IIT TREC-2005: 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-2005 Genomics Track ad-hoc retrieval task, we report on the development of a scalable information retrieval engine based on a relational data model for the integration of structured data and text.

Our objectives are to meet the need for the integrated search of heterogeneous data sets of biomedical literature and structured data found in biological databases, and to demonstrate the efficacy of using a relational database for a large biomedical information retrieval application.

Utilizing pivoted document normalization (PN) [1], pseudo relevance feedback [2, 3], and without performing stemming or domain specific normalization of biological terms, we received a mean average precision (MAP) of 0.1913 that places our results at the median of 32 Genomics track ad-hoc retrieval participants.

Subsequent to our participation in TREC, we have added a new gene/protein term normalization scheme, and have evaluated additional retrieval strategies including: BM25 [15], pivoted *unique* normalization [1], and language models utilizing absolute discounting, Dirichlet, and Jelinek-Mercer smoothing techniques [12, 13, 16].

With the addition of Porter stemming [17], gene/protein term normalization, and the BM25 probabilistic retrieval strategy, we received a MAP of 0.2879 that places us among the top results for official manual runs reported for the TREC Genomics track.

1. Introduction

Since biomedical research involves the use and integration of such a wide variety of different types of data, including structured descriptions and classifications of disease, drug interactions, protein descriptions, gene sequences, genus species, and study type, the integration of heterogeneous data sets, textual literature, and associated meta-data is required to improve search precision [4, 5, 6].

Unfortunately, these data sets are fragmented into many heterogeneous types, formats, vocabularies and databases. Most information retrieval systems have been developed utilizing custom inverted index structures that make integrated search across biomedical literature text and structured databases difficult.

Furthermore, many top performing participants in past TREC Genomics ad-hoc retrieval tasks have had success improving retrieval performance through utilization of external databases to normalize variations in biomedical terms [7, 8].

To address the need to integrate structured data with text, we explore the use of an information retrieval engine based on a standard relational database management system (RDBMS) for the TREC-2005 Genomics Track ad-hoc retrieval task.

The Genomics ad-hoc retrieval task utilizes a corpus of 4,591,008 Medline citations (~15GB) and 50 query topics drawn from 5 categories of information needs of molecular biology researchers [9].

2. Relational Data Model

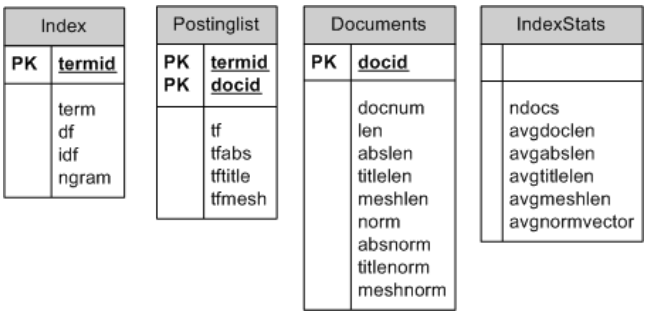
The mainstream approach in the development of information retrieval systems uses a customized inverted index to represent text with additional custom software required to integrate disparate structured and unstructured data sources.

To facilitate the integration of structured and unstructured biomedical data, we developed a new retrieval engine based on a relational information retrieval approach to provide a foundation for our research efforts [10, 11].

A relational information retrieval approach uses relations to model an inverted index. Storing the full text in a relational environment integrates the search of unstructured data with the traditional structured data search of RDBMS.

Our relational model implements an inverted index as a set of relational database tables: *index, postinglist, documents*. An *indexstats* table is created for capturing corpus wide statistics to support various normalization schemes, and a query table is created for storing the current topic.

Figure 1: Relational Model



All search queries are implemented using SQL with the aggregate SUM function implementing the similarity coefficient.

The following query implements a standard cross-product cosine with PN normalization:

```
select p.docid, max(d.docnum) docnum,
    sum(i.idf*(1+ln(q.tf))*idf*(1+ln(p.tf))*d.NORM )) as sc
from index i, postinglist p, documents d, query q
where p.docid=d.docid
and   i.termid=p.termid
and   i.term=q.term
group by p.docid
order by sc desc;
```

The performance of different retrieval strategies can be evaluated by modifying the aggregate SUM in the *select* clause.

Pre-computed normalization values can be easily updated with SQL. The following SQL updates the *documents* table for pivoted vector length normalization:

```
update documents set norm
        = 1/(0.8+((0.2/avgdoclen)*len));
```

Different term weighting schemes can be implemented for title, abstract, and MeSH terms by modifying the SUM equation as follows:

```
select p.docid, max(d.docnum) docnum,
      sum(i.idf*(1+ln(q.tf))*idf*
            (w1*(1+ln(p.tftitle)))+
            (w2*(1+ln(p.tfabs)))+
            (w3*(1+ln(p.tfmesh))) )
            *d.NORM )) as sc …
```

In addition, by loading the treceval qrels file containing listings of relevant and non-relevant documents into a database table, SQL reports can be generated to evaluate queries, relevant documents, and documents retrieved.

3. System Description

Indexing, retrieval, and analysis applications were developed in Java and the system utilizes the Oracle 9i Standard 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. Indexing

Official TREC results are reported with single term indexing, i.e., no bi-gram phrases. Stop terms were removed and no stemming was utilized. Some term and abbreviation normalization was performed when parsing the input data; however no domain specific term normalization of biomedical terms including variations in gene and protein names was performed.

After indexing, terms occurring in more than 30% of all documents were pruned from the index to improve performance. This modification did not affect accuracy and significantly improved query execution performance.

Subsequent indexing with bi-gram terms did not yield a statistically significant improvement in precision and significantly increased the size of the index.

Subsequent to our official TREC run we included Porter stemming [17], and added a new gene/protein term normalization technique based on the concepts used by Buttcher, et al., in the 2004 TREC Genomics track [8].

Biological Term Normalization

Gene/protein normalization requires identification of terms with mixed case, alpha-to-numeric, or numeric-to-alpha

character transitions that are not separated, separated by a space, or separated by a hyphen. Sample terms include Nurr-77, ApoE, NM23, and TGF-beta1. Nurr77, Nurr-77, and Nurr 77 would all be normalized to "Nurr 77'.

To capture additional variations, each variation of a normalized term is generated, so TGF-beta1 is first normalized to "tgf beta 1', and each component of the normalized term is generated prior to removing stop words: "tgf beta", "beta 1", "tgf", "beta", and "1".

5. Query Processing

The title and narrative from each topic was utilized to formulate the query, and the same preprocessing, i.e., term normalization, stop word removal, and Porter stemming utilized for indexing was utilized for query processing. Each of the following retrieval strategies were executed on the same index with the same preprocessing.

5.1 Pivoted Normalization

Standard normalization techniques such as cosine over penalize longer documents, i.e., shorter documents are more likely to be retrieved and less likely to be relevant [1]. Utilizing a slope "*s*" adjustment, pivoted normalization adjusts the retrieval curve to more closely represent the likelihood of retrieval.

$$\sum_{wq} \frac{idf * \ln(1+tf_q) * idf * \ln(1+tf_d)}{(1-s)+s*(\frac{doclen}{avgdoclen})}$$

We received our best results with s=0.3.

5.2 Pivoted Unique Normalization

Due to very high term frequencies, standard pivoted normalization can overweight long documents [1]. Since approximately 25% of the genomics MEDLINE citations have no abstracts, utilizing distinct term counts may better represent relevance with respect to citations with and without abstracts.

$$\sum_{wq} \frac{idf * \ln(1+tf_q) * idf * \ln(1+tf_d)}{(1-s) * avgDistDocLen + s * distDocLen}$$

We received our best results with s=0.25.

5.3 BM25

We utilized the standard BM25 probabilistic algorithm [11, 15]. After several trials, we received our best results with k1=1.4, k2=0, k3=7, and b=0.75.

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

5.4 BM25 with IDF

Utilizing the same constants used for standard BM25 term weighting, we received approximately the same result utilizing IDF weighting.

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

5.5 Language Models

Utilizing a most-likelihood-term unigram language model with uniform document priors we evaluated Jelinek-Mercer, Dirichlet, and absolute discounting smoothing. To improve performance, the *where* clause of the SQL query was modified to only include counts for documents which included at least one term that matched a query term.

5.5.1 Jelinek-Mercer

Jelinek-Mercer smoothing utilizes a linear interpolation to distribute the probability mass between terms *seen* in the document with the likelihood of the term occurring in the collection.

$$\sum_{wq} \ln((1-\lambda) * P_{ml}(w \mid d) + \lambda * P(w \mid C))$$

$P_{ml}(w \mid d) = tf_d / doclen$ and the collection model $P(w \mid C)$ represents the frequency of the term in the collection.

We received our best results with $\lambda = 0.1$

5.5.2 Bayesian Smoothing with Dirichlet Prior

Bayesian smoothing with Dirichlet prior:

$$\sum_{wq} \ln\left(\frac{tfd + \mu * P(w \mid C)}{docLen + \mu}\right)$$

We received our best results with $\mu = 2000$.

5.5.3 Absolute Discounting

Absolute discounting lowers the probability of seen words by subtracting a constant δ for seen term counts.

$$\sum_{wq} \ln\left(\frac{\max(tfd - \delta), 0}{docLen} + \frac{\delta * distDocLen}{docLen}\right)$$

We received our best results with $\delta = 0.8$

6. Results

We first report results from preprocessing improvements, and subsequently utilize our best preprocessing methods to systematically evaluate each retrieval strategy.

6.1 Preprocessing

To evaluate stemming and biological term normalization, we utilized PN (s=0.3) without relevance feedback, stemming, and biological term normalization as our baseline. **Table 1** lists the results of our preprocessing evaluation.

Table 1: Preprocessing Evaluation

	MAP	**Improvement over baseline**
Baseline PN	0.204	
+ stemming	0.213	4.4%
+ stemming +term norm.	0.251	23.0%
+ stemming +term norm. + term norm. variants	**0.266**	**29.4%**

Our first observation is the improvement in MAP from our official run (0.1913) to our new baseline by excluding relevance feedback. In our official run, we utilized one feedback iteration and expanded the original query with 3 additional terms selected by ranking terms by *df*idf* from the top 10 retrieved documents. After further examination of TREC results, we believe the difference is due to the relatively small number of relevant documents identified per query. Reducing the number of top documents (~5) from which to select query expansion terms negates the negative effect of relevance feedback, but does not improve mean average precision.

Stemming improved performance and the term normalization scheme dramatically improved mean average precision.

6.2 Retrieval Strategy Evaluation

Utilizing our *best* performing preprocessing, i.e., stemming and term normalization with variants, we evaluated each retrieval strategy. Results with PN used as a baseline are listed in **Table 2**.

Table 2: Retrieval Strategy Evaluation

Retrieval Strategy	**Parameter**	**MAP**	**Improvement over baseline**
PN	s=0.30	0.264	
PN Unique	s=0.25	0.268	0.8%
BM25	**k1=1.4, k2=0, k3=7, b=0.75**	**0.287**	**8.0%**
BM25 w/ IDF	k1=1.4, k2=0, k3=7, b=0.75	0.286	7.5%
LM Jelinek-Mercer	$\lambda = 0.1$	0.235	-11.2%
LM Dirichlet	$\mu = 2000$	0.240	-9.8%
LM Absolute discounting	$\delta = 0.8$	0.251	-5.6%

We consistently received the best results using BM25 for a wide range of parameters. Using the standard BM25 term weighting formula or BM25 with *idf* term weighting had no significant impact on results.

There was no significant difference between utilizing standard pivoted cosine normalization, or the pivoted unique normalization method that utilizes distinct term counts. Since pivoted unique normalization is most

effective in large documents and MEDLINE citations are relatively short, this is expected.

The performance of the language models for all smoothing techniques performed below the PN baseline. We also evaluated use of KL-Divergence [14] to measure the cross-entropy between a query model (represented by the likelihood of a term for a given query) and the unigram document model with each smoothing technique listed in Table 2. KL-Divergence did not improve performance. In all fairness, we utilized relatively basic models.

Use of relevance feedback did not improve performance for any of the retrieval strategies we evaluated.

7. References

[1] Amit Singhal, Chris Buckley, Mandar Mitra (1996). Pivoted Document Length Normalization. Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval.

[2] J. J. Rocchio (1971). The SMART Retrieval System. Experiments in Automatic Document Processing, chapter on Relevance Feedback in Information Retrieval, pages 313-323. Prentice Hall.

[3] Chris Buckley, Gerard Salton (1995). Optimization of Relevance Feedback Weights. Proceedings of the 18th annual international ACM SIGIR Conference on Research and Development in Information Retrieval.

[4] S.B. Davidson, C. Overton, and P. Buneman (1995). Challenges in integrating biological data sources. Journal of Computational Biology, 2(4), 557-572.

[5] W. David Fenstermacher (2005). Introduction to Bioinformatics. Journal of the American Society for Information Science and Technology, Volume 56, Issue 5, Pages 440 – 446.

[6] W. John MacMullen, and Sheila O Denn. (2005). Information problems in molecular biology and bioinformatics. Journal of the American Society for Information Science & Technology 56(5), 447-456.

[7] S. Fujita (2004). Revisiting again document length hypotheses: TREC 2004 Genomics Track experiments at Patolis. Proceedings of the Thirteenth Text REtrieval Conference, Gaithersburg, MD.

[8] S. Buttcher, CLA Clarke, and GV Cormack (2004). Domain-specific synonym expansion and validation for biomedical information retrieval (MultiText experiments for TREC 2004). TREC 2004 Genomics Track experiments at Patolis. Proceedings of the Thirteenth Text REtrieval Conference, Gaithersburg, MD.

[9] W. Hersh, et al. (2005). TREC 2004 genomics track overview.

[10] Ophir Frieder, Abdur Chowdhury, David Grossman, & M. Catherine McCabe (1999). DELOS Workshop: Information Seeking, Searching and Querying in Digital Libraries.

[11] David A Grossman, Ophir Frieder (2004). Information Retrieval: Algorithms and Heuristics. Springer Publishing.

[12] F. Song and W. B. Croft (1999). A general language model for information retrieval. Proceedings of the 22nd annual international ACM SIGIR conference, pages 279–280.

[13] C. Zhai and J. Laferty (2001). A study of smoothing methods for language models applied to ad hoc information retrieval. Proceedings of the 24th ACM SIGIR Conference on Research and Development in Information Retrieval.

[14] C. Zhai and J. Lafferty (2001). 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.

[15] S.E. Robertson, S. Walker (2000). Okapi/Keenbow at TREC-8. NIST Special Publication 500-246:The Eighth Text REtrieval Conference.

[16] J.M. Ponte and W.B. Croft (1998) A language modeling approach to information retrieval. Proceedings of the 21st ACM SIGIR Conference on Research and Development in Information Retrieval.

[17] M.F. Porter (1980) An algorithm for suffix stripping. Program, 14:130–137.