Using Relevance Feedback within the Relational Model for TREC-5

David A. Grossman
Office of Information Technology
3E09 Plaza B
Washington, DC 20505
dgrossm1@mason1.gmu.edu

Carol Lundquist
George Mason University
Fairfax, VA 22091
clundqui@osf1.gmu.edu

John Reichart
P2000 Technology Inc.
P.O. Box 916
Hanover, NH 03755
jreicher@lccinc.com

David Holmes
NCR
2 Choke Cherry Drive
Rockville, MD 20850
holmed@uf4725p02.WashingtonDC.ncr.com

Abdur Chowdhury
George Mason University
Fairfax, VA 22091
achowdhu@gmu.edu

Ophir Frieder*
Department of Computer Science
Florida Institute of Technology
ophir@cs.fit.edu

Abstract

For TREC-5, we enhanced our existing prototype that implements relevance ranking using the AT&T DBC-1012 Model 4 parallel database machine to include relevance feedback. We identified SQL to compute relevance feedback and ran several experiments to identify good cutoffs for the number of documents that should be assumed to be relevant and the number of terms to add to a query. We also tried to find an optimal weighting scheme such that terms added by relevance feedback are weighted differently from those in the original query.

We implemented relevance feedback in our special purpose IR prototype. Additionally, we used relevance feedback as a part of our submissions for English, Spanish, Chinese and corrupted data. Finally, we were a participant in the large data track as well. We used a text merging approach whereby a single Pentium processor was able to implement adhoc retrieval on a 4GB text collection.

* This work supported in part by the National Science Foundation under contract number IRI-9357785 and industrial matching funds under the National Young Investigator Program. Ophir Frieder is currently on leave from the Department of Computer Science at George Mason University.

1. Introduction

For TREC-5, we implemented relevance ranking queries using SQL on an AT&T DBC-1012 (formerly Teradata) parallel database machine [1]. This was an extension to our prior work which implemented the vector-space model as an application of a relational DBMS. Additionally, we implemented a special purpose IR prototype to test a new index compression algorithm and to provide performance comparisons to the relational approach.

We submitted official results for the 2GB English collection, both for automatic and manual adhoc queries and against the Spanish and Chinese collections. We also submitted results using n-grams to process the corrupted data. Each of these submissions included relevance
feedback.

We briefly describe the implementation of relevance feedback in our relational prototype and our special-purpose prototype in Section 2. More detailed descriptions are found in [2, 3]. Sections 3, 4, and 5 will describe the results obtained for our English, Spanish, and Chinese submissions. Section 6 describes our corrupted data results. Our conclusions are outlined in Section 7.

2. Implementation of Relevance Feedback

We developed two separate implementations, a parallel relational approach and a special purpose IR approach.

2.1 Implementation on the DBC

Our approach treats the information retrieval (IR) problem as an application of a relational database system. While parallel implementations of relational database systems are common, parallel implementations of IR systems are rare. Work done on large scale relevance feedback did include a parallel machine, but this work did not include within document frequencies. We implemented relevance feedback as an extension of the vector space model with standard *tf-idf* weights.

We model an inverted index with a relation DOC_TERM(*doc_id, term, tf*). A relation, QUERY(*query, term, tf*) indicates the terms and their frequency in the query. DOC(*doc_id, doc_name, doc_weight*) contains the document name and the normalized weight for each document. QUERY_WEIGHT(*query, query_weight*) contains the normalized query weight for each query. Finally, the IDF(*term, idf*) relation stores the inverse document frequency for each term.

Given these relations, the following SQL computes a cosine similarity coefficient for a given query: *query_number*.

```
Ex: 1   SELECT a.query, c.doc_name, SUM(a.tf * b.tf * e.idf * e.idf)/
                SQRT(d.query_weight * c.doc_weight)
        FROM QUERY a, DOC_TERM b, DOC c, QUERY_WEIGHT d, IDF e
        WHERE a.term = b.term
                AND a.term = e.term
                AND b.docid = c.docid
                AND a.query = d.query
                AND a.query = query_number
        GROUP BY a.query, c.docname, d.query_weight, c.doc_weight
        ORDER BY 3 DESC;
```

Assume the query given above has been executed, and the top *n* document identifiers are stored in the relation TOP_DOC(*doc_id*). To compute relevance feedback, the top *t* terms (sorted by some sort criteria) found in the top *n* documents are added to the query. This is accomplished with standard SQL used in each of the following steps:

Step 1 - Identify the top *n* documents for each query through relevance ranking.
Step 2 - Identify the terms from the top *n* documents.
Step 3 - Select the feedback terms to be used for relevance feedback.
Step 4 - Merge the feedback terms with the original query.
Step 5 - Identify the top documents for the modified queries through relevance ranking.

Each step is implemented by a standard SQL statement. Although a single SQL statement could be implemented, for clarity we use separate SQL statements. Hence, it is relatively straightforward to extend the relational approach to include relevance feedback.

2.2 Special Purpose IR Prototype

We also extended our special-purpose IR system to include relevance feedback. Our system implements relevance ranking using the vector-space model with the cosine similarity measure using *tf-idf* weights [4]. Implementation of relevance feedback was done by obtaining the top *n* documents and parsing their original text to find the terms in these documents. The terms were then sorted according to a specified sort order and added to the original query.

3. English Results

3.1 Automatic

We submitted both manual and automatic results for the adhoc collection. Each section of the corpus was loaded into a corresponding relation, and a larger query to UNION all the different relations was implemented. In addition to simply loading terms, we also loaded phrases which were recognized with a crude phrase parser. A phrase was defined as a two term sequence that did not contain a punctuation mark or a stop word. The topics were parsed in the same fashion and both terms and phrases were incorporated into the queries. Phrase inverse document frequency (IDF) was computed as if the phrase was a single term. All terms other than stop words were used in the query.

Our first submission, gmu96au1 used our relational prototype. Only terms from the <desc> portion (i.e., short version) of the query were used. Terms from the top 10 documents for the original query were identified. These were sorted by *n*idf* as given in [5] where *n* is the number of top ranked documents that have the term (*n* is between 1 and 10). The top 10 terms were added to the original query, duplicates were removed, and the query was executed again. Only a single iteration of relevance feedback was used.

The second submission, gmu96au2, used our special purpose IR prototype. Terms from both the <desc> and the <title> components of the query (i.e., long versions of the query) were used. The cosine similarity measure was executed for these terms, and again, the top 10 documents were assumed to be relevant. These terms were sorted by the same *n*idf* measure; however, the top 20 terms were added to the original query. These terms were added to the original query, and the cosine measure was computed. A scaling factor of .4 was applied to the new terms (several scales were tested on the TREC-4 collection).

3.2 Manual

The key difference in our two manual adhoc submissions is the use of manually assigned weights versus automatically assigned weights.

3.3 Manually Assigned Weights

Our first manual submission, gmu96ma1, used manually assigned terms and manually assigned weights. The terms for each query were derived by examining the initial query and identifying terms and phrases that appeared relevant. Since the document collection was not stemmed, many variants based on prefixes and suffixes are included. Relevance feedback was also used. Queries were executed using manual term selection and terms in the top ranked documents that appeared to be of potential benefit were then added to the query. Subsequently, a new query based on this manual feedback was executed and our final run used the results from this query.

The assumption is that queries are about one or more concepts. Terms are grouped into a *"concept"* via the operator given below. Up to three concepts are supported, hence an operator of 1, 2, or 3 indicates the term is in a particular concept. For a document to be ranked, it must have at least one term in each concept (unless the term is placed in a special concept 0 – in this case the document may not have the term and still be ranked). Once this condition is satisfied, all other terms are used to contribute to the similarity coefficient. The similarity coefficient is computed as the sum of the manually assigned weights for which a match occurs. This score is then divided by the total number of terms and phrases in a document (not including stop words). Negative weights were assigned for query terms that were specifically excluded relevant documents (i.e., "find info about taxes worldwide, NOT in the US")

3.4 Automatically assigned weights

In our second manual run, gmu96ma2, the basic approach was similar to the first run. All terms remain the same, but the term weights and ranking algorithm differ. The term weights that were used were automatically computed as the *idf* (inverse document frequency). The ranking algorithm still used the three concept sets. The similarity coefficient was computed using the cosine similarity coefficient. However, normalization was done based on the total number of non-stop words in the document rather than the typical cosine length normalization.

3.5 Results

Our overall results for English are given below:

Test Run	Description	Avg. Precision	Above Median	Below Median	Equal Median
gmu96au1	Automatic (Relational IR, relevance feedback with top 10 terms)	.1079	10	37	3
gmu96au2	Automatic (Special IR, relevance feedback with top 20 terms, .4 scaling factor for new terms)	.1331	13	35	2
gmu96ma1	Manual (manually assigned weights)	.2147	21	27	2
gmu96ma2	Manual (automatic assigned weights)	.2141	19	26	5

It is reasonable to expect that our two automatic implementations would have similar results as their basic techniques are nearly identical. Our calibrations showed that a scaling factor did slightly improve precision/recall, and that is verified here. It should be noted that our calibrations consistently found precision/recall in the .20 to .22 range on the TREC-4 collection. We are currently investigating the reason for this reduction in precision/recall when the same approach was used on the TREC-5 collection. The manually assigned weights performed no better than automatically assigned weights for the manually constructed queries.

3.6 Failure Analysis

One of the interesting features of relevance feedback is that while relevance feedback improves precision/recall for some queries, it also decreases precision/recall for others. It would be useful to be able to predict those queries which would benefit from relevance feedback so that relevance feedback would not be applied to those queries whose precision/recall would decrease. In an effort to identify such a predictor, we analyzed the query terms both before and after relevance feedback for the gmu96au1 run. Based on this analysis, the queries were divided into three groups: queries that relevance feedback improved precision/recall (16 queries), queries that relevance feedback did not change precision/recall (14 queries), and queries that relevance feedback decreased precision/recall (20 queries).

TOPICS	AVG # TERMS PER QUERY	AVG IDF OF TERMS Original Query			AVG IDF OF TERMS After RF		
		IDF	MAX IDF	MIN IDF	IDF	MAX IDF	MIN IDF
IMPROVED BY RF 254,257,258,259,264,265,267,273,280,282, 283,284,287,288,298,299	15.1	2.46	0.90	4.80	2.96	1.63	4.43
UNCHANGED BY RF 252,253,256,260,263,268,272,278,279,281, 292,296,297,300	16.3	2.38	0.82	4.73	2.61	1.32	4.30
DECREASED BY RF 251,255,261,262,266,269,270,271,274,275, 276,277,285,286,289,290,291,293,294,295	16.6	2.41	0.90	4.83	2.74	1.43	4.43
ALL QUERIES	16.0	2.41	0.88	4.79	2.77	1.46	4.39

The table below illustrates the terms both before and after relevance feedback for several queries.

TOPIC	IMPACT OF RELEVANCE FEEDBACK	TERMS AND PHRASES Original Query	New Terms Identified by RF
#264 Identify instances where U.S. citizens have been or are being held in foreign jails since the year 1900.	Improved 320%	• jails since • foreign jails • identify instances • jails • being held • instances • identify • citizens • held • foreign • year • being • since	• longest held • **nine americans** • chief middle • **south lebanon** • prisoners • release
# 272 Medically, is outpatient surgery more prevalent now than ever before?	No Change	• surgery more • outpatient surgery • medically • outpatient • surgery • ever • more	• surgery centers • inpatient • surgical • health care • hospital • medical • care

# 262 Is seasonal affective disorder syndrome (SADS) (also known as seasonal absence of daylight syndrome), a worldwide disorder?	Decreased 100%	• sads • seasonal affective • affective disorder • affective • daylight • disorder • syndrome • seasons • worldwide • absence • known	• phobias • **tics** • disorders • symptoms • depression • illness • disease

As seen in the table above, the average *idf* of terms in the queries is very similar for those queries that were improved by relevance feedback to those queries that were not improved. However, after relevance feedback was applied, queries which benefited from relevance feedback had a slightly higher average *idf* than queries whose effectiveness was decreased by relevance feedback. The relationship between the weight of the terms in a query and the improvement obtained from relevance feedback needs further investigation.

4. Spanish Results

For the Spanish data, we used the relational prototype to obtain our automatic results. Both results were done with relevance feedback with only a difference in scale between each result.
We developed a Spanish stop word list by identifying the top 500 most frequent terms and asking a Spanish linguist to determine which ones were really not so common across the language that they should be in a stop list.

Essentially the same approach was used as during our adhoc run. The top ten documents were identified by using Spanish terms and the usual cosine measure. The terms in these documents were ordered by the *n*idf*, and the top ten terms were added to the query. The initial run does not do any scaling while the second run increases all phrase weights in the query by a factor of five.

4.1 Results

Our results for Spanish data are given below:

Test Run	Description	Avg. Precision	Above Median	Below Median	Equal Median
	Cosine	.2215	6	19	0
gmu96sp1	Cosine+rf	.2403	6	18	0
gmu96sp2	Cosine+rf+scale	.1900	4	21	1

The baseline run was not submitted as an official result, but it is given here to measure the effect of relevance feedback. Relevance feedback without scaling improved precision by a small margin (around 8%). The use of scaling **clearly did not improve** performance for these queries.

5. Chinese Results

For the Chinese data, we used our special purpose IR prototype. We implemented both manual and automatic relevance feedback.

5.1 Automatic

The first run is a baseline run. The cosine measure was used with tf-idf weights. To parse Chinese, we took the simplistic approach of assuming each term was one two-byte character. No stop words were used. We used all Chinese components of the query (description, narrative, and title). The second run using automatic relevance feedback with the same technique as described for adhoc English. Without any training data we were forced to assume that Chinese would perform in a similar fashion to English for relevance feedback.

The results from the first run were obtained, and the top 10 documents for each query were identified. The top 20 terms were selected based on the *n*idf* measure. The original query was then augmented to use the new terms, and a scaling factor of .4 was applied to the new terms. These values were obtained from calibration using the English data with TREC-4 qrels and mirror one of our English submissions.

5.2 Manual

Both of our manual runs use manual relevance feedback. Instead of blindly assuming that the top 10 documents were relevant, we asked two people who were fluent in Chinese to read the top ten documents and indicate which ones were relevant. Once this was done the top ten terms from these documents were added to the collection, and the same computation as done for the automatic runs was computed. Two differences exist between the two manual runs. The first is that a different relevance assessor was used for each run. The second is that the entire query is used in run 1, but only the <description> portion of the query is used in run 2.

5.3 Results

Our results for Chinese data are given below:

Test Run	Description	Avg. Precision	Above Median	Below Median	Equal Median
gmu96ca1	Automatic Cosine	.2955	8	10	1
gmu96ca2	Automatic Cosine+rf	.3274	12	6	1
gmu96cm1	Manual Cosine+rf+whole query	.3279	12	7	0
gmu96cm2	Manual Cosine+description	.3065	11	8	0

The results indicate that relevance feedback is of benefit for Chinese data. Also, we again find that the manual effort (in this case reading nearly 200 pages of printed Chinese) did not yield any significant improvement over the automatic approach.

6. Corrupted Data Results

With corrupted data, we relied upon 4-grams (overlapping sequences of four characters) to be resilient to errors in text. Our first submission used a standard cosine measure on all three test collections (baseline, 5% corrupted, and 20% corrupted). Our second submission incorporated relevance feedback using n-grams.

As the test collection for the confusion track did not contain data for which relevance assessments existed, it was not possible to calibrate for the data on this collection. Hence, we used the exact same relevance feedback technique that we used for the adhoc English collection except that terms are replaced by n-grams. The standard cosine measure was used with the exception that terms were replaced with 4-grams. The n-grams spanned term boundaries. Hence, for an input phrase of New York, the n-grams were: new_, ew_y, w_yo, york, ork_.

The query was generated by parsing the input query and generating its component n-grams. We used the same stop word list as used for the adhoc English collection. Any term that was found on this list was eliminated before n-grams were generated. In addition to this, we generated a stop-n-gram list in which the top .05% n-grams were eliminated. The n-grams were sorted based on their collection frequency. Also, we ensured that no more than 150 n-grams were added to this list. The stop-n-gram list was generated for each version of the corrupted data (baseline, 5% corrupted, and 20% corrupted). We used the same relevance feedback process as used for the adhoc English collection. N-grams were parsed and a cosine measure was computed using n-grams instead of terms. The top n-grams in the top ten documents were identified and were sorted by the same n*idf measure. The top 20 n-grams were used with a scaling coefficient of .4.

Our results for corrupted data are given below. Since this was a search for a known item, we give the mean of the reciprocal of the rank at which the known item was found for all 49 queries.

Test Run	**Description**	**Degrade 0**	**Degrade 5%**	**Degrade 20%**
gmu961	Cosine	.39	.31	.22
gmu962	Cosine+rf	.20	.19	.15

7. Conclusions and Future Work

Given that this was only our second year as a Category A participant, we still see much room for improvement. Overall, we confirmed a known result that relevance feedback is clearly of benefit to English language processing. Our new work in this area is that relevance feedback can be implemented using the relational model. This yields a portable, parallel approach to computing relevance feedback. We experimented with several sort orders to find the optimal number of documents to retrieve and number of terms to add to the query and we ended up at 10 documents retrieved with either 10 or 20 terms to add to the query. As work done in [6] indicates that other term weighting methods outperform the *tf-idf* weights, we will incorporate this information and develop relational implementations using standard SQL of alternative term weighting methods. Also, we will continue the search for indicators of when relevance feedback should be applied and when it should not be applied to individual queries.

The use of relevance feedback clearly helped our manual queries as well. Our manual English results were more than double the precision of our automatic approach. However, a significant amount of time per query (15 to 30 minutes) was used to develop these queries. Relevance feedback worked reasonably well with Chinese data and we have an initial result that suggests that manual relevance feedback does not improve on automatic relevance feedback.

Overall, our final numerical results were similar to our results for TREC-4. Our calibration during the year suggested that our effectiveness would increase by 10 to 20 percent, but we were unable to calibrate with Chinese or corrupted data. We will use the collection from TREC-5 as training data for future work.

References

[1] AT&T Global Information Systems. *Teradata DB2-1012 Concepts and Facilities*, March 1992.

[2] D. Grossman, O. Frieder, D. Holmes, and D. Roberts. Integrating Structured Data and Text: A Relational Approach, *Journal of the American Society of Information Science*, January 1997.

[3] D. Grossman, Integrating Structured Data and Text: A Relational Approach, Thesis, George Mason University, 1996.

[4] G. Salton, C.S. Yang, and A. Wong. A vector-space model for information retrieval. C*ommunications of the ACM*, 18, 1975.

[5] D. Harman. "Relevance Feedback Revisited," *Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Ed. Nicholas Belkin, Peter Ingwersen and Annelise Mark Pejtersen, SIGIR Forum, June 21-24, 1992.

[6] C. Buckley, A. Singhal, M. Mitra, and G. Salton, "New Retrieval Approaches Using SMART: TREC 4," Text Retrieval Conference, sponsored by National Institute of Standards and Technology and Advanced Research Projects Agency, November 1995.