Document Retrieval Using
The MPS Information Server
(A Report on the TREC-5 Experiment)

François Schiettecatte
(francois@fsconsult.com)
FS Consulting, Inc.
1890 Highland Avenue, Rochester, NY, 14618
http://www.fsconsult.com/

1 Introduction

This paper summarizes the results of the experiments conducted by FS Consulting, Inc. as part of the Fifth Text Retrieval Experiment Conference (TREC-5). We participated in Category C, ran the ad-hoc experiments and participated in the database merging track, producing three sets of official results (fsclt3, fsclt4 and fsclt3m) as well as some unofficial results (fsclt4a). Our long-term research interest is in building information retrieval systems that help users find information to solve real-world problems. Our TREC-5 participation centered on two goals: to see if automatic query reformulation (relevance feedback) provides better results than the searcher's query reformulation; and to evaluate the effectiveness of the document scoring algorithms when searching across multiple databases. Our TREC-5 ad-hoc experiments were designed around a model of an experienced end user of information systems, one who might regularly use a system like the MPS Information Server while seeking information in a workplace or library setting

2 Overview of FS Consulting TREC-5 Experiments

In the TREC-5 experiments we set out to answer two questions:

- Does automatic query reformulation (relevance feedback) provides better results than the searcher's query reformulation?

We began with the assumption that our information seeker had previous experience using on-line retrieval systems commonly available in libraries (e.g., on-line library catalogs and bibliographic systems like MEDLINE). Although the search interface varies considerably, most library systems default to a novice-type search interface that allows a searcher to enter one or two terms and apply a Boolean operator to relate them. To aid the more advanced end-user/searcher, academic libraries typically provide search aids and training sessions that teach searchers to construct more complicated Boolean statements and employ controlled vocabularies for term selection. Additionally, information seekers who patronize librarian-mediated search services have observational experience to draw upon: often, they are asked to create query statements, or to work directly with librarians performing the search as they construct and revise search statements. It is this combination of context and experience that constitutes the background knowledge of our experienced searcher.

For our first experiment, we did three runs (fsclt3, fsclt4 and fsclt4a). We manually constructed queries for all the topics. The searcher was permitted to employ any number of terms into simple or complex Boolean arguments. For this experiment, a single user query was entered for each topic, and a relevance ranked output was generated for each, using standard system features of the MPS Information Server. Once the queries were constructed, they were run producing a first set of results (fsclt3). We then allowed the searcher to review the ranked output and select up to two documents which would be applied as relevance feedback to the search (fsclt4). We then did a third run where the system automatically selected the first two documents from the initial set of results (fsclt3) and applied them as relevance feedback to the search to produce a new set of results (fsclt4a).

- What is the effectiveness of the document scoring algorithm when searching across multiple databases?

For our second experiment, the TREC-5 corpus was split and indexed into 100 separate databases as defined by the database merging track. We then took the manually constructed queries from the first experiment and ran them across the databases, merging the results into a single ranked list (fsclt3m). We then compared these results to the baseline run fsclt3.

The ranking algorithm was initially developed, tested and refined using the TREC-4 corpus and database merging track guidelines set out in TREC-4.

3 Searcher Model and Guidelines

Because all the runs employed the same query formulations, the same searcher model and guidelines apply to all of them. All query statements for the experiments were constructed by one person. The initial parameters of the searcher 'model' were defined as follows:

- s/he regularly searches on-line catalogs and bibliographic databases in an academic setting;

- s/he may have some search training, but is not a professional searcher;
- s/he dislikes reviewing large search outputs;
- s/he is seeking information to solve a real-life problem;
- s/he may not be a content expert in the topic area of a given question.

3.1 Instructions to Searcher

The following instructions guided query formulation:

- prepare a single search statement that will capture the most relevant documents for a given topic;
- use single or multiple terms, employing wild card capability to capture multiple versions of a word, and/or quotes around several words (e.g., "cardiac arrest") to create a fixed phrase;
- apply boolean logic as desired, using AND, OR or NOT operators. Create nested statements using parentheses if desired;
- consult the stop word list if needed, but no other databases are available for consultation;
- the total time taken to prepare a single query should not exceed 5 minutes.

3.2 Searcher Training

In preparation for the experiment, the searcher performed training exercises using the TREC-5 training data. First, general capabilities of the system and features of the search engine were described. Then, three topics were selected from the TREC-4 topics by the searcher. For each topic, a query formulation constructed by the searcher was run against the test database. Results were analyzed using the Trec Eval program. Lists of document headlines were provided to the searcher for examination. The searcher was allowed to reformulate and re-run training queries as many times as she desired. The search interface for training exercises employed a custom client application created for this experiment and MacWais, a freely available WAIS client for the Macintosh.

4 System Configuration

The MPS Information Server is a commercial full-text retrieval system that runs on a large number of Unix based platforms. Given a user query, the MPS system returns a list of relevance-ranked documents from a database. The system is capable of performing simple or complex term or phrase searching using parentheses, wildcards and Boolean operators. Soundex, typographical variation (for example, missing letters, 'color' would also pick up 'colour', and juxtaposition of letters, 'animal' would also pick up 'ainmal') and fielded searching are also supported. The system is designed to favor precision over recall when performing searches. Because it supports a number of different protocols, including WAIS-88, Z39.50-V2, HTTP, CGI-BIN, Gopher+ as well as two internal protocols, the MPS Information Server is capable of responding to search requests from a wide variety of clients applications.

The TREC-5 experiments employed version 3.1 of the MPS Information Server running on a SparcStation 5/110. Four gigabytes of disk space were set aside and split evenly between data and indices. For the purposes of the experiments, we built a custom client application. Running on the SparcStation, the custom client communicated with the MPS Information Server using the WAIS-88 protocol. This client application was designed to read TREC topic files, build a query by extracting a specific field (or fields) from the individual topic entries, run the queries against the server and save the query results in the TREC result format to a specified file. The results files could then be processed by the Trec Eval program to obtain the precision-recall values for the run.

A special parser was built to index the TREC database. We created an artificial headline for each document by concatenating the document ID field (the <DOCNO>) and the document title (where a title was available). We then suppressed a number of fields judged to be 'noise' based on inspection of sample documents from the various sources (a complete list of the field suppressed is provided in appendix A). The rest of the documents were indexed as plain text, with the SGML tags extracted from the text, and the words stemmed using a plural stemmer. No additional information was extracted apart from the word positions to allow phrase and proximity searching if desired by the searcher. All keywords in the news articles were suppressed, as required by the instructions.

While the MPS system's indexer starts up with an initial stop-word list (containing 377 words), it can choose to convert a word to a stop word if that word's total occurrence in the database reaches a specified value. The stop word value, which is site- and collection-dependent, would typically be set anywhere in the range of 20,000 to 50,000 occurrences. For the TREC-5 experiments, it was set at 150,000 occurrences to retain as many words as possible in the database. This resulted in a final stop-word list of 446 words.

Two databases were created, one containing disks 2 and 3 (the ad-hoc training database) and the other containing disks 2 and 4 (the ad-hoc test database). Each database took approximately 8 hours to build; their index sizes were about 530 MB each (from an initial size of 1.9GB for the TREC-4 corpus, and 1.6GB for the TREC-5 corpus).

5 TREC-5 Results for FS Consulting Experiments

5.1 First Experiment

In the first experiment, the searcher initially created a single written query for each of 50 ad hoc topics. The queries were then run producing a first set of results (fsclt3). The searcher was then allowed to review the ranked output and select up to two documents from the initial set of results which would be applied as relevance feedback to the search (fsclt4). We then did a third run where the system automatically selected the first two documents from the initial set of results and applied them as relevance feedback to the search (fsclt4a).

The relevance feedback algorithm employed for query expansion in this experiment works by ranking all terms in selected documents by frequency of occurrence. The ten most 'interesting' terms were chosen from that list for further use. The most 'interesting' terms were defined as neither the most frequent or infrequent terms. Rather, frequency parameters were specified to eliminate the high and low ends of the spectrum. The experiment's final result sets were produced by expanding each original query to include new terms, assigning weights to the old and new terms, and re-ordering documents based on new relevance weights.

This automated query expansion feature was designed as a tool that could be used by information seekers who, having retrieved a large set from an initial search, wish to increase the likelihood that all relevant documents retrieved were listed in the first 30 or 40 titles in the result set.

5.1.1 Searcher Performance

Training exercises influenced the searcher's query formulation behavior in the following ways:

- she preferred to use the wild-card capability to increase recall, rather than entering multiple forms of a word;
- she was reluctant to add long lists of multiple synonyms, believing that they would dilute search results;
- she tried work-arounds to avoid the stop-list for common words like 'states' and 'united' (e.g., using "United States" as a bounded phrase).

The following examples are typical formulations used for fsclt3. The searcher wrote out the formulations, which were entered into the system by the researcher (FS) without further discussion or modification.

Topic 252: "illegal alien" OR "illegal immigra*"

Topic 259: ((John OR Jack) AND Kennedy) OR JFK) AND assassin*)

Topic 262: "seasonal affective disorder" OR SADS OR (seasonal AND disorder)

Topic 271: "solar power" OR "solar energy"

Most query formulations for fsclt3 employed parentheses, wildcards and the AND and OR Boolean operators. As the examples indicate, not all capabilities of the system were employed (e.g., field searching, soundex, typographical variation and "NOT" operator were not used for example). The bounded phrase was the most common special feature used. Missing parentheses in the examples above suggest that logic statements were not always constructed properly.

5.1.2 Server Performance

The results for fsclt3 produced the following precision/recall figures over all of the topics:

```
Queryid (Num):       50topics  fsclt3
Total number of documents over all queries
    Retrieved:    30987
    Relevant:      5524
    Rel_ret:       1866
Interpolated Recall - Precision Averages:
    at 0.00       0.5178
    at 0.10       0.3371
    at 0.20       0.2410
    at 0.30       0.1830
    at 0.40       0.1297
    at 0.50       0.1106
    at 0.60       0.0899
    at 0.70       0.0554
    at 0.80       0.0375
    at 0.90       0.0201
    at 1.00       0.0201
Average precision (non-interpolated) over all rel docs
                  0.1368
Precision:
  At    5 docs:   0.2960
  At   10 docs:   0.2720
  At   15 docs:   0.2627
  At   20 docs:   0.2510
  At   30 docs:   0.2333
  At  100 docs:   0.1584
  At  200 docs:   0.1120
  At  500 docs:   0.0631
  At 1000 docs:   0.0373
R-Precision (precision after R (= num_rel for a query) docs
retrieved):
    Exact:        0.1870
```

Overall, for all topics, 34% of the relevant documents were retrieved from the database and only 6% of the documents retrieved were relevant.

The results for fsclt4 produced the following precision/recall figures over all of the topics:

```
Queryid (Num):       50topics  fsclt4
Total number of documents over all queries
    Retrieved:    50000
    Relevant:      5524
    Rel_ret:       2159
Interpolated Recall - Precision Averages:
    at 0.00       0.5757
    at 0.10       0.3632
    at 0.20       0.2824
    at 0.30       0.2147
    at 0.40       0.1599
    at 0.50       0.1297
    at 0.60       0.0918
    at 0.70       0.0660
    at 0.80       0.0385
    at 0.90       0.0241
    at 1.00       0.0187
Average precision (non-interpolated) over all rel docs
                  0.1559
Precision:
  At    5 docs:   0.3320
  At   10 docs:   0.3080
  At   15 docs:   0.2813
  At   20 docs:   0.2600
  At   30 docs:   0.2393
  At  100 docs:   0.1644
  At  200 docs:   0.1159
  At  500 docs:   0.0685
  At 1000 docs:   0.0432
R-Precision (precision after R (= num_rel for a query) docs
retrieved):
    Exact:        0.2006
```

Overall, for all topics, 39% of the relevant documents were retrieved from the database and just over 4% of the documents retrieved were relevant.

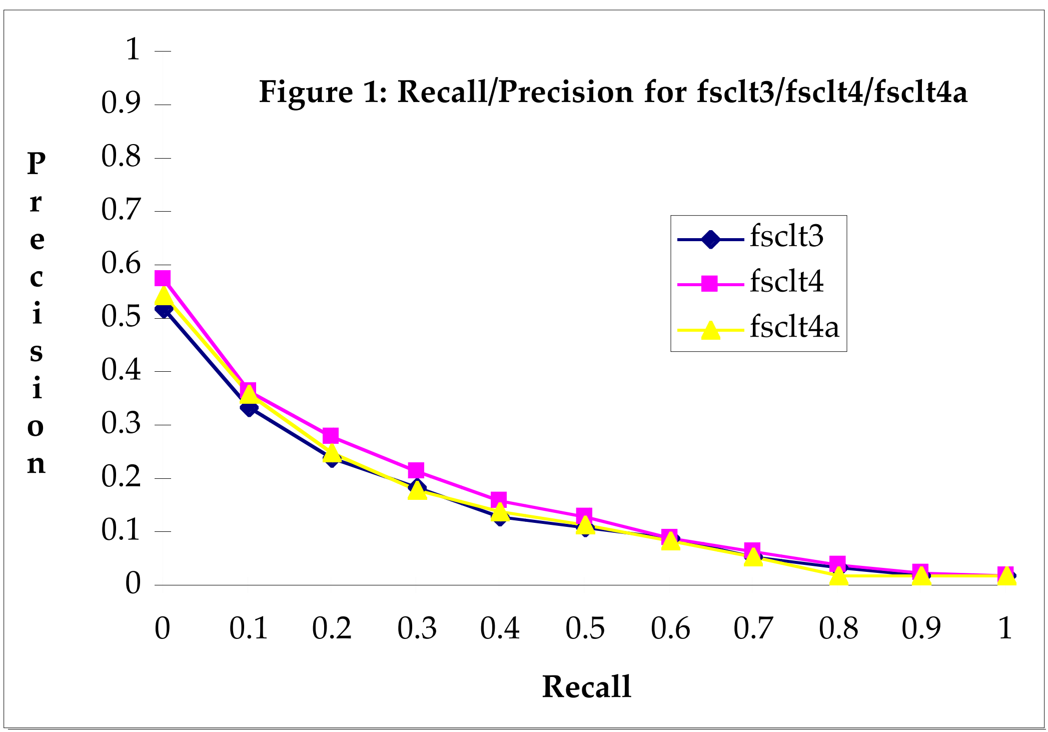
The results for fsclt4m produced the following precision/recall figures over all of the topics:

```
Queryid (Num):       50
Total number of documents over all queries
    Retrieved:    50000
    Relevant:      5524
    Rel_ret:       2047
Interpolated Recall - Precision Averages:
    at 0.00       0.5439
    at 0.10       0.3605
    at 0.20       0.2502
    at 0.30       0.1812
```

```
    at 0.40       0.1381
    at 0.50       0.1130
    at 0.60       0.0830
    at 0.70       0.0561
    at 0.80       0.0225
    at 0.90       0.0194
    at 1.00       0.0189
Average precision (non-interpolated) over all rel docs
                  0.1415
Precision:
  At    5 docs:   0.3120
  At   10 docs:   0.3040
  At   15 docs:   0.2787
  At   20 docs:   0.2640
  At   30 docs:   0.2400
  At  100 docs:   0.1588
  At  200 docs:   0.1101
  At  500 docs:   0.0655
  At 1000 docs:   0.0409
R-Precision (precision after R (= num_rel for a query) docs
retrieved):
    Exact:        0.1993
```

Overall, for all topics, 37% of the relevant documents were retrieved from the database and just over 4% of the documents retrieved were relevant.

Figure 1 below show the precision-recall curve for both fsclt3, fsclt4 and fsclt4a.



These results would seem to suggest that using automatic relevance feedback would produces better results than not using it, but that letting the user choose which documents to apply as relevance feedback is still the best policy.

When comparing these results with our TREC-4 results [1], two differences jump out: the first one is that these runs returned a lot more documents overall as well as more relevant documents (better recall); the other is that the precision was higher in the TREC-4 results. This is in line with the design of the search engine which is to favor precision against recall, so when fewer documents are retrieved, precision increases.

5.2 Second Experiment

The second experiment was to measure the effectiveness of the document scoring algorithms when searching across multiple databases (database merging track). The TREC-5 corpus was split and indexed into 100 separate databases as defined by the database merging track. We then took the manually constructed queries from the first experiment and ran them across the databases, merging the results into a single ranked list (fsclt3m). We then compared these results to the baseline run fsclt3.

5.2.1 Server Performance

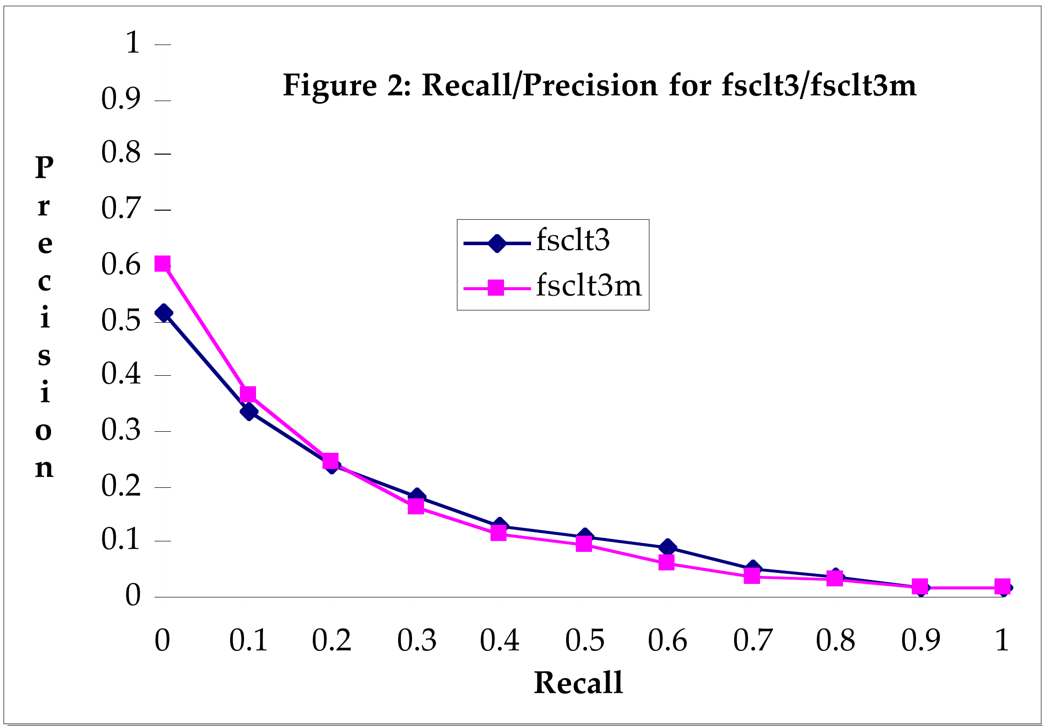
The results for fsclt3m produced the following precision/recall figures for all the topics:

```
Queryid (Num):       all  fsclt3m
Total number of documents over all queries
    Retrieved:    31189
    Relevant:      5524
    Rel_ret:       1389
Interpolated Recall - Precision Averages:
    at 0.00       0.6019
    at 0.10       0.3694
    at 0.20       0.2458
    at 0.30       0.1640
    at 0.40       0.1182
    at 0.50       0.0960
    at 0.60       0.0612
    at 0.70       0.0396
    at 0.80       0.0337
    at 0.90       0.0197
    at 1.00       0.0197
Average precision (non-interpolated) over all rel docs
                  0.1354
Precision:
  At    5 docs:   0.3720
  At   10 docs:   0.3240
  At   15 docs:   0.3013
  At   20 docs:   0.2760
```

```
  At   30 docs:   0.2467
  At  100 docs:   0.1606
  At  200 docs:   0.1063
  At  500 docs:   0.0531
  At 1000 docs:   0.0278
R-Precision (precision after R (= num_rel for a query) docs
retrieved):
    Exact:        0.1890
```

Overall, for all topics, 25% of the relevant documents were retrieved from the database and only 4.5% of the documents retrieved were relevant.

Figure 2 below show the precision-recall curve for fsclt3 and fsclt3m.



What is interesting to note is that while this run (fsclt3m) retrieved fewer relevant documents than fsclt3, precision is higher. This would seem to suggested that the document scoring algorithms works well when searching across databases.

While working on the document scoring algorithm we also ran the TREC-4 database merging track and got results very similar to those above whether the corpus was segmented into two databases or eight databases (as was required by the track). We repeated this experiment with the TREC-5 data and got the same results as well.

The implication of this is that we can segment a very large database across a number of machines to take advantage of parallelization (the MPS search engine can searches multiple databases in parallel) and be able to present the user with a single, meaningfully ranked, results set. In addition one would also gain in terms of system redundancy where portions of the database would still be available for searching if one of the machines was down for repairs or maintenance.

6 Discussion of FS Consulting TREC-5 Results

The MPS search engine is designed to operate in an interactive setting, where quick response and high precision are generally preferable to high recall. (High recall can be achieved by creating several different queries for the same topic; this is the recommended search strategy when high recall searches are required). Comparing the TREC-5 results with our TREC-4 [1] results really illustrates this. While our TREC-4 results returned less documents overall, the precision was higher.

6.1 Searcher improvements

Examination of the query formulations indicated that the searcher did not take full advantage of system features, and constructed queries that were not optimal for the search system. For example, the wildcard feature in a lot of cases, thereby increasing recall at the expense of precision. Whether or not these formulations are typical of average searchers, it seems clear that the training period did not produce sufficient understanding of the system's strengths and weaknesses, nor were optimal query models presented and reinforced.

6.2 System improvements

While the relevance feedback algorithms work adequately at this point, it is hard not to want better performance from them. In that light we will be running a number of experiments this year prior to TREC-6 to fine-tune the relevance feedback algorithms further.

7 Future Work

TREC-5 experiments provided baseline results and in a non-interactive environment and allowed exploration of possible directions for future work. Several themes emerged that will guide our research efforts in preparation for participation in TREC-6.

- The system will be tuned and improved. The query expansion tool will continue to be tested and revised. Additional relevance feedback algorithms will also be tested. A user interface will be constructed to allow the searcher to review results and make other decisions about search parameters.

- Additional examples of manual query formulations will be gathered and tested for TREC-5 topics, in order to build and improve the model of an 'average' searcher. Efforts will be made to gather formulations from searchers with different backgrounds (e.g., librarians, medical students, academic faculty, administrative and clerical staff). These data will be used to improve searcher training, and to suggest additional user tools.

- An interactive experiment will be designed based on improvements and searcher modeling to be undertaken this year.

8 Acknowledgments

The author wishes to thank Valerie Florance for her invaluable help with the design of this experiment and in the creation of the queries.

9 References

[1] Schiettecatte, François and Florance, Valerie, "Document Retrieval Using the MPS Information Server". In Harman D. (Ed) The Fourth Text Retrieval Conference (TREC-4). National Institute of Standards and Technology Special Publication 500-236, Gaithersburg, Md. 20899.

Appendix A

The following fields were suppressed at indexing time:

AP Newswire:
- <DOCNO>
- <FILEID>
- <FIRST>
- <SECOND>
- <DATELINE>
- <BYLINE>
- <NOTE>
- <UNK>

Congressional Record:
- <DOCNO>
- <DOCID>
- <CENTER>
- <DATE>
- <FLD001>
- <FLD002>
- <FLD003>
- <SO>

Federal Register 88:
- <DOCNO>
- <DOCID>

Federal Register 94:
- <DOCNO>

Financial Times:
- <DOCNO>
- <PROFILE>
- <DATE>
- <PUB>
- <PAGE>

Wall Street Journal:
- <DOCNO>
- <DD>
- <SO>
- <IN>
- <DATELINE>
- <G>
- <GV>
- <AN>
- <RE>
- <MS>
- <NS>
- <DATE>
- <DO>
- <ST>

Ziff:
- <DOCNO>
- <DOCID>
- <DESCRIPT>
- <JOURNAL>
- <AUTHOR>

Patents:
- <DOCNO>
- <WKU>
- <SRC>
- <APN>
- <APT>
- <ART>
- <APD>
- <ISD>

SanJose Mercury
- <DOCNO>
- <ACCESS>
- <CAPTION>
- <DESCRIPT>
- <SECTION>
- <MEMO>
- <BYLINE>
- <COUNTRY>
- <CITY>
- <EDITION>
- <CODE>
- <NAME>
- <PUBDATE>
- <DAY>
- <MONTH>
- <PG.COL>
- <PUBYEAR>
- <REGION>
- <STATE>
- <WORD.CT>
- <DATELINE>
- <COPYRGHT>
- <LIMLEN>
- <LANGUAGE>