Document Retrieval Using The MPS Information Server (A Report on the TREC-6 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 Six Text Retrieval Experiment Conference (TREC-6). We participated in Category C and ran the ad-hoc experiments, producing three sets of official results (fsclt6, fsclt6r and fsclt6t), only one of which was judged (fsclt6). We also produced two sets of unofficial results as part of a database merging experiment that we ran (fsclt6m2 and fsclt6m5).

Our long-term research interest is in building information retrieval systems that help users find information to solve real-world problems. Our TREC-6 participation centered on two goals: to see if automatic query reformulation[1] provides better results than the searcher's initial query formulation; and to continue to evaluate the effectiveness of the document scoring algorithms when searching across multiple databases located on multiple servers.

Our TREC-6 ad-hoc experiments were designed around a model of an end user of information systems who is not a search professional, but one who would occasionally use a system like the MPS Information Server while seeking information in a workplace, or would be familiar with various Internet search engines such as HotBot or AltaVista.

2 Overview of FS Consulting TREC-6 Experiments

In the TREC-6 experiments we set out to answer two questions:

- Does automatic query reformulation provides better results than the searcher's initial query formulation?

We began with the assumption that our information seeker had previous experience using the MPS Information Server and/or various Internet search engines such as HotBot or AltaVista. Although the search interfaces to these systems vary considerably, most systems default to a novice-type search interface[2] that allows a searcher to enter a number of terms and, optionally, apply some sort of operator to relate these terms together such as Boolean operators or phrase searching. To aid the more advanced end-user/searcher, most systems provide a more advanced search interface allowing the user to construct more complex queries as well as some advanced query constructs such as Boolean operators, phrase searching, proximity searching and range

[1] Using relevance feedback.
[2] Usually a single search field.

searching[3]. Typically users don't use these more advanced features, being content to use the novice-type search interface and looking at the first screen of results for a satisfying document[4].

For our first experiment, we did three runs (fsclt6, fsclt6r and fsclt6t), only one of which was judged (fsclt6). We manually constructed queries for all the topics. The searcher was permitted to employ any number of terms along with any search feature offered by the search engine. 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 (fsclt6). We then did a second run where the system automatically selected the first two documents from the initial set of results (fsclt6) and applied them as relevance feedback to the search to produce a new set of results (fsclt6r). We did a third run (fsclt6t) using the topic titles to compare the results to the first run (fsclt6).

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

For our second experiment, the TREC-6 corpus was split and indexed into two separate databases and into five separate databases. 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 (fsclt6m2 and fsclt6m5). We then compared these results to the baseline run (fsclt6).

The ranking algorithm was initially developed, tested and refined using the corpus and database merging tracks guidelines set out in TREC-4, and were tested more fully in the TREC-5 database merging track.

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 occasionally uses Internet search engines;
- s/he occasionally uses search engines in the work 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:

[3] Such as 'since a certain date' and 'before a certain date', or 'within the last week' or 'within the last month', for example.

[4] It was mentioned in a SIGIR 97 session by Doug Cutting of Excite that users provide an average of 2 terms per search and that only 20% of users request that the second screen of results be displayed after looking at the first screen of results (each screen displays 20 hits). This means that 80% of users either find the document they are looking for in the first screen of results or give up the search.

- 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;
- 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-6 training data. First, general capabilities of the system and features of the search engine were described. Then, three topics were selected from the TREC-5 topics by the searcher. For each topic, a query formulation constructed by the searcher was run against the test database in an interactive fashion via a Web based interface. Results were also analyzed using the Trec Eval program. The searcher was allowed to reformulate and re-run training queries as many times as he desired.

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 searching, phrase searching and proximity searching using parentheses, wildcards and Boolean operators. Soundex, typographical variation [5]. Fielded searches and numerical range searches 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 (WAIS-V2 profile), STARTS and Gopher as well as two internal protocols, LWPS and Direct [6], the MPS Information Server is capable of responding to search requests from a wide variety of clients applications.

The TREC-6 experiments employed version 4.2 of the MPS Information Server running on a SparcStation 5/110 with 128 megabytes of RAM. Four gigabytes of disk space were set aside and split evenly between data and indices. For the purposes of the experiments, we used a driver application which was built for TREC-4 and was used in TREC-5. Running on the SparcStation, the driver application communicated with the MPS Information Server using the LWPS protocol. This driver application was designed to read TREC topic files, build a query by extracting a specific field, or fields, from the individual topic entries [7], run the queries against the MPS Information 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 that run.

[5] For example, missing letters, 'color' would also pick up 'colour', and juxtaposition of letters, 'animal' would also pick up 'ainmal'.
[6] LWPS is an inter-server communications protocol and Direct is a protocol which allows for rapid integration into front-end development application tools such as Perl or Tk/Tcl.
[7] In fact the queries formulated by the user are embedded into the TREC topic file and are marked up with SGML tags .

A special parser was built to index TREC databases for the TREC-4 & TREC-5 experiments and was reused, with a few modifications, for the TREC-6 experiments. We created an artificial headline for each documentby concatenating the documentID field [8] and the documenttitle [9]. The rest of the documentswere 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 from the text except for 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 guidelines.

While the MPS Information Server's indexing application starts up with a default stop-word list (containing 377 words), it can be made to convert a word to a stop word if that word's total occurrencein the database reaches a specific threshold value. The stop word value, which is site- and collection-dependent, would typically be set anywhere in the range of 20,000 to 500,000 occurrences.For the TREC-6 experiments, it was set at 500,000 occurrencesto retain as many words as possible in the database. This resulted in a final stop-word list of 382 words.

Two databases were created, one containing disks 2 and 4 (the ad-hoc training database) and the other containing disks 4 and 5 (the ad-hoc test database). Each database took approximately 8 hours to build. Their index sizes were about 540 megabytes for disks 2 and 4, and 610 megabytes for disks 4 and 5, from an initial data size of 2.0 gigabytes and 2.1 gigabytes respectively. In addition seven other databases were created for the database merging experiments from disks 4 and 5.

5 TREC-6 Results for FS Consulting Experiments

5.1 First Experiment

In the first experiment, the searcher initially created a single written query for each of the 50 ad-hoc topics. The queries were then runproducinga first set of results (fsclt6). We then did a secondrunwhere the system automatically selected the top two documentsfrom the initial ranked set of results and applied them as relevance feedback to the search (fsclt6r).

The relevance feedback algorithm employed for query expansion used for the secondrunis the standard one implemented in the MPS Server. It works by scoring all the terms in the selected documents.The top twenty terms were chosen from that list for further use. This run's final result sets were produced by expanding each original query to include these new terms, assigning weights to the old and new terms, and re-ordering documentsbased on new relevance weights. The relevance feedback can be set to either increase recall or improve precision and the former setting was chosen for these experiments.

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 documentsretrieved were listed in the first 30 or 40 titles in the result set.

---

[8] The <DOCNO> </DOCNO> field.

[9] Where a title was easily identifiable and available from the document, this was the usually the case for news data, but less so for other data.

5.1.1 Searcher Performance

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

- he preferred to use the wild-card capability selectively to increase recall, rather than entering multiple forms of a word and diluting the precision of the query;
- he added multiple synonyms, believing that it would increase recall in a selective fashion;
- he kept the queries short;

The following examples are typical formulations. The searcher wrote out the formulations, which were embedded into the TREC topic file without further modification.

Topic 332: "united states" AND "tax evasion" AND investigations

Topic 338: aspirin AND (adverse OR risks)

Topic 344: (email e-mail "electronic mail") AND (abus* spam*)

Topic 349: catabolic anabolic metabolic metabolism glycolysis krebs

Most query formulations 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, proximity searching, soundex, typographical variation and "NOT" operators were not used for example). The bounded phrase was the most common special feature used.

5.1.2 Server Performance

The results for fsclt6 produced the following precision/recall figures over all of the topics:

```
Queryid (Num):       all  fsclt6
Total number of documents over all queries
    Retrieved:    12119
    Relevant:      4611
    Rel_ret:       1300
Interpolated Recall - Precision Averages:
    at 0.00       0.7064
    at 0.10       0.4472
    at 0.20       0.3356
    at 0.30       0.2034
    at 0.40       0.1581
    at 0.50       0.1255
    at 0.60       0.0719
    at 0.70       0.0370
    at 0.80       0.0333
    at 0.90       0.0167
    at 1.00       0.0167
Average precision (non-interpolated) over all rel docs
                  0.1691
Precision:
  At    5 docs:   0.4520
  At   10 docs:   0.3900
  At   15 docs:   0.3467
  At   20 docs:   0.3120
  At   30 docs:   0.2680
  At  100 docs:   0.1642
  At  200 docs:   0.1037
  At  500 docs:   0.0500
  At 1000 docs:   0.0260
```

```
R-Precision (precision after R (= num_rel for a query) docs retrieved):
    Exact:        0.2173
```

Overall, for all topics, 28% of the relevant documentswere retrieved from the database and only 11% of the documents retrieved were relevant.

The results for fsclt6r produced the following precision/recall figures over all of the topics:

```
Queryid (Num):       all  fsclt6r
Total number of documents over all queries
    Retrieved:    49001
    Relevant:      4611
    Rel_ret:       1659
Interpolated Recall - Precision Averages:
    at 0.00       0.6847
    at 0.10       0.4703
    at 0.20       0.3571
    at 0.30       0.2414
    at 0.40       0.1665
    at 0.50       0.1040
    at 0.60       0.0436
    at 0.70       0.0239
    at 0.80       0.0040
    at 0.90       0.0000
    at 1.00       0.0000
Average precision (non-interpolated) over all rel docs
                  0.1660
Precision:
  At    5 docs:   0.4280
  At   10 docs:   0.3800
  At   15 docs:   0.3400
  At   20 docs:   0.3150
  At   30 docs:   0.2800
  At  100 docs:   0.1682
  At  200 docs:   0.1121
  At  500 docs:   0.0570
  At 1000 docs:   0.0332
R-Precision (precision after R (= num_rel for a query) docs retrieved):
    Exact:        0.2142
```

Overall, for all topics, 36% of the relevant documentswere retrieved from the database and only 3.4% of the documentsretrieved were relevant. This last figure is much lower than the same figure for the previous run (fsclt6) because there were many more documentsretrieved in this run (49,001 documents) compared to the previous run (12,119 documents). This increase in the numberof documentsis due to the relevance feedback feature being set to increase recall rather than precision.

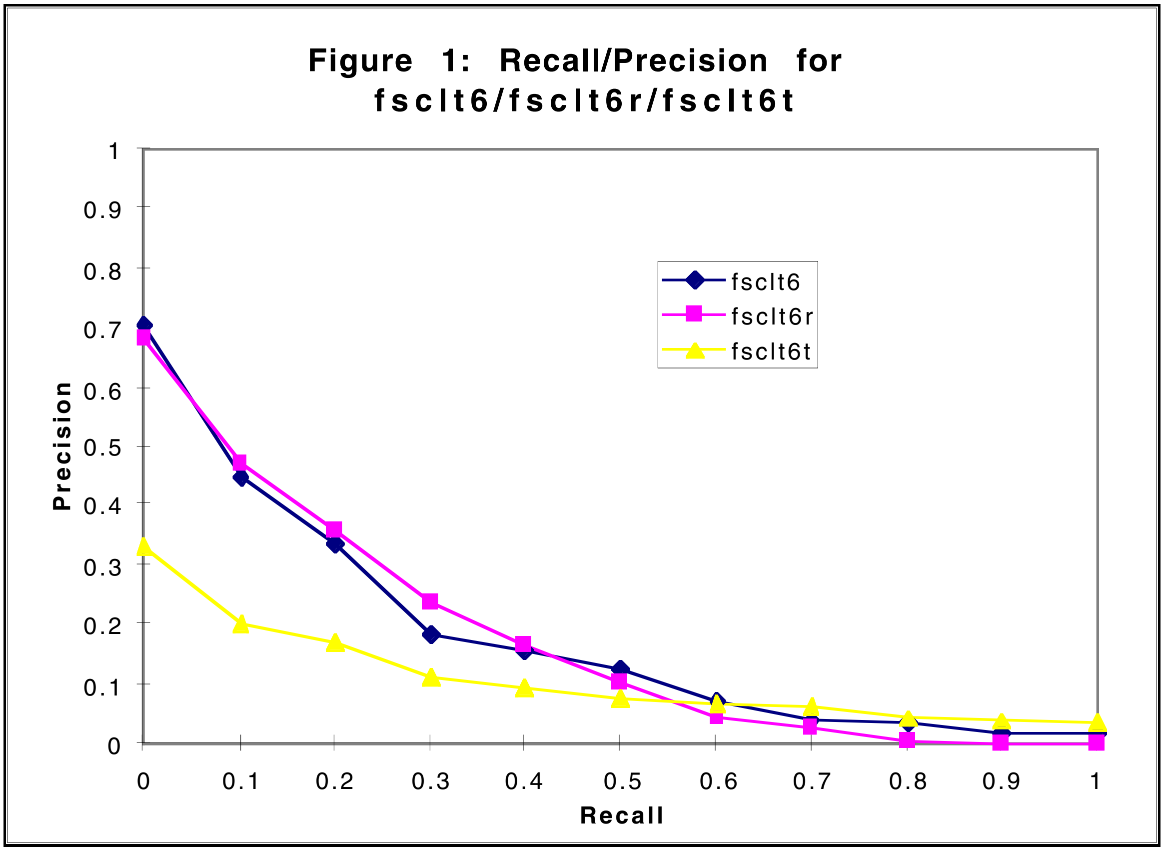
The results for fsclt6t produced the following precision/recall figures over all of the topics:

```
Queryid (Num):       all  fsclt6t
Total number of documents over all queries
    Retrieved:    43156
    Relevant:      4611
    Rel_ret:       1172
Interpolated Recall - Precision Averages:
    at 0.00       0.3340
    at 0.10       0.1992
    at 0.20       0.1718
    at 0.30       0.1125
    at 0.40       0.0943
```

```
     at 0.50       0.0756
     at 0.60       0.0670
     at 0.70       0.0620
     at 0.80       0.0410
     at 0.90       0.0367
     at 1.00       0.0334
Average precision (non-interpolated) over all rel docs
                   0.0958
Precision:
  At    5 docs:   0.1720
  At   10 docs:   0.1680
  At   15 docs:   0.1533
  At   20 docs:   0.1430
  At   30 docs:   0.1233
  At  100 docs:   0.0752
  At  200 docs:   0.0531
  At  500 docs:   0.0336
  At 1000 docs:   0.0234
R-Precision (precision after R (= num_rel for a query) docs retrieved):
    Exact:         0.1279
```

Overall, for all topics, 25% of the relevant documents were retrieved from the database and just under 3% of the documentsretrieved were relevant. This run is considerably worse than the previous two run strongly suggesting that merely using the topic title as a search string is not a good approach for this search engine.

Figure 1 below show the precision-recall curve for fsclt6, fsclt6r and fsclt6t.



These results would seem to suggest that using automatic relevance feedback would produces better results than not using it, and it appears to improve recall at the expense of some precision.

When comparing these results with our TREC-5 results [2], two differences jump out: the first one is that these runs returned fewer documents overall as well as proportionally fewer relevant documents; the other is that the precision was higher in these results than in the TREC-5 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. The TREC-6 corpus was split and indexed into 2 separate databases (disks 4 and 5) and into 5 separate databases (cr, fbis, fr94, ft and latimes). 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 (fsclt6m2 and fsclt6m5). We then compared these results to the baseline run (fsclt6).

It should be noted that these databases were located behind individual MPS Information Servers which were accessed by the MPS Information Server Gateway. This gateway presented the various physical databases as a single logical database to the driver application. The driver application was unaware of the fact that multiple physical databases were being searched and that results sets were being merged, or where these databases were located. In addition, none of the MPS Information Servers were aware of each other's presence, so no collection information was exchanged between them. Each database was searched individually by a single MPS Information Server.

5.2.1 Server Performance

The results for fsclt6m2 produced the following precision/recall figures for all the topics:

```
Queryid (Num):       all fsclt6m2
Total number of documents over all queries
    Retrieved:    13701
    Relevant:      4611
    Rel_ret:       1293
Interpolated Recall - Precision Averages:
    at 0.00       0.7095
    at 0.10       0.4430
    at 0.20       0.3237
    at 0.30       0.2081
    at 0.40       0.1704
    at 0.50       0.1146
    at 0.60       0.0921
    at 0.70       0.0520
    at 0.80       0.0365
    at 0.90       0.0194
    at 1.00       0.0167
Average precision (non-interpolated) for all rel docs(averaged over queries)
                  0.1712
Precision:
  At    5 docs:   0.4440
  At   10 docs:   0.4000
  At   15 docs:   0.3440
```

```
  At   20 docs:   0.3040
  At   30 docs:   0.2607
  At  100 docs:   0.1540
  At  200 docs:   0.1002
  At  500 docs:   0.0478
  At 1000 docs:   0.0259
R-Precision (precision after R (= num_rel for a query) docs retrieved):
    Exact:        0.2216
```

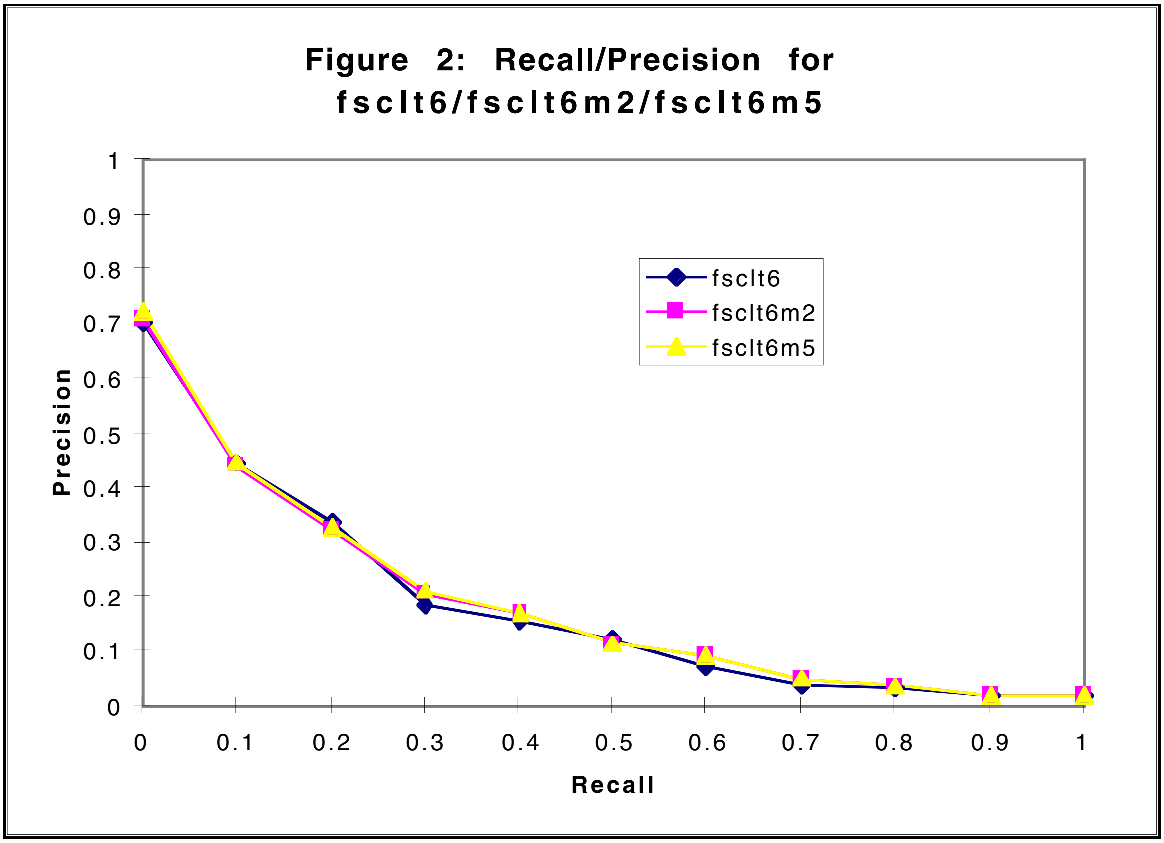
Overall, for all topics, 28% of the relevant documentswere retrieved from the database and only 9.4% of the documents retrieved were relevant.

The results for fsclt6m5 produced the following precision/recall figures for all the topics:

```
Queryid (Num):       all fsclt6m5
Total number of documents over all queries
    Retrieved:    13683
    Relevant:      4611
    Rel_ret:       1303
Interpolated Recall - Precision Averages:
    at 0.00       0.7217
    at 0.10       0.4472
    at 0.20       0.3253
    at 0.30       0.2130
    at 0.40       0.1704
    at 0.50       0.1161
    at 0.60       0.0915
    at 0.70       0.0521
    at 0.80       0.0365
    at 0.90       0.0194
    at 1.00       0.0167
Average precision (non-interpolated) for all rel docs(averaged over queries)
                  0.1729
Precision:
  At    5 docs:   0.4520
  At   10 docs:   0.4100
  At   15 docs:   0.3493
  At   20 docs:   0.3110
  At   30 docs:   0.2620
  At  100 docs:   0.1566
  At  200 docs:   0.1015
  At  500 docs:   0.0485
  At 1000 docs:   0.0261
R-Precision (precision after R (= num_rel for a query) docs retrieved):
    Exact:        0.2225
```

Overall, for all topics, 28% of the relevant documentswere retrieved from the database and only 9.5% of the documents retrieved were relevant.

Figure 2 below show the precision-recall curve for fsclt6, fsclt6m2 and fsclt6m5.



What is interesting to note is that the two merged runs (fsclt6m2 and fsclt6m5) retrieved as many documents as fsclt6, and that the precision is generally the same and even a little higher for certain recall values. This would seem to suggested that the document scoring algorithms works well when searching across databases. These results are essentially the same as the ones we got in the TREC-5 [2] database merging experiments.

The implication of this is that we can segment a very large database across a number of machines to take advantage of parallel processing[10] 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 unavailable due to repairs or maintenance.

6 Discussion of FS Consulting TREC-6 Results

The MPS Information Server is designed to operate in an interactive setting, where quick response and high precision are generally preferable to high recall[11]. Comparing the TREC-6 results with our TREC-5 [2] results really illustrates this. While our TREC-5 results returned more documents overall, the precision was lower. In fact the TREC-6 results mirror the TREC-4 [1] results more closely.

---

[10] The MPS Information Server Gateway searches multiple databases in parallel.
[11] High recall can be achieved by using relevance feedback; this is the recommended search strategy when high recall searches are required.

6.1 System improvements

While the relevance feedback algorithm works adequately at this point, it is hard not to want better performance from it. In that light we will be runninga numberof experiments this year prior to TREC-7 to fine-tune the relevance feedback algorithms further.

7 Future Work

TREC-6 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-7, as follows:

- 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.
- Currently multiple database searching is performed using the same search engine and the same ranking algorithms to search all the databases. While this works well in situations where one can use the same search engine to access databases across a numberof systems, it would not work well in situations where multiple different search engines are used[12]. For TREC-7 we plan to develop a merging gateway which would allow us to combine the search results from multiple search engines in a meaningful manner. For this work we plan to use the STARTS [3] protocol recently developed at Stanford as part of the Digital Library Project there.
- Finally we plan to participate in the VLDB track. We had planned to do so for this experiment, but delays and technical problems prevented us from doing so.

8 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.

[2] Schiettecatte, François, "Document Retrieval Using the MPS Information Server". In Harman D. (Ed) The Fifth Text Retrieval Conference (TREC-5). National Institute of Standards and Technology Special Publication,
Gaithersburg, Md. 20899.

[3] STARTS, Stanford Protocol Proposal for Internet Search and Retrieval, Luis Gravano, Kevin Chang, Hector Garcia-Molina, Carl Lagoze, and Andreas Paepcke. Digital Library Project, Stanford University, CA.

[12] As is frequently the case in most 'real world' work environments.