Expanding Relevance Feedback in the Relational Model

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

David O. Holmes
NCR Corporation
Rockville, Maryland
David.Holmes@WashingtonDC.NCR.COM

David A. Grossman
Office for Research & Dev.
Washington, DC
dgrossm1@osf1.gmu.edu

Ophir Frieder*
Florida Institute of Technology
Melbourne, Florida
ophir@ee.fit.edu

M. Catherine McCabe
George Mason University
Fairfax, Virginia
cmccabe@gmu.edu

Abstract:

In TREC-6, we participated in both the automatic and manual tracks for category A. For the automatic runs, we used the short versions of the queries and enhanced our existing prototype by expanding the relevance feedback methodology to include additional term weighting methods (i.e., the typical "*ltc-lnc*" or "*nidf*" weights) as well as feedback term scaling. We also experimented with eliminating infrequently occurring terms to determine if the relevance ranking scores between documents and queries could be improved by eliminating certain highly weighted terms. For our manual runs, we used pre-defined concept lists with terms from the concept lists combined in different ways. We continued to use the AT&T DBC-1012 Model 4 parallel database machine as the platform for our information retrieval system which continues to be implemented in the relational database model using unchanged SQL.

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

1. Introduction

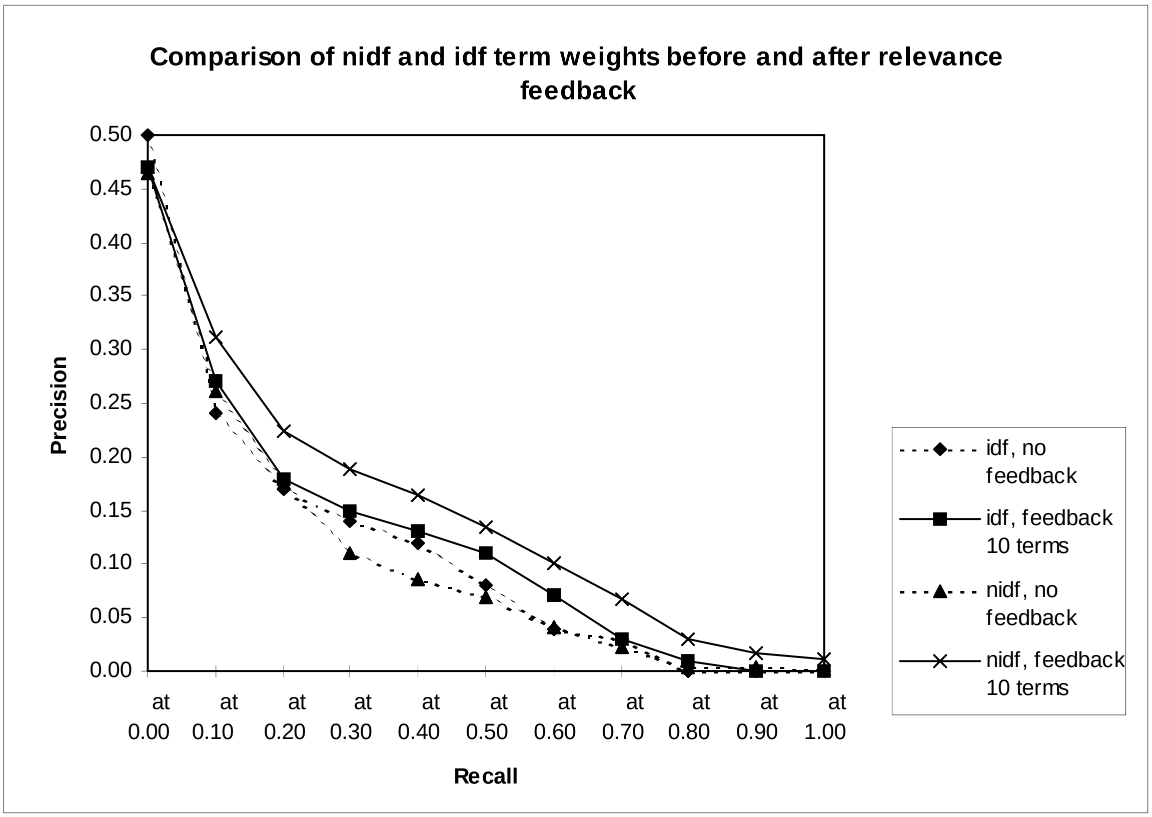
Our work for TREC-6 is a continuation of the work started in TREC-4 when we implemented an information retrieval system as an application of a relational database (RDBMS). We used unchanged SQL to implement vector-space query relevance ranking (Grossman95, Grossman96). The TREC-4 work was expanded upon for TREC-5 when we implemented a basic form of relevance feedback, also using unchanged SQL. For TREC-6, we expanded our relevance feedback methodology to include the lnc-ltc term weights (Singhal96) as well as feedback term scaling. In addition to expanding and improving our relevance feedback methodology, we also experimented with methods to improve the precision and recall scores of our pre-relevance feedback baseline run. To explore the assumption that certain infrequently occurring terms with high collection weights may actually be artificially inflating the query-to-document relevance ranking scores, we experimented with eliminating infrequently occurring terms from the collection. This approach shows promise for improving the baseline scores and has other advantages such as reducing the processing time per query and disk storage space for the document collection.

Our manual runs also represent a continuation of the work started in TREC-4. In TREC-4, we assigned the query terms in up to three concept lists and used general world knowledge to expand the query to include other similar terms not found in the topic. In TREC-5, we continued to use the concept lists and experimented with the use of manually assigned weights to the query terms as well as using manual relevance feedback to identify additional terms. For TREC-6, we augmented our prior work with inexact term matching and an automatically generated thesaurus based on term-to-term co-occurrence. Our first run uses up to three concept lists. To assess the value of using concept lists, our second run uses the same terms and scoring algorithm as the first run, but all of the query terms are placed into a single list. Essentially, multi-concept topics were changed from an intersection to a union of documents. We also introduce a Soundex variation (Celko95) as a tool for expanding the concept lists with similar terms. Finally, an association rule is used to identify co-occurring terms. Full details of these methods and the methods used for the automatic runs are described in sections 3 and 4.

2. Implementation of an Information Retrieval system using the Relational Model

This section provides a brief overview how our information retrieval (IR) system is implemented using the relational model. Full details of the implementation can be found in (Grossman97 and Lundquist97a).

To test the effectiveness of the lnu-ltc or "nidf" term weights over the inverse document frequency or "idf" term weights, we ran several calibration runs on the TREC-5 data to compare the differences in precision and recall both before and after relevance feedback. Figure 1 shows the difference in precision and recall for the two term weighting methods.



--- Figure 1 ---

Using 10 feedback terms, with feedback terms selected by the n * term weight method when relevance feedback was done, and using a subset of documents from Tipster disks 2 and 4 along with the TREC-5 queries, the following results were obtained:

Type of Relevance Feedback	Average Precision	Percent change	Exact Precision	Percent change
idf, no feedback	.0966	----	.1410	----
idf, feedback 10 terms	.1100	+14%	.1421	+1%
nidf, no feedback	.0914	----	.1306	----
nidf, feedback 10 terms	.1400	+53%	.1755	+34%

Table 1 -- Comparison of average and exact precision

An additional benefit to using the relational model for IR is the ability to exploit parallel processing via the DBMS. We implemented an IR system using Teradata's RDBMS on a 4 processor DBC/1012 parallel processing machine. The Teradata DBC/1012 Database Computer is a special purpose machine designed to run a relational database management system using standard SQL.

3. Automatic Results

3.1 First Automatic Run

Our first automatic run used standard relevance feedback similar to that originally proposed by Rocchio in (Rocchio71). For this run, we used the formulas described in (Ballerini96 and Buckley95) to perform an initial relevance ranking to identify the 20 top-ranked documents for each query. We selected the 10 top-ranked feedback terms contained in these documents using the N * *nidf* sort order where N is the number of documents out of the 20 top-ranked documents containing the term and *nidf* is the weight of the term in the document collection. The 10 feedback terms were then adjusted by a scaling factor of 0.5 and added to the original query. The query-to-document relevance ranking was then recomputed using the modified query, and the 1000 top-ranked documents were identified. Further details on the experiments done to determine the optimal number of top-ranked documents and relevance feedback terms to use along with the sort order and scaling for the feedback terms can be found in (Lundquist97b).

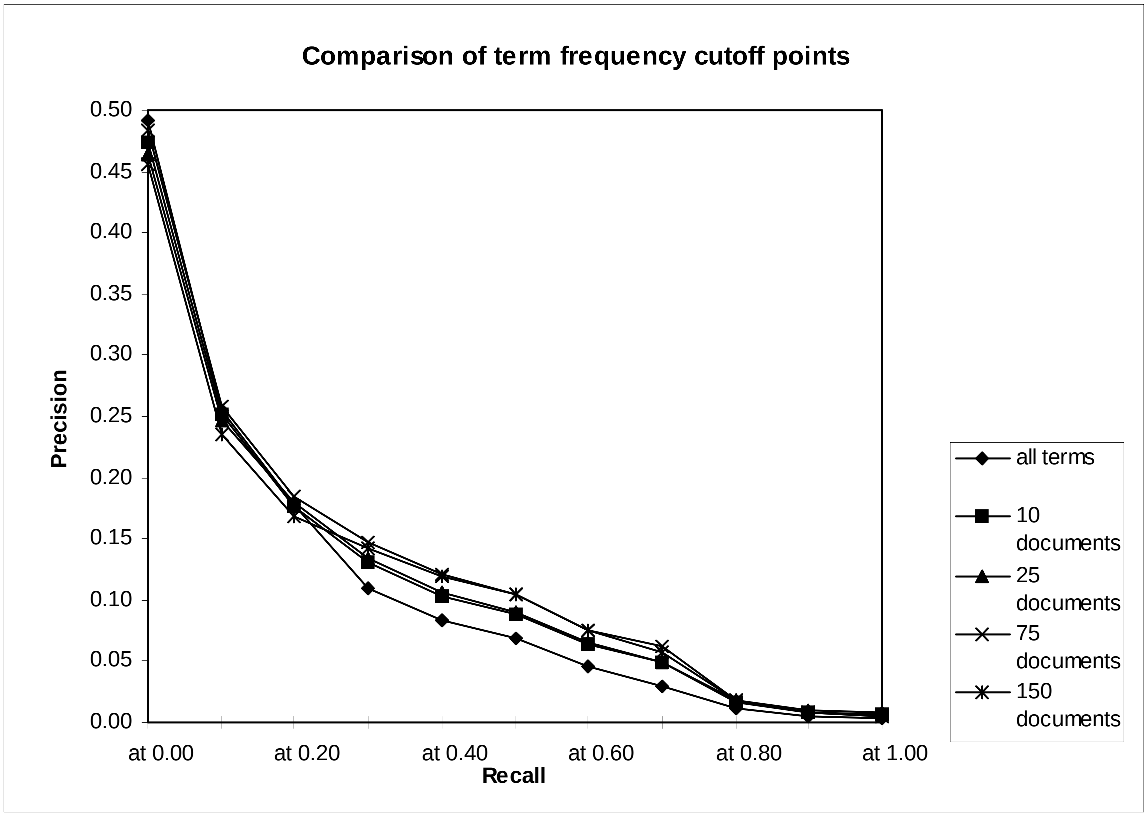
Table 2 shows the comparison of the results from our first automatic run with the other short topic automatic runs submitted and lists the number of queries where we achieved results that were either best, above the median, at the median, or below the median.

	Best	**Above Median**	**At Median**	**Below Median**
Average precision (non-interpolated)	1	29	1	19
Number of relevant documents retrieved	10	23	8	9

Table 2 -- Results comparison for gmu97au1

3.2 Second Automatic Run

In our second automatic run, we did not use relevance feedback. Instead, we attempted to improve the precision and recall scores of our baseline run by experimenting with term frequency cutoff points. To do this, we essentially expanded the stopword list to exclude terms which occurred infrequently in the document collection. To explore the possibility that the large term weights of the infrequently occurring terms may be artificially inflating the relevance ranking scores of documents, we eliminated all terms that occurred in less than 75 documents in the document collection and performed the routine query-to-document relevance ranking. A comparison of the precision and recall levels at different frequency cutoff points can be seen in Figure 2.



--- Figure 2 ---

Using only relevance ranking, *nidf* term weight method, and documents from Tipster disks 2 and 4 with the TREC-5 queries, we obtained the following results:

Terms eliminated if occurring in less than N documents	**Average Precision**	**Percent change**	**Exact Precision**	**Percent change**
all terms	.0928	----	.1346	----
10 documents	.1032	+11%	.1426	+6%
25 documents	.1051	+13%	.1423	+6%
75 documents	.1149	+24%	.1514	+12%
150 documents	.1083	+17%	.1444	+7%

Table 3 -- Comparison of average and exact precision

Since Tipster disks 4 and 5 combined contain approximately 525,000 documents, 75 documents represents approximately .014% of the document collection. Since infrequently occurring terms make up a large percentage of the number of distinct terms, eliminating terms occurring in less than 75 documents allowed us to reduce the amount of storage required by 26%. Table 2 shows the average and exact precision scores obtained during our calibration runs using the TREC-5 queries. Based on these calibration runs, eliminating terms occurring in less than 75 documents generated the most improvement (i.e., 24%) over the baseline scores.

The calibration runs on the TREC-5 queries showed that while using term frequency cutoff points did not perform as well as relevance feedback, it did produce a significant improvement over the baseline scores. At the same time, the term frequency cutoff points allowed for a significant reduction in processor time because the second relevance ranking run necessary for relevance feedback was not done. Using term frequency cutoff points also allows overall disk storage to be considerably reduced by eliminating certain terms.

Table 4 shows the comparison of the results from our first automatic run with the other short topic automatic runs submitted and lists the number of queries where we achieved results that were either best, above the median, at the median, or below the median.

	Best	**Above Median**	**At Median**	**Below Median**
Average precision (non-interpolated)	3	17	3	20
Number of relevant documents retrieved	6	17	9	18

Table 4 -- Results comparison for gmu97au02

4. Manual Results

4.1 First Manual Run

Query creation for our first manual run included multiple processing steps. To initially create the manual runs, we examined each topic and selected terms and two word phrases that appeared relevant. We used one pass of relevance feedback and a term-term association list (based on term-term co-occurrence) to give the user potential terms to use in a query. Our user then selected terms and phrases thought to be relevant. The terms were grouped into concept lists based on the assumption that every topic relates to one or more concepts. To be ranked for a given topic, a document had to contain at least one term from each concept list. The remaining terms in the concept list simple increase the similarity measure – they are not all required to be present in a document. A catch-all list, not part of a concept and not used to qualify documents, had words used for weighting qualified documents. Qualified documents were scored by considering the number of query terms (Q1) shared by a document (X1). The number of distinct terms (K1) tempered results for large documents.

$$\text{relevance score} = (Q1 \cap X1)/K1$$

A Soundex variation was used to expand queries with similar terms. Phrases were assigned two soundex codes, one for each word. Terms and phrases with matching soundex codes were ranked using a similarity coefficient (Pfeifer96) which uses the digram sets for the condition (D1) and result (D2) terms. Digram sets include one leading and one trailing blank to weight the beginning and ending of terms. For example, the word "dog" has the digrams: "_d", "do", "og", and "g_".

$$\text{similarity coefficient} = (D1 \cap D2)/(D1 \cup D2)$$

For a limited number of queries we collected associated terms using an improvement formula (Berry97) used for market basket analysis. Our minimum support was ten documents and the maximum support was 1,000. This deviates from the minimum support of 75 used in the automatic runs.

$$\text{improvement} = p(\text{condition and result})/ (p(\text{condition})\ p(\text{result}))$$

Finally, we implemented a casual relevance feedback technique. The initial query was run and a list of terms from a few of the top-ranked documents were inspected. If some terms appeared relevant, then they were added to the query and it was run again to produce final results. In most cases, associated and feedback terms were limited to proper nouns. In a few cases, such as topic 349, terms were removed as a result of feedback. For topic 349, the terms "anabolic" produces a large number of documents related to the use of steroids by athletes which did not appear relevant.

4.2 Second Manual Run

Our first manual run, like TREC-4 and TREC-5, used concept lists which create a qualified list of documents that are an intersection of every concept related to a topic. The goal was to create a concise and precise answer to a search request. To measure our assumption, the second run uses the same terms and scoring algorithms as the first run, but instead creates a union of the documents. As discussed in section 4.4, the intersection approach results in better precision.

4.3 TREC-6 Failure Analysis of Manual Queries

Our manual results did not contribute to the judged relevant document collection and therefore our precision and recall scores may be artificially low. Table 5 presents, at various document retrieval levels, the number of documents judged relevant or non-relevant and not judged at all. An interesting measure that may compensate for the lack of relevance assessments is to omit non-judged documents from the measure of precision – this assumes non-judged documents were neither relevant nor retrieved. Precision is then defined as the ratio of the number of judged relevant documents to the number of judged documents at various retrieval levels. Using this measure, the difference in precision is dramatic. Nearly 40% of our results at 100 documents retrieved were not evaluated. By eliminating non-judged results, our precision increased from 19.38% to 34%.

Documents Retrieved	Judged Relevant	Judged Not Relevant	Not Judged	Pct Not Judged	TREC-6 Precision	Precision on Judged Only
at 1	16	19	15	0.3000	0.3200	0.4571
at 5	81	101	68	0.2720	0.3240	0.4451
at 10	150	204	145	0.2906	0.3000	0.4237
at 15	232	293	219	0.2944	0.3093	0.4419
at 20	293	390	306	0.3094	0.2930	0.4290
at 30	408	589	482	0.3259	0.2720	0.4092
at 100	969	1881	1873	0.3966	0.1938	0.3400
at 200	1346	3340	4115	0.4676	0.1346	0.2872
at 1000	2228	7557	17601	0.6427	0.0446	0.2277

Table 5 -- Document Retrieval Level Performance

Table 6 below indicates the query-by-query examination of our first manual run. Interestingly, when over half of the documents were judged, twenty-eight of thirty-four queries were at or above the median. When under half of the documents were judged, only six of the sixteen remaining queries were at or above the median.

Topic	# of Topic Terms	# of Concepts	Judged Relevant 100 docs	Judged Not Relevant 100 docs	Not Judged 100 docs	Estimate Relevant 100 docs	TREC-6 Best 100 documents	TREC-6 Median 100 Documents
301	45	1	4	10	86	33	87	61
302	25	1	50	47	3	51	58	31
303	44	1	10	90	0	10	10	9
304	106	2	41	26	33	52	78	27
305	57	1	2	72	26	11	13	2
306	89	2	7	30	63	28	84	43
307	39	1	20	36	44	35	84	28
308	7	1	2	7	0	2	4	3
309	28	1	0	59	41	14	2	0
310	23	2	3	18	13	7	10	4
311	24	1	90	7	3	91	97	71
312	33	1	9	17	74	34	11	8
313	17	1	74	14	12	78	82	56
314	11	1	16	36	48	32	33	16
315	92	1	6	30	64	28	38	6
316	12	1	34	14	13	38	34	22
317	27	1	5	26	69	28	13	8
318	35	2	3	19	78	30	14	3
319	21	1	13	9	78	40	43	28
320	15	2	5	54	25	14	6	4
321	41	1	4	4	92	35	68	29
322	34	2	16	43	41	30	26	5
323	15	1	34	46	0	34	36	25
324	25	3	81	13	6	83	88	62
325	22	1	7	78	15	12	14	8
326	30	1	24	65	11	28	46	25
327	32	1	3	63	34	15	12	5
328	5	1	9	38	8	12	9	6
329	24	2	20	35	45	35	35	13
330	27	2	18	45	37	31	37	13
331	23	1	17	19	64	39	72	44
332	37	2	56	31	13	60	99	34
333	19	1	26	50	24	34	44	26
334	41	1	13	67	20	20	17	10

335	30	1	45	46	9	48	59	24
336	45	1	5	88	7	7	6	2
337	26	1	33	49	18	39	52	33
338	16	1	4	94	2	5	5	3
339	13	1	7	71	22	14	10	7
340	27	1	29	47	24	37	52	19
341	54	2	10	20	70	34	44	30
342	36	2	9	37	54	27	15	8
343	56	1	14	6	80	41	84	14
344	14	1	4	54	42	18	4	3
345	36	1	7	31	62	28	24	9
346	66	2	1	16	83	29	34	5
347	31	1	27	9	64	49	68	26
348	11	1	2	6	92	33	5	5
349	30	1	23	56	21	30	36	19
350	32	1	27	33	40	41	54	27
Total			969	1881	1873	1606		

Table 6 -- Individual Topic Performance

4.4 Comparative Results

Our measured results varied greatly by topic. Sometimes the results varied because of the complexity of the topic and other times because of the number of documents evaluated. Figure 3 aggregates our results into five groups based on the number of documents in the result set judged for TREC-6. Table 7 shows how far, in terms of the cumulative number of documents, our results were from the median as well as count the number of queries within the group that were at, above or below the median. For the ten most judged topics, nine out of ten had more than the median number of relevant documents retrieved. Similarly, for the ten least frequently judged documents, eight were below the median.

A possible explanation for having so many results unique to our queries is the use of association rules and soundex searches to expand or replace query terms. For example, we did not use a single word or phrase directly from topic 301. Instead we used some of the original terms as input to an association rule to identify the names of individuals, organizations, or activities associated with crime. Table 8 shows all of our query terms and phrases for topic 301. By probably not sharing many topic critical words with other teams, our results for query 301 were largely unevaluated. Table 9 identifies similar terms found by doing a soundex search. We hypothesize that other teams found many of the same results as our team for topic 302 because we shared topic critical words such as “polio”, but ours ranked fairly well because we stacked the query with several similar words which helped weight relevant documents.

The first and second manual runs used the exact same scoring metric and query terms. The initial run used concept lists to intersect documents by requiring the existence of at least one term from each concept list. The second run required only a single term from the entire query to retrieve a document. Any queries having more than one concept list, or a single concept list and additional weighting terms produced different results. Intersections provided much greater precision. Table 10 compares results at various retrieval levels.

Groups Ranked by # of Docs Judged	**Above Median**	**Median**	**Below Median**
Top 10	9	1	0
Upper Middle	5	3	2
Middle	5	1	4
Lower Middle	4	4	2
Bottom 10	1	7	2

Table 7 -- Performance versus Median

abbas musawi	john gotti	enrique camarena	plo gunman
abu nidal	khan younis	ernesto samper	rafael abello
ahmed yassin	lockerbie bombers	evaristo porras	rodriguez gacha
aldo moro	lockerbie bombing	giovanni falcone	royal ulster
alvarez machain	luis ochoa	giulio andreotti	saeb erekat
cali	martinez romero	gravano	shining path
car bomb	medellin	hamas	sicilian mafia
cocaine cartel	miguel maza	hezbollah	sinn fein
cosa nostra	muammer gadaffi	ira gunman	suicide bomber
drug baron	nicola mancino	ira gunmen	toto riina
drug cartel	pablo escobar	islamic jihad	drug lords

Table 8 -- Topic 301

paralytic polio	polio myelitis	polio vaccines
polio	polio outbreak	polio virus
polio cases	polio type	poliomyelitis
polio epidemic	polio vaccine	poliovirus

Table 9 -- Topic 302

Retrieval Level	Run 1 Precision	Run 2 Precision
at 5 docs	0.3280	0.0680
at 10 docs	0.3000	0.0580
at 15 docs	0.3080	0.0680
at 20 docs	0.2980	0.0710
at 30 docs	0.2720	0.0640
at 100 docs	0.1938	0.0492
at 200 docs	0.1347	0.0421
at 500 docs	0.0748	0.0321
at 1000 docs	0.0446	0.0231

Table 10 -- Comparing Intersection and Union runs

5. Conclusions and Future Work

For TREC-6, we focused on improving relevance feedback using the relational model. While the changes in our relevance feedback process significantly improved the precision and recall scores of our results, we still need to look into improved methods of choosing the feedback terms to eliminate the "bad" terms which occasionally surface for some of the queries. Another area we have begun to investigate is raising the precision and recall scores of the baseline run prior to relevance feedback. One of the methods we have found to do this involves the use of term frequency cutoff points and additional work needs to be done to further investigate the relationship between the query-to-document scores and the term weights of the infrequently occurring terms.

For our manual runs, we focused on using new methods such as Soundex and an improvement formula based on market basket analysis to identify query expansion terms. Further work needs to be done to better identify the appropriate query expansion terms.

References:

(Ballerini96) Ballerini, J., M. Buchel, D. Knaus, B. Mateev, E. Mittendorf, P. Schauble, P. Sheridan, and M. Wechsler, " SPIDER Retrieval System at TREC-5," Proceedings of the Fifth Text REtrieval Conference (TREC), sponsored by the National Institute of Standards and Technology and the Advanced Research Projects Agency, November 1996.

(Berry97) Berry, M., and G. Linoff, "Data Mining Techniques," Wiley Computer Publishing, 1997.

(Buckley95) Buckley, C. A. Singhal, M. Mitra, and G. Salton, "New Retrieval Approaches Using SMART: TREC-4," sponsored by the National Institute of Standards and Technology and the Advanced Research Projects Agency, November 1995.

(Celko95) Celko, J., "SQL for Smarties: Advanced SQL Programming," Morgan Kaufmann, 1995.

(Grossman95) Grossman, D., D. Holmes, O. Frieder, M. Nguyen, and C. Kingsbury, "Improving Accuracy and Run-Time Performance for TREC-4," Proceedings of the Fourth Text REtrieval Conference (TREC), sponsored by the National Institute of Standards and Technology and the Advanced Research Projects Agency, November 1995.

(Grossman96) Grossman, D., C. Lundquist, J. Reichert, D. Holmes, and O. Frieder, "Using Relevance Feedback within the Relational Model for TREC-5," Proceedings of the Fifth Text REtrieval Conference (TREC), sponsored by the National Institute of Standards and Technology and the Advanced Research Projects Agency, November 1996.

(Grossmam97) Grossman, D., D. Holmes, O. Frieder, and D. Roberts, " Integrating Structured Data and Text: A Relational Approach," *Journal of the American Society of Information Science*, January 1997.

(Lundquist97a) Lundquist, C., D. Grossman, O. Frieder, and D. Holmes, "A Parallel Implementation of Relevance Feedback using the Relational Model," *Proceedings of the World Multiconference on Systemics, Cybernetics, and Informatics*, July 1997.

(Lundquist97b) Lundquist, C., D. Grossman, and O. Frieder, "Improving Relevance Feedback in the Vector-Space Model," to appear in Proceedings of the Sixth ACM International Conference on Information and Knowledge Management, 1997.

(Pfeifer96) Pfeifer, U., T. Poersch, and N. Fuhr, "Retrieval Effectiveness of Proper Name Searches",*Information Processing and Management*, Vol. 32, No. 6, pp. 667-679.

(Rocchio71) Rocchio, Jr., J. J., "Relevance Feedback in Information Retrieval," Gerard Salton, Editor, The SMART Retrieval System, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1971.

(Singhal96) Singhal, A., C. Buckley, and M. Mitra, "Pivoted Document Length Normalization," *Proceedings of the Nineteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Ed. Hans-Peter Frei, Donna Harman, Peter Schauble and Ross Wilkinson, SIGIR Forum, August 18-22, 1996.