IIT TREC-2006: Genomics Track

Jay Urbain
Information Retrieval Laboratory
Computer Science Department
Illinois Institute of Technology
Chicago, IL
urbajay@iit.edu

Nazli Goharian
Information Retrieval Laboratory
Computer Science Department
Illinois Institute of Technology
Chicago, IL
goharian@iit.edu

Ophir Frieder
Information Retrieval Laboratory
Computer Science Department
Illinois Institute of Technology
Chicago, IL
frieder@iit.edu

Abstract

For the TREC-2006 Genomics Track, we report on the effectiveness of composite information retrieval functions based on a dimensional data model for improving document, passage, and aspect search precision of genomics literature.

We designed an approach, and developed a corresponding search engine, based on a novel dimensional data model capable of document, paragraph, sentence, and passage level retrieval of genomics literature. By constructing a data warehouse style index with the flexibility of aggregating term statistics at multiple levels of document granularity, and incorporating key biological entities through shallow parsing of individual sentences, composite retrieval models combining multiple levels of contextual evidence can be efficiently developed to improve retrieval performance.

The genomics track for 2006 measured document, passage, and aspect retrieval using 27 topics created by active biological researchers. Each topic fit within one of four question-oriented topic templates: the role of a gene in a disease, the effect of a gene on a biological process, how genes interact in organ function, and how mutations in genes influence biological processes. Documents for this task come from a corpus of 162,048 full-text biomedical articles. Each form of retrieval was measured with a variant of mean average precision (MAP).

We submitted automatically generated results from three composite models to the TREC forum. All three models delivered results that significantly exceed the median results reported for the 2006 TREC Genomics track. The results of our best performing TREC model had MAP of 0.426 for document retrieval (53% above median), 0.055 for passage retrieval (129% above median), and 0.262 for aspect retrieval (125% above median).

1. Introduction

Biomedical literature makes heavy use of complex noun phrases, compound words, and acronyms in various forms to identify biological entities. Due to this complexity, it is imperative to match entities within a query to document terms within the proper local context to ensure high-precision document retrieval. For example, the entities "bovine spongiform encephalopathy", "transforming growth factor", and "insulin degrading enzyme" have much higher relevance when matched against document terms when they co-occur within a phrase, sentence, or passage versus being spread throughout a document where the component terms would no longer identify these entities. Indeed, several studies have shown that in many, but not all cases, passage retrieval alone can improve document retrieval performance. Passage retrieval has also been shown to be a key step for identifying the proper context for question-answering systems (Callan, 1994; Ittycheriah and Roukos, 2001; Kaszkiel and Zobel, 1997, 2001; Lin, 2006; Tellex, et al., 2003; White, et al., 2005). We therefore posit that identifying these entities in their various forms and in the correct local context requires inclusion of evidence at finer levels of granularity of document structure, and that retrieval models utilizing entity, sentence, passage, and document level information can improve contextual evidence and therefore improve retrieval precision for all modes of genomics literature search.

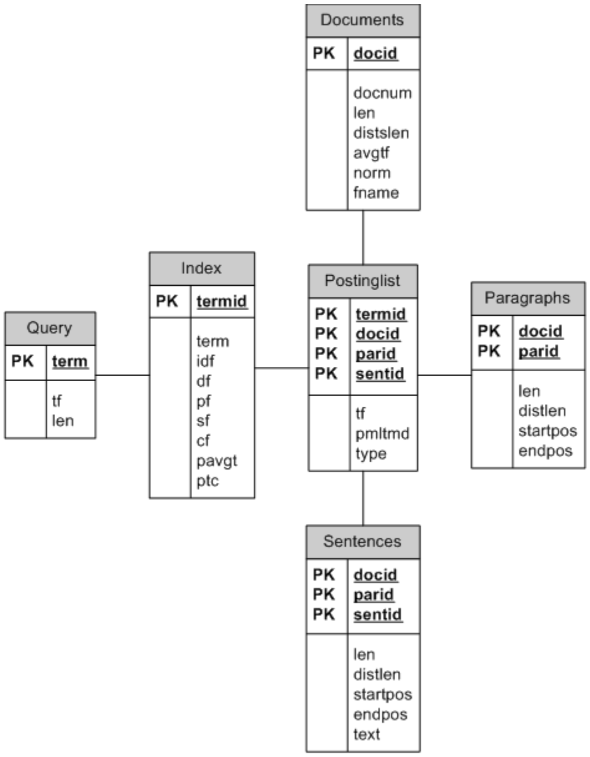
Integrated search of structured data and biomedical literature is critical for accurate retrieval, and thus, we designed a retrieval engine utilizing a dimensional data model developed using a standard relational database. The concept of building a search engine on top of relational technology is not new (Grossman, et al., 1997, Grossman and Frieder, 2004); however, such a multilevel approach had not been capitalized upon in the biomedical literature search domain. Building a text retrieval engine using a dimensional data model on a relational database allows flexible aggregation of term, sentence, passage, paragraph, and document statistics. Simultaneous search of structured data from biological databases and text-based biomedical literature can be accomplished using a single SQL query through seamless integration of both structured and unstructured data. Query augmentation, enhanced indexing techniques, and efficient evaluation of retrieval models can be accomplished through modification of SQL aggregation functions. In addition, queries can easily be developed for data and query analysis, allowing research efforts to focus on retrieval techniques rather than implementation details by leveraging off of the commercial database industry's investment in scalability, concurrency, and query optimization.

2. Dimensional Data Model

Our genomic retrieval engine is based on a relational implementation of information retrieval functions (Grossman, et. al., 1997) and uses relations to model an inverted index. Unlike previous relational and non-relational information retrieval data models, we employ a data warehousing inspired dimensional data model allowing us to aggregate document term statistics at multiple levels of document structure and granularity. It is best to visualize the central table, the *postinglist*, as a cube that can be sliced and diced to aggregate terms statistics. By utilizing such a dimensional data model, we facilitate development of simplified, efficient, and uniform retrieval functions capable flexibly aggregating statistics from multiple levels of document granularity.

As shown in **Figure 1**, the inverted index is implemented as a set of relational database tables: *Index, Postinglist, Documents, Paragraphs, Sentences, a Query table, and auxiliary tables with corpus statistics and meta-data (MeSH) from related structured data sources*.

Figure 1: Relational Model



Auxiliary tables are generated from structured data while parsing documents to capture corpus wide aggregate statistics and meta-data. An acronym table is generated during preprocessing and populated whenever a valid acronym expansion is located adjacent to an acronym. The Acronyms table is used to expand acronyms identified within queries, or add an acronym to a query when an acronym expansion is identified within the query terms.

Queries are formulated by joining the Query, Index, Postinglist, and Documents tables for document retrieval; adding a join of the Paragraphs table for paragraph retrieval; and also joining the Sentences table for sentence level retrieval. Passage retrieval is performed algorithmically as a set of contiguous sentences within a paragraph. Similarity coefficients are implemented as aggregate SQL functions within the select statement, and the Query table is populated with topic terms prior to query execution.

The following example illustrates document-level retrieval using the dimensional data model with the probabilistic BM25 retrieval function (Robertson and Walker, 2000). The subquery *"p"* aggregates *postinglist* sentence-level term statistics to obtain document-term statistics by grouping on document and term. Second, an outer query calculates document similarity scores by aggregating the results of the BM25 formula (**bold**) applied to each document-term statistic by grouping on document (**bold**). Finally, the document results are ordered in descending order of similarity.

```
select p.docid, max(d.docnum) docnum,
    sum( ln((s.ndocs-i.df+0.5)/(i.df+0.5))*
    (((k1+1)*p.tf)/(k1*((1b)+b*(d.len/s.avgdoclen))+p.tf))*
    ((k3+1)*q.tf/(k3+q.tf)) ) as sc
from index i,  documents d, query q, indexstats s,
    ( select p2.docid, p2.termid, sum(p2.tf) tf
    from postinglist p2, invertedindex i2, query q2
    where i2.termid=p2.termid and i2.term=q2.term
    group by p2.docid, p2.termid ) p
    where p.docid=d.docid
    and   i.termid=p.termid
    and i.term=q.term
    group by p.docid
    order by sc desc;
```

The same retrieval function can be used for paragraph-level retrieval by aggregating by document and paragraph. The data model aggregates term statistics at the sentence level; so no subquery is required to pre-aggregate statistics for sentence-level retrieval. Additional retrieval functions can be implemented by modifying the aggregate SUM function (**bold**) for either document, paragraph, or sentence retrieval.

3. System Description

Indexing, retrieval, and analysis applications were developed in Java and the system utilizes the Oracle 10g Personal Edition database. The system is platform and database independent. TREC retrieval runs were performed on a 3.1GHz Pentium 4 PC with 2 GB of main memory.

4. Preprocessing

Medline document abstracts were downloaded and parsed. MeSH were integrated with the relational model during indexing.

Frequent/infrequent terms were pruned from index. The stop-word list was augmented with frequently occurring genomics terms (disease, biology) and terms that do not support relevance (analysis, study). All non-acronym terms were stemmed with Porter postfix stemming.

Gene/protein terms were normalized with variants, e.g., TGF-beta1 -> {tgfbeta1, tgfbeta, beta1, beta, 1}.

During indexing, acronyms were parsed from sentences using a variation of the Schwartz and Hearst (2003) algorithm which identified acronyms and their adjacent expansions.

Query Processing

Queries were augmented with acronyms, whose expansion from indexing the collection matched extracted noun sequences in the query. For example, the acronym *EPT* was added for the term *electroporation*.

Queries were also augmented with compound terms, which were generated from successive noun terms within the query provided the generated compound was indexed with a normalized *idf* > 0.5.

Non function words, i.e., determiners, were also removed from the query.

5. Document, Paragraph Retrieval

We utilized the standard BM25 probabilistic algorithm for both document and paragraph retrieval. We developed but did not fully evaluate several other retrieval functions for this task including language models with Dirichlet, absolute discounting, and Jelinek-Mercer smoothing (Zhai and Laferty, 2001a, 2001b) and a relevance weighted language model (Hiemstra and de Vries, 2000). In our experience, BM25 has been more stable for a variety of IR tasks.

BM25:

$$\sum_{wq} \ln\left(\frac{N - df + 0.5}{df + 0.5}\right)\left(\frac{(k_1 + 1) * tf_d}{k1 * (1 - b) + b * (\frac{docLen}{avgDocLen}) + tfd}\right)\left(\frac{(k_3 + 1) * tfq}{k_3 + tf_q}\right)$$

We used k1=1.4, k2=0, k3=7, and b=0.75.

6. Passage Retrieval

Lacking a clear definition of a "passage", we defined passages as the longest set of contiguous sentences within a paragraph where the first and last sentences contain query terms. In addition to calculating the similarity coefficient scores using the traditional retrieval document functions defined above, we also defined two new retrieval functions to give more weight to query term density within a sentence or passage.

The first technique, document term proximity (DTP), measures document term proximity by calculating a co-occurrence value for each query term as the sum of the normalized IDF's of all other *distinct* query terms a particular query term appears with within a sentence:

$$DTP = \sum_{j=1}^{n} \sum_{i=1, i \neq j}^{j} NIDF(i)$$

DTP can be aggregated per sentence, passage, paragraph, or document.

The second technique, query term match (QTM), is similar to IBM's passage match score (Ittycheriah, et al., 2001). The QTM measurement sums the normalized IDF's of each *distinct* matching query term at the sentence level. Passage scores are aggregated as the top 3 sentence-level scores.

$$QTM = \sum_{I=1}^{n} NIDF(i)$$

7. Composite Scoring

Scores for the composite models were generated as linear weighted sums of the similarity coefficients (SC) at the document, passage, and sentence level:

$SC_{composite} = w_1SC_1 + w_2SC_2 + \ldots + w_nSC_n$

All scores are first normalized to between 0 and 1 before inclusion in the composite model.

8. Results

The results for all three of our automatically generated runs submitted to TREC are shown in **Table 1** below. All three results significantly outperformed the median results for the track. Results were scored as a composite of BM25 for document and paragraph retrieval, and our QTM function for passage and sentence scoring. The weighting of the composite function for each submission can be found under the SC column of the table. Assigning heavier weights to evidence from local context through the QTM function appears to have significantly improved results. We also believe a significant portion of our success is due our preprocessing and query augmentation techniques. **Table 2** shows the results of our top run *IITx1* versus the median results for the track.

Table 1 - Results from runs submitted to TREC

Run	SC	Doc	Passage	Aspect
IITx1	.66 * (sent. QTM) + .33 * (passage QTM)	0.426	0.055	0.262
IITx2	.5 * (sent. QTM) + .5 * (passage QTM)	0.388	0.036	0.187
IITx3	1.0 * (sent. QTM) + .10 * (passage QTM) + .01 * (document)	0.416	0.0513	0.255

Table 2 – Best run: IITx1 versus Track Median (MAP)

Retrieval Task	IITx1	Track Median	Difference	% Above Median
Document	0.426	0.279	0.147	53%
Passage	0.055	0.024	0.031	129%
Aspect	0.262	0.117	0.146	125%

9. References

Callan, J., 1994. Passage-Level Evidence in Document Retrieval, Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.

Grossman, D., Frieder, O., Holmes, D., Roberts, D., 1997. Integrating Structured Data and Text: A Relational Approach,. JASIS, 48(2).

Grossman D., Frieder, O., 2004. Information Retrieval: Algorithms and Heuristics, Second Edition; Springer Publishers, ISBN 1-4020-3003-7 (hardcover), 1-4020-3004-5 (paperback).

Hiemstra, D., de Vries, A. P. 2000. Relating the new language models of information retrieval to the traditional retrieval models, Technical Report TR-CTIT-00-09, Centre for Telematics and Information Technology.

Ittycheriah, A., Roukos, S., 2001. IBM's Statistical Question Answering System, TREC-11.

Kaszkiel, M., Zobel, J., 1997. Passage retrieval revisited, Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.

Kaszkiel, M., Zobel, J., 2001. Effective Ranking with Arbitrary Passages, JASIS.

Lin, Jimmy, 2006. The Role of Information Retrieval in Answering Complex Questions. Proceedings of the 21th International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistic.

Porter, M.F., 1980. An algorithm for suffix stripping, Program, 14:130–137.

Robertson S.E., Walker, S., 2000. Okapi/Keenbow at TREC-8, NIST Special Publication 500-246.

Schwartz, A., Hearst, M., 2003. A simple algorithm for identifying abbreviation definitions in biomedical text, Pacific Symposium on Biocomputing, Kauai.

Tellex, S., Katz, B., Lin, J., Fernandes, A., Marton, G., 2003. Quantitative Evaluation of Passage Retrieval Algorithms for Question Answering. Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.

White, R., Jose, J., Ruthven, I., 2005. Using Top-Ranking Sentences to Facilitate Effective Information Access, Journal of the American Society for Information Science and Technology, Volume 56, Issue 10.

Zhai, C., and Laferty, J., 2001a. A study of smoothing methods for language models applied to ad hoc information retrieval, 24th ACM SIGIR Conference on Research and Development in Information Retrieval.

Zhai, C., and Laferty, J., 2001b. Model-based feedback in the KL-divergence retrieval model, Proceedings of the Tenth International Conference on Information and Knowledge Management (CIKM 2001), pages 403–410.