene_info* file, we removed gene names which are synonyms of more than one gene names. This was done to avoid ambiguity. We did not use all possible synonyms in order to fasten the search time. To select the best synonyms, we ranked the synonyms by the following measure:

$$Score(gene_name, synonym) = \frac{LCS_Length(gene_name, synonym)}{Length(gene_name) + Length(synonym)}$$

where *gene_name* is the original gene name, *synonym* is the synonym of *gene_name*, *Length* is a function that returns the length of a string, *LCS_Length* is a function that returns the length of the longest common subsequence of two strings. For each gene name, the two synonyms with the highest scores were chosen as the synonyms for query expansion. In effect, we wanted to find short synonyms that resembled the original gene name.

If a gene name did not appear in the *gene_info* file and hence did not have readily available synonyms, we applied the number heuristics to create possible variants of the gene name. In the number heuristics, we converted Roman numerals from one to ten into Arabic numerals, and vice versa. We also added or deleted spaces between a number and the preceding term. To form the query, we used phrase operators on a gene name and its synonyms. We then used the synonym set operator on the phrase operators containing the gene names.

3.3 Query Phrase Recognition and Expansion

Query phrases were the important non-gene phrases in this study. A query phrase was found by finding the longest substring in the topic template which matched a phrase in the PRINT ENTRY or ENTRY fields of the MeSH descriptor file d2005.bin [1]. Once a query phrase has been found, the phrases in the PRINT ENTRY and ENTRY fields of the same MeSH term were used as the synonyms of query phrase. To form a query, we used the mandatory operators on the query phrases and their synonyms.

4. Experimental Results

We submitted two runs for evaluation. The first run (NTUgah1) employed the methods described above. For the second run (NTUgah2), we lowered the ranking of documents which contained query terms only in their MeSH fields. The performance results are listed in Table 1.

Table 1: Overall Performance of the Two Runs

Run	P10	P100	MAP
NTUgah1	0.3918	0.1998	0.2173
NTUgah2	0.3980	0.1996	0.2204

5. Concluding Remarks

In this paper, we demonstrated how our system was constructed. The TI, AB, MH, RN and GS fields of the Medline entries were considered as a representation of the article. The Entrez Gene database and MeSH database were used for query expansion. Four kinds of operators, i.e., phrase, mandatory, optional and synonym set, were used in this study. We submitted two runs where the first run used the total fields of the Medline entries, while the second run lowered the ranking of documents which contained query terms only in their MeSH fields.

REFERENCES

- [1] Medical Subject Headings. <http://www.nlm.nih.gov/mesh/meshhome.html>, 2005.
- [2] NCBI Entrez Gene. <http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?CMD=search&DB=gene>, 2005.
- [3] Stefan Büttcher, Charles L.A. Clarke, and Gordon V. Cormack. Domain-Specific Synonym Expansion and Validation for Biomedical Information Retrieval (MultiText Experiments for TREC 2004). In *Proceedings of Text REtrieval Conference 2004 (TREC 2004)*, November 2004.
- [4] Stephen Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. Okapi at TREC-3. In *Proceedings of the Third Text REtrieval Conference (TREC-3)*, November 1994.
- [5] TREC 2005 Genomics Track Protocol. <http://ir.ohsu.edu/genomics/2005protocol.html>, 2005.