York University at TREC 2007: Genomics Track

Xiangji Huang[1], Damon Sotoudeh-Hosseini[2], Hashmat Rohian[2] and Xiangdong An[2]
[1]School of Information Technology, York University, Toronto, Ontario, Canada
e-mail: jhuang@yorku.ca
[2]Department of Computer Science & Engineering, York University, Toronto, Ontario, Canada
e-mail: damon,hrohian,xan@cs.yorku.ca

Abstract

Our Genomics experiments in this year mainly focus on improving the passage retrieval performance in the biomedical domain. We address this problem by constructing different indexes. In particular, we propose a method to build word-based index and sentence-based index for our experiments. The passage mean average precision (passage MAP) for our first run "york07ga1" using the word-based index was 0.095 and the passage MAP for our second run "york07ga2" using the sentence-based index was 0.086. However, the passage MAP for our third run "york07ga3" using both the word-based index and UMLS for query expansion degraded to 0.060. All these three official runs are automatic. The evaluation results show that using the word-based index is more effective than using the sentence-based index for improving the passage retrieval performance. We find that pseudo-relevance feedback can make a positive contribution to the retrieval performance. However, we also find that query expansion using UMLS and Entrez Gene does not improve the retrieval performance, and in some cases it makes a negative contribution to the retrieval performance.

1 Introduction and Motivations

This paper describes the work done by members of York University for the TREC 2007 Genomics track. Our goal of participating in TREC Genomics track this year is to evaluate the Okapi system at document-level, passage-level and aspect-level retrieval in the biomedical domain. In this year, we focus on evaluating passage-level retrieval.

In our 2004's Genomics experiments, we did not incorporate domain expertise and did not use external biomedical resources [4]. Our experiments on Genomics experiments in 2005 focused on the following methodologies: (1) We design two new algorithms for biomedical query expansion. (2) We build structured queries based on extended query terms. (3) We use external biomedical resources for further synonym expansion on the manual run. (4) We use an extended stop word set for improving the retrieval performance. In 2005, we mainly focused on addressing three major problems in biomedical information retrieval [5]. Last year our experiments mainly focused on using the automatic query expansion algorithm to construct structured queries for document-level retrieval and applying several data mining techniques for passage-level retrieval and aspect-level retrieval. However, our retrieval performance last year on passage-level and aspect-level was poor since we built a wrong index for the passage-level retrieval [7]. This year we focus on improving the passage retrieval performance by proposing a new method to build indexes.

The test corpus used in this year's Genomics experiments consists of 162,259 full documents in HMTL format from 49 journals with a total size of 12.3GB. The HTML filename is named as the document's PMID followed by the .html extension. There are 50 topics in total this year. Those topics are available at the address "http://ir.ohsu.edu/genomics/data/2007/2007topics.txt".

The rest of the paper is organized as follows. We first give a brief description of the Okapi information retrieval system in Section 2. Then, our proposed ideas and methods are presented in Section 3. Following that, experimental results are provided in Section 4. Finally, the conclusion and future work are given in Section 5.

2 Weighting and Ranking

We used Okapi BSS (Basic Search System) as our main search system. Okapi is an information retrieval system based on the probability model of Robertson and Sparck Jones [9]. The retrieval documents are ranked in the order of their probabilities of relevance to the query. Search term is assigned weight based on its within-document term frequency and query term frequency. The weighting function used is BM25 [2].

$$w = \frac{(k_1+1)*tf}{K+tf} * \log \frac{(r+0.5)/(R-r+0.5)}{(n-r+0.5)/(N-n-R+r+0.5)} * \frac{(k_3+1)*qtf}{k_3+qtf} \quad \oplus \quad k_2 * nq * \frac{(avdl-dl)}{(avdl+dl)} \tag{1}$$

where N is the number of indexed documents in the collection, n is the number of documents containing a specific term, R is the number of documents known to be relevant to a specific topic, r is the number of relevant documents containing the term, tf is within-document term frequency, qtf is within-query term frequency, dl is the length of the document, $avdl$ is the average document length, nq is the number of query terms, the k_is are tuning constants (which depend on the database and possibly on the nature of the queries and are empirically determined), K equals to $k_1 * ((1-b) + b * dl/avdl)$, and $\oplus$ indicates that its following component is added only once per document, rather than for each term.

In our experiments, the values of k_1, k_2, k_3 and b in the BM25 function are set to be 1.4, 0, 8 and 0.55 respectively. Our system also supports the structured queries for searching. That is: several different terms that are connected by '+' sign can be used to represent the same concept. For example, "COPII+COP2" stands for "COPII" and "COP2" are synonym.

3 Word and Sentence Based Indexing

One important issue that information retrieval systems have to deal with is the size of the retrieved passages. In the context of text retrieval, size can be defined as the length of the retrieved passage. A very short text may not contain enough information and a long text may contain either unnecessary or redundant information. It is not trivial to decide on the size that would provide the best result. We address this problem by constructing indexes that take the length of retrieved passages into account. Two kinds of indexes were built in this year's experiments, which will be described in detail next.

3.1 Sentence Based Indexing

Since a passage is essentially a sequence of sentences, we initially measured the length of passages as the number of sentences they contain. We then analyzed the passages of the gold standard of 2006 TREC, and found that, on average, the length of these passages is 3. We broke each paragraph into passages of three sentences with no overlap; i.e. the first three sentences in the paragraph form the first passage, the next three sentences form the second passage, and so on. We built an index that mapped a keyword to the three-sentence passages that contained it.

3.2 Word Based Indexing

The problem with Sentence Based Indexing is that sentences can be too short or too long. For example, a passage of three long sentences may contain redundant information. To overcome this

problem, we analyzed the length of the passages specified by the gold standard in terms of the number of words they contain. On average, there are 47 words per passage. We broke each paragraph into its sentences, and formed passages by adding one sentence at a time until either we had three sentences in the passage or the number of words in the passage exceeded 47. This leads to formation of passages that contain one, two or three sentences each, where the number of words in the passages may only slightly exceed 47.

Another difference between the word and sentence based indices is that for word based index we allow for maximum overlap of sentences. For a paragraph of five sentences, sentence based index has two passages only; one containing first, second, and third sentences, the other containing fourth and fifth sentences. The word based index however has three passages; first, second, and third sentences form one passage, second, third and fourth sentences form another passage, and third, fourth, and fifth sentences form the last passage (assuming the first two sentences of each passage do not exceed 47 word limit). This approach creates redundancy in the passages but it has the advantage that we can look at all the possible passages. Word based indexing is described in Figure 1. The algorithm takes a paragraph as input and outputs its passages. A paragraph here is defined as the sequence of sentences between the and tags from the HTML data set. A passage is a sequence of sentences inside that paragraph with length of at most 3 (in terms of the number of sentences). As a result, a paragraph may contain a single passage, or multiple overlapping passages depending on the number of sentences it has, and the number of words in each sentence.

Input: a paragraph containing a number of sentences
Output: a list of passages obtained from the paragraph

Method:
(1) Break the paragraph into a list of its sentences
(2) Until the list of sentences becomes empty
  (3) create an empty passage
  (4) until the passage has 3 sentences or 47 words
    (5) Read the next sentence and add it to passage
  (6) remove first sentence of the list
  (7) remove second sentence of the list only if three sentences were added
  (8) start reading from the beginning of the list

Figure 1: The algorithm for word based index

4 Experiments

Our experiments were conducted on a double-processor server which has 2 Intel Xeon 2.40GHz CPU and 2G memory. The version of Linux kernel we used is version 2.4.26. York University submitted three runs in total for the 2007 TREC Genomics track. All runs are automatic. The first run "york07ga1" used word based index with no topic expansion. The second run used sentence based index with topic expansion using Entrez Gene on the 11 gene-related topics. Our third run used word based index with topic expansion based on UMLS for all the topics. All the terms, including expanded terms obtained from Entrez Gene (in second run) and UMLS (in third run), are given a weight by Okapi (the main retrieval component). Our retrieval system multiplies the weight of expansion terms by multiplicative factor to ensure that expanded terms are given less importance during retrieval process. In all three runs, expansion weight is 0.4.

To improve the performance, feedback analysis was performed on all the three runs by our retrieval system. In feedback analysis, the topics are not examined directly; instead the retrieval system retrieves ten passages that are deemed most relevant for a particular topic, and forms a list

of the most recurring words from those passages. It is assumed frequently recurring words in top relevant passages of a topic should have some relevance to the topic. Each topic is expanded by those words, and then relevant passages for the extended topic is retrieved. Each feedback term is assigned a weight by Okapi (the main retrieval component). This weight is multiplied by a feedback weight to ensure that feedback terms are given less importance during retrieval. In our experiments, feedback weight was set to 0.25. In general, applying feedback analysis improved the performance of the retrieval system by a small margin.

The document mean average precision (MAP) of our first run "york07ga1" is 0.21534132, the passage MAP is 0.09465191, and the aspect MAP is 0.10169889. A summary of performance of all three runs is available in Table 1. Descriptive statistics for all the 66 official runs are given in Table 2.

Run	Description	Document MAP	Passage MAP	Aspect MAP
york07ga1	word based index, no topic expansion	**0.21534132**	**0.09465191**	0.10169889
york07ga2	sentence based index, gene expansion only	0.21502667	0.08587690	**0.13061169**
york07ga3	word based index, and topic expansion	0.19165312	0.05953786	0.06112168

Table 1: Official results at the 2007 Genomics track

Statistics	Document MAP	Passage MAP	Aspect MAP	Passage2 MAP
Minimum	0.0329	0.0029	0.0197	0.0008
Median	0.1827	0.0565	0.1311	0.0377
Maximum	0.3286	0.0976	0.2631	0.1148

Table 2: Minimum, median and maximum results at the 2007 Genomics track

We conducted many experiments to investigate the influence of topic expansion on the performance of retrieval system using various resources such as UMLS, and MedLine. However, none of the methods improved the performance of the retrieval system, and in some cases the performance even degraded. Changing the expansion weight also did not improve the results. We continue our investigation of finding a systematic method for topic expansion from biomedical resources.

We also implemented a mechanism to merge two continuous retrieved passages from the same paragraph into a single passage. Experiments showed this method is not effective in producing better results.

5 Conclusions

The contributions of our work are as follows. First, we propose two new methods to build indexes for improving the passage retrieval performance in the biomedical domain. Both word-based and sentence-based indexing methods are powerful to improve the passage retrieval performance. Second, using the word-based indexing method is more effective to improve passage retrieval performance than using the sentence-based indexing method. Third, using external resources such as UMLS and MedLine does not improve the retrieval performance, and in some cases the performance even degrades.

6 Acknowledgements

This research is supported in part by the research grant from the Natural Sciences and Engineering Research Council (NSERC) of Canada. We also would like to thank Peter Gorys and Miao Wen for their help.

References

[1] The Medstract Project: AcroMed 1.1. URL address: http://medstract.med.tufts.edu/acro1.1/

[2] M. Beaulieu, M. Gatford, X. Huang, S. E. Robertson, S. Walker and P. Williams (1996), Okapi at TREC-5. *Proceedings of 5th Text REtrieval Conference*, pp. 143-166, 1996.

[3] Stefan Buttcher, Charles L. A. Clarke, Gordon V. Cormack (2004), Domain-Specific Synonym Expansion and Validation for Biomedical Information Retrieval (MultiText Experiments for TREC 2004). *Proceedings of the 13th Text Retrieval Conference*, 2004.

[4] X. Huang, Y. R. Huang, M. Wen and M. Zhong (2004). York University at TREC-13: HARD and Genomics Tracks. *Proceedings of the 13th Text Retrieval Conference*, 2004.

[5] X. Huang, M. Zhong and S. Luo (2005). York University at TREC 2005: Genomics Track. *Proceedings of the 14th Text Retrieval Conference*, 2005.

[6] X. Huang, B. Hu and H. Rohian (2006). York University at TREC 2006: Genomics Track. *Proceedings of the 15th Text Retrieval Conference*, 2005.

[7] W. Zhou, C. Yu, V. Torvik, N. Smalheiser (2007). A Concept-based Framework for Passage Retrieval in Genomics. *Proceedings of the 15th Text REtrieval Conference*, 2006.

[8] LocusLink. URL address: http://www.ncbi.nih.gov/locuslink/ *National Center for Biotechnology Information*, 2005.

[9] S. E. Robertson, J. K. Sparck. Relevance Weighting of Search Terms. *Journal of the American Society for Information Science* 27, May-June 1976, p129-146