NTU at TREC 2006 Genomics Track

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 present a system for information retrieval of biomedical texts at passage level. Our system used KL-divergence as the underlying retrieval model. We further added query expansion and performed post-processing on the results. We were able to obtain a Document MAP of 0.3563, Passage MAP of 0.0464 and Aspect MAP of 0.2255 on one of the three runs.

1. Introduction

The Genomics Track this year only has a single task, which is information retrieval. Unlike the retrieval tasks of previous years, the task this year deals with the retrieval of passages from full-text documents rather than abstracts. The query format is based on last year's generic topic templates (GTT). In fact, this year's queries are generated from four of last year's templates. The use of GTT allows us to identify the occurrences of gene names, disease names, biological processes and organ functions more easily than freeform queries.

The corpus consists of full-text biomedical articles in HTML format. We extracted the content texts from the HTML files and separated them into paragraphs for indexing. Our system used the Lemur implementation of KL-divergence retrieval model as the main search algorithm [7]. Lemur is a toolkit aimed at making information retrieval research easier. It provides basic indexing functionalities and retrieval models, such as TF-IDF, Okapi and KL-divergence.

Several techniques independent of Lemur were applied in an attempt to increase the retrieval performance. Our system expanded queries prior to submitting them to Lemur. After the returning of possibly relevant paragraphs from Lemur, the paragraphs were further analyzed to locate the relevant passages within the paragraphs.

The rest of the paper is organized as follows. Section 2 sketches the overview of the system architecture. The details of the proposed system are explained in Section 3. Our

evaluation results are presented in Section 4. We also make some discussions in this section. Finally, we make some conclusions in Section 5.

2. System Overview

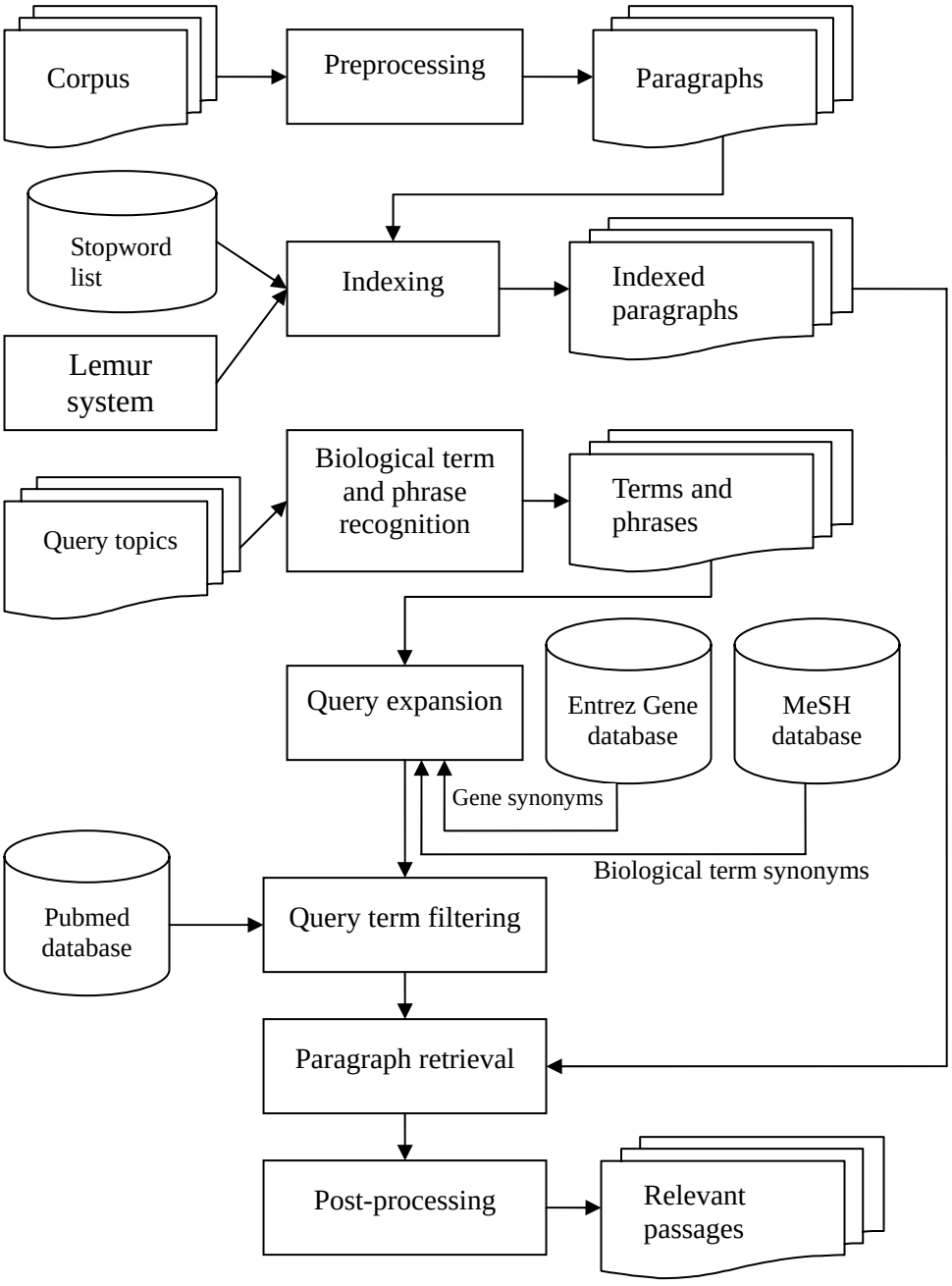

Figure 1: System architecture

Figure 1 shows the overall architecture of the proposed system. First, we prepared the corpus for indexing. Then, we identified biological terms and phrases in the query. After that, query expansion was made before we retrieved paragraphs from the indexed paragraphs. Finally, we post-processed the retrieval results.

3. Methods

3.1 Corpus Preprocessing and Indexing

Since the retrieval task this year requires the output to be passages, each of which is no longer than a paragraph, we separated every document in the corpus into paragraphs and indexed them. As the documents are in the HTML format, we defined a paragraph to be a portion of the HTML text bounded by the HTML tag <P> or a blank line.

The next step in corpus preprocessing was to convert the HTML paragraphs into human-readable text without the HTML tags. We did this by passing the HTML passages into Lynx, a text-based web browser, and output the formatted texts using the dump option [4].

We used Lemur to build an index suitable for doing searches by the KL-divergence method. For the stopwords, Pubmed's list was used [6]. We also used Porter's stemming algorithm to stem each word.

3.2. Biological Term and Phrase Recognition

Since the topics follow specific formats, we extracted gene names, disease names, biological processes and organ functions from the topics by simple pattern matching.

3.3 Query Expansion

Query expansion was done before retrieval to increase recall. Our source of synonymous names for genes came from the NCBI Entrez Gene database [1]. We downloaded the gene_info file from Entrez Gene and constructed sets of synonyms from the symbol, synonyms and description fields of the gene entries. As for expanding other biological terms, such as disease names, biological processes and organ functions, the MeSH database was used [5]. We used MH, PRINT ENTRY and ENTRY fields in the d2005.bin file to identify synonyms.

We did not use every single synonym we found to expand queries. Instead, only those synonyms that co-occurred at least once in Pubmed Medline abstracts with other

terms in the original query were selected [2]. We will use the query “What is the role of PrnP in mad cow disease?” as an example. For the query, three synonyms of "PrnP" are "Prn-p", "prp", and "prion protein". For each of the three synonyms, we checked whether it appeared together with mad cow disease in at least one Pubmed Medline abstract. If it did, we added it to the expanded query. A similar procedure was applied to the synonyms of mad cow disease: we checked whether they co-occur with "PrnP" or not.

3.4 Retrieval Model

We used Lemur to perform the retrieval of paragraphs using the KL-divergence model, which was introduced by Lafferty *et al*. in 2001 [3]. The basic idea behind model is to compute $p(d|q)$: the probability of a document d given the query q. We also used Lemur’s implementation of pseudo-relevance feedback. The number of feedback documents was set to 5.

3.5 Result Post-Processing

According to the task protocol, the output of the retrieval system has to be passages each no longer than a paragraph. Since we indexed the corpus by paragraphs, the output generated by Lemur was a list of paragraphs. We trimmed each paragraph returned by Lemur to filter out irrelevant parts of the paragraph surrounding the potentially relevant passage. To do this, each paragraph was first segmented into sentences. The first sentence and the last sentence in the paragraph which contained at least one term from the expanded query were identified. For the final answer passage, we kept, inclusively, only the sentences between these two sentences.

4. Results and Discussion

We submitted three runs to TREC for evaluation. The first run is NTUadh1, which was constructed using all the methods we described in Section 3 of this paper. The second run is NTUadh2, which is similar to NTUadh1 except that query expansions were not used. Our last run is NTUadh3, which used manually-edited queries. On the task protocol webpage, Nur-77 is considered to be a synonym of Nurr-77, which appears in topics 164 and 171. We added the term Nur-77 to these two topics manually. After adding Nur-77, the same methods that were applied to generate NTUadh1 were used to generate NTUadh3. The results for the three runs are given in Table 1.

Table 1: Results of the Runs

Run	Document MAP	Passage MAP	Aspect MAP
NTUadh1	0.3563	0.0464	0.2255
NTUadh2	0.3509	0.0429	0.2348
NTUadh3	0.3570	0.0463	0.2231

From Table 1, we see slight differences between the Document MAP of the three runs. NTUadh3 is better than NTUadh1 and NTUadh2. As we checked the Document MAP for topics 164 and 171, we noticed an increase in MAP for both of these topics when we included the term Nur-77. This explains why NTUadh3 has a higher Document MAP than NTUadh1. We also examined the difference in Document MAP between NTUadh1 and NTUadh2 for each of the topics to see whether query expansion was helpful for the majority of topics. We found that query expansion increased Document MAP for 9 topics, decreased Document MAP for 11 topics and did not affect the score of the rest of the topics. The absolute value of total increase in Document MAP for the 9 topics is higher than absolute value of the total decrease in Document MAP for the 11 topics, so the overall score of NTUadh1 is higher than NTUadh2. In reality, our query expansion method was harmful to more topics than it was helpful with.

For Passage MAP, there is almost no difference between the scores of NTUadh1 and NTUadh3. This is not surprising, since the two runs only differ from two of the topics. NTUadh1 has a slightly higher score than NTUadh2. As we had done for the comparison of Document MAP, we checked the Passage MAP for each of the topics. We discovered that query expansion increased Passage MAP for 11 topics and decreased Passage MAP for 9 topics. So, our query expansion was neither completely helpful nor completely harmful to Passage MAP.

Unlike other two evaluation measures, NTUadh2 has the highest Aspect MAP among the three runs. But further comparing the Aspect MAP for each topic yielded the same observations as we had obtained for Document MAP and Passage MAP: the number of topics that were benefited by query expansion was about the same as the number of topics that were harmed by query expansion.

5. Conclusion

In this paper, we presented our methods for information retrieval at passage level. We

submitted three different runs for evaluation. Based on the comparison done on our runs, we saw that our query expansion does not affect the retrieval performance very much.

Acknowledgements

Research of this paper was partially supported by National Science Council, Taiwan, under the contract NSC-95-2752-E-001-001-PAE.

References

[1] Entrez Gene. http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?CMD=search&DB=gene.

[2] Entrez Pubmed. http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?DB=pubmed.

[3] J. Lafferty and C. Zhai. Document language models, query models, and minimization for information retrieval. In *24th ACM SIGIR Conference on Research and Development in Information Retrieval* (SIGIR'01), 2001.

[4] Lynx. http://lynx.browser.org/.

[5] Medical Subject Headings. http://www.nlm.nih.gov/mesh/.

[6] Pubmed Stopwords. http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Search&db=books&doptcmdl=GenBookHL&term=stopwords+AND+helppubmed%5Bbook%5D+AND+404029%5Buid%5D&rid=helppubmed.table.pubmedhelp.T43.

[7] The Lemur Toolkit for Language Modeling and Information Retrieval. http://www.lemurproject.org/.