Research in Automatic Profile Creation and Relevance Ranking with LMDS

Julian A. Yochum

Logicon, Inc.
2100 Washington Blvd.
Arlington, VA 22204-5703
jyochum@logicon.com

Abstract

This paper describes the development of a prototype system to generate routing profiles automatically from sets of relevant documents provided by a user, and to assign relevance scores to the documents selected by these profiles. The prototype was developed with the Logicon Message Dissemination System (LMDS) for participation in the Third Text REtrieval Conference (TREC-3).

Each generated profile contains two sets of terms: a very small set to select documents, and a much larger set to assign a relevance score to each document selected. The profile generator chooses each term and assigns a weight to it, based on its frequency of occurrence in the set of documents provided by the user, and on its frequency of occurrence in a large representative corpus of documents. The LMDS search engine uses the resulting profiles to select documents, and then passes the documents to the scoring prototype for ranking. The score assigned is a function of the weights of all profile terms found in the document.

Performance figures and TREC-3 results are included.

1. Introduction

LMDS is a commercial off-the-shelf (COTS) product, designed specifically for high-speed document routing on a wide range of hardware and operating system platforms. LMDS users create interest profiles to specify the types of documents that they wish to receive. Each profile contains a list of tokens (or "search terms") which may appear in such a document, together with a Boolean expression indicating the logical combination in which the tokens must occur for LMDS to select the document. In addition, the user may optionally attach a weight to each search term.

When LMDS selects a document, it passes the document to a scoring routine, together with pointers to the set of search terms discovered in the document, and to their associated weights. In addition, LMDS provides an API which allows any installation to customize the scoring routine for its own purposes. The resulting document scores are then used to order the documents for display.

A feature not yet available in LMDS is the ability to generate profiles automatically, based on a sample of documents which the user deems relevant to a specific topic. Since the relevance judgments associated with the TREC-3 training documents provide 50 such sample sets, our objective for TREC-3 was to use these sample sets to develop a prototype system for automatic profile generation and relevance ranking.

LMDS is designed to run thousands of profiles against incoming documents in a minimum amount of time and with minimum hardware requirements. To qualify as workable enhancements to LMDS, therefore, any algorithms for automatic profile generation and relevance ranking must be not only effective, but also compact and fast. The prototype system was designed with these goals in mind, and a hardware configuration was used which reflected typical real-world resource constraints:

Processor:	Dedicated SPARCstation IPC
CPU Clock:	25 MHz
RAM:	24 MB
Hard Disk:	4 GB

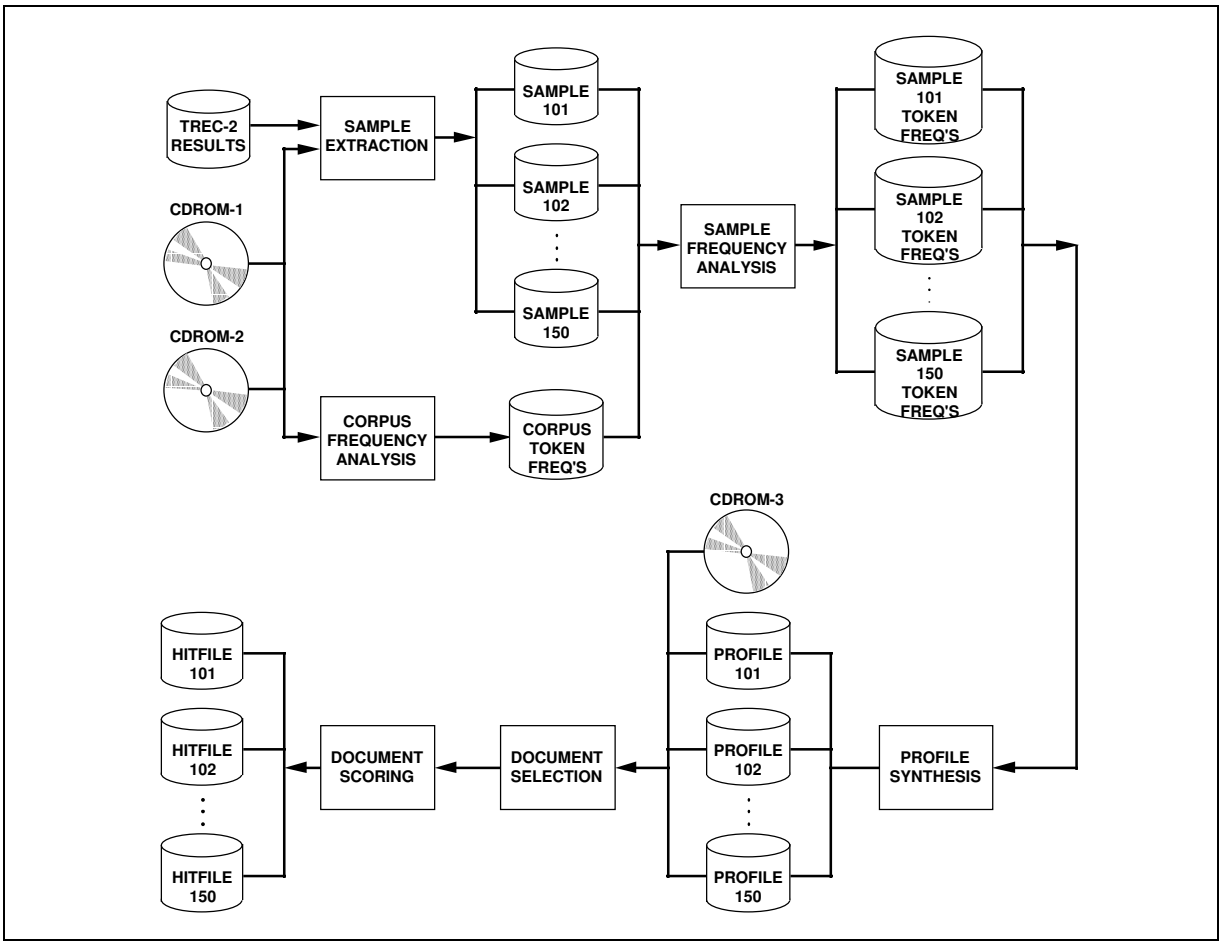


Figure 1: System Data Flow

2. Problem Description

Routing participants in TREC-3 were provided with 742,000 training documents, 2.1 gigabytes on two CDROM disks. Documents were selected from the *Wall Street Journal*, *AP Newswire*, *Computer Select* disks, *Federal Register*, and DOE abstracts, and were stored with embedded SGML formatting tags. Participants also received 50 routing topics, each a full-page description of a user's information needs, together with 50 sets of document numbers (TREC-2 retrieval results) identifying the training documents which satisfied each topic.

Participants were to use the above data to generate routing profiles and to develop relevance scoring algorithms. They were then provided with 336,000 test documents, 1.1 gigabytes on another CDROM. They were to run their routing profiles against the test documents to identify the 1,000 highest-scoring documents for each topic. They were then were to sort the set of hits for each topic in descending order of relevance, and were to return the 50 sets of 1,000 hits for evaluation by TREC-3 judges.

3. System Overview

As shown in Figure 1, the prototype system consists of six separate processes. For TREC-3, each had its own specific purpose:

1. Corpus Frequency Analysis analyzed a large representative corpus of documents to create a database of token-frequency data.

2. Sample Extraction used the TREC-2 results to extract all documents relevant to each routing topic from CDROM-1 and CDROM-2, and created a database of topic samples.

3. Sample Frequency Analysis analyzed the sample set of documents associated with each topic to determine the set of tokens that were

statistically most descriptive of the sample, and assigned a weight to each token based on its observed frequencies of occurrence in the sample and in the corpus.

4. Profile Synthesis incorporated the most descriptive tokens into a LMDS profile, appended a Boolean logic statement to indicate which tokens would be used for document selection, and sent the completed profile to LMDS for activation.

5. Document Selection used the LMDS routing engine to dispatch each active profile against each document on CDROM-3. Whenever the Boolean logic statement in a profile was satisfied by the document, LMDS sent the document to the Document Scoring process, together with the token and weighting data necessary to score the document.

6. Document Scoring calculated a document score for each profile which selected the document, based on the weights of all profile terms that were found in the document. The scoring routine then stored the document and its score in a profile-designated hitfile for TREC-3 evaluation.

The details of each of the above processes are discussed below in the context of TREC-3. Performance figures for each process are provided.

3.1. Corpus Frequency Analysis

This preliminary process analyzed all training documents on CDROM-1 and all training documents with relevance judgments on CDROM-2. (This total set of documents will be referred to hereafter as the "corpus.") For each document, the process tokenized all alphanumeric strings, and then eliminated those tokens which:

1. Were only one character long.
2. Contained one or more numeric characters.
3. Were in the stopword list.
4. Served as SGML tags.
5. Occurred only once in the corpus.
6. Were in an area of the document deemed non-searchable by TREC-3 rules.

The process next counted the number of documents in which each token occurred, then sorted the tokens alphabetically and stored them in a file with their associated document counts. A portion of this file is shown in Figure 2. The following figures summarize the performance of the process:

487,000	Documents analyzed
171	Stopwords used
241,000	Tokens identified
0.79	Wall-clock seconds/document, average

3.2. Sample Extraction

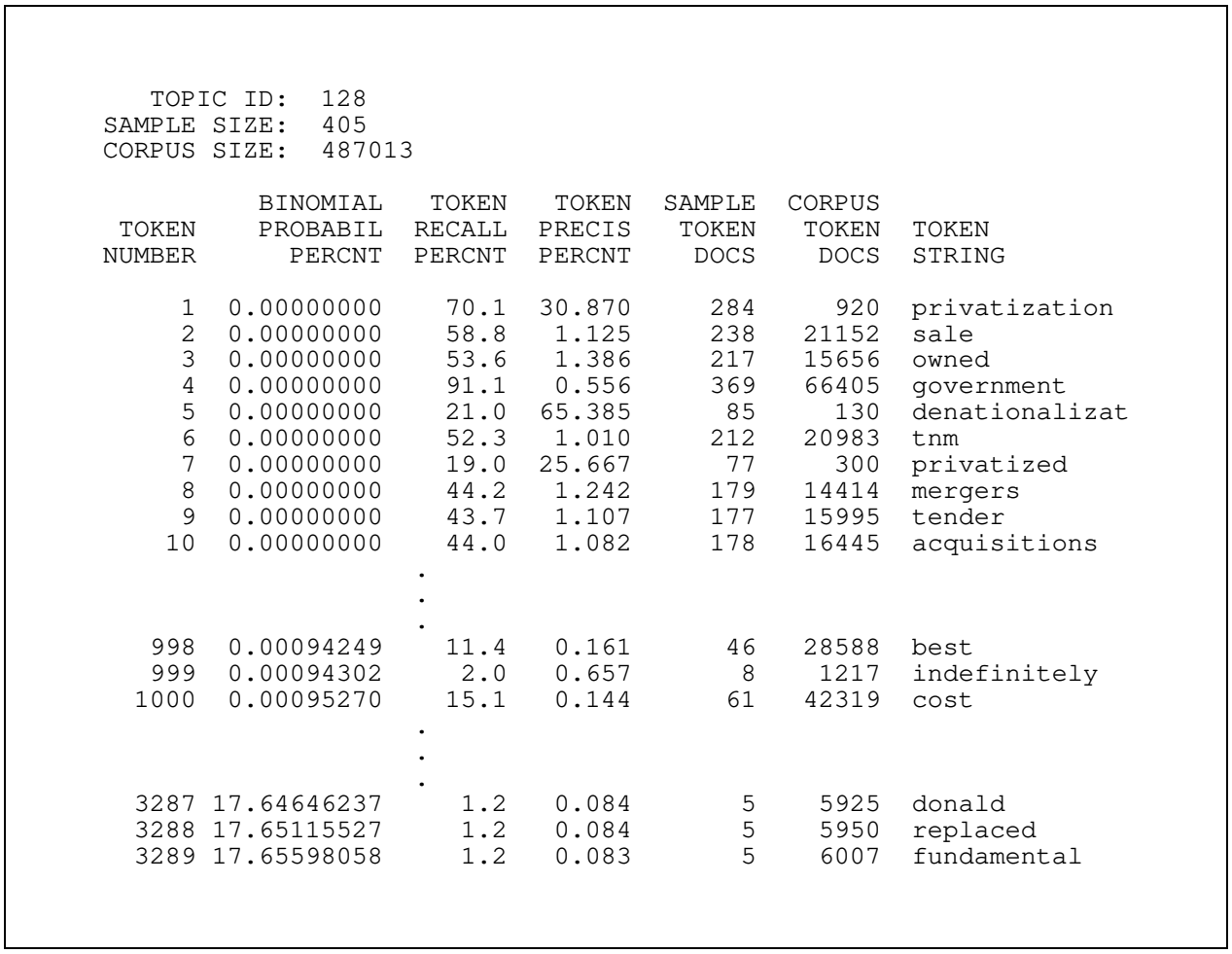
This preliminary process used the TREC-2 results to extract all relevant documents for each topic from CDROM-1 and CDROM-2, and to store these sets of documents in 50 separate directories.

```
CORPUS SIZE: 487013

CORPUS
 TOKEN  TOKEN
  DOCS  STRING

   506  aa
   126  aaa
     2  aaaa
     2  aaac
        .
        .
        .
    99  privatisation
    14  privatise
    38  privatised
    16  privatising
   920  privatization
    93  privatizations
   239  privatize
   300  privatized
     2  privatizer
     7  privatizes
   141  privatizing
        .
        .
        .
    57  zz
     5  zzz
    60  zzzz
     7  zzzzbest
```

Figure 2: Corpus Analysis Results



```
   TOPIC ID:  128
SAMPLE SIZE:  405
CORPUS SIZE:  487013

          BINOMIAL   TOKEN   TOKEN  SAMPLE  CORPUS
 TOKEN    PROBABIL  RECALL  PRECIS   TOKEN   TOKEN  TOKEN
NUMBER      PERCNT  PERCNT  PERCNT    DOCS    DOCS  STRING

     1  0.00000000    70.1  30.870     284     920  privatization
     2  0.00000000    58.8   1.125     238   21152  sale
     3  0.00000000    53.6   1.386     217   15656  owned
     4  0.00000000    91.1   0.556     369   66405  government
     5  0.00000000    21.0  65.385      85     130  denationalizat
     6  0.00000000    52.3   1.010     212   20983  tnm
     7  0.00000000    19.0  25.667      77     300  privatized
     8  0.00000000    44.2   1.242     179   14414  mergers
     9  0.00000000    43.7   1.107     177   15995  tender
    10  0.00000000    44.0   1.082     178   16445  acquisitions
                    .
                    .
                    .
   998  0.00094249    11.4   0.161      46   28588  best
   999  0.00094302     2.0   0.657       8    1217  indefinitely
  1000  0.00095270    15.1   0.144      61   42319  cost
                    .
                    .
                    .
  3287 17.64646237     1.2   0.084       5    5925  donald
  3288 17.65115527     1.2   0.084       5    5950  replaced
  3289 17.65598058     1.2   0.083       5    6007  fundamental
```

Figure 3: Sample Analysis Results for Topic 128

Each directory thus contained the sample for a given topic. No figures were kept on the time required to perform this extraction, but sample sizes varied from a low of 28 documents to a high of 792 documents.

3.3. Sample Analysis

This process was responsible for producing the sample statistics necessary for automatic profile generation. For each topic sample, the process first tokenized each document in the sample, using the same tokenizing rules as Corpus Frequency Analysis. The process then counted the number of sample documents in which each token occurred. The process then combined the sample count for each token with the corpus count for that token to calculate the binomial probability distribution *P(r)* for the token, as shown in the following formula:

$$P(r) = \frac{n!}{r!(n-r)!}\left(\frac{p}{q}\right)^r\left(1-\frac{p}{q}\right)^{n-r}$$

where:

n = documents in sample
r = sample documents containing token
p = corpus documents containing token
q = documents in corpus

Calculated in this way, the value *P(r)* can be used as a measure of how "descriptive" each token is with regard to a given sample of documents, with lower values indicating greater descriptive power.

The process next calculated the weight *w* for each token, as shown in the following formula:

$$w = \frac{r}{p}$$

```
! Profile:  P_128X_999.DIS
! Topic:    128

! This is a machine-generated profile, based on an analysis
! of the word frequencies in a sample of relevant documents,
! and of the word frequencies in the corpus as a whole.

DISSEM:  jy_1/C-128X-999
SOURCE:  all

TERMS:

1     DOC CONTAINS      privatization     (H 30870)
2     DOC CONTAINS      sale              (H 1125)
3     DOC CONTAINS      owned             (H 1386)
4     DOC CONTAINS      government        (H 0556)
5     DOC CONTAINS      denationalizat    (H 65385)
6     DOC CONTAINS      tnm               (H 1010)
7     DOC CONTAINS      privatized        (H 25667)
8     DOC CONTAINS      mergers           (H 1242)
9     DOC CONTAINS      tender            (H 1107)
10    DOC CONTAINS      acquisitions      (H 1082)
          .
          .
          .
998   DOC CONTAINS      best              (H 0161)
999   DOC CONTAINS      indefinitely      (H 0657)
1000  DOC CONTAINS      cost              (H 0144)

LOGIC:  ANY OF 1 THRU 10
```

Figure 4: Automatically Generated Profile for Topic 128

The process then created a table containing each unique token in the topic sample, together with its associated values for *P(r)* and *w*. Finally, the process sorted this table on *P(r)*, the measure of descriptiveness.

A portion of one such table is shown is Figure 3. Its rows contain the statistics for each unique token in the topic sample. Its columns contain the information described below:

1. Token Number is a one-up token identification number.

2. Binomial Probability Percent is the value *P(r)* expressed as a percent -- the probability that this token has occurred in this sample as often as it has, purely by chance.

3. Token Recall Percent is the value:

$$\frac{r}{n}$$

expressed as a percent -- the probability that this token will occur in a relevant document.

4. Token Precision Percent is the value *w* expressed as a percent -- the probability a document will be relevant if it contains this token.

5. Sample Token Documents is the value *r* -- the total sample documents containing the token.

6. Corpus Token Documents is the value *p* -- the total corpus documents containing the token.

7. Token String is the first 14 characters of the token.

3.4. Profile Synthesis

To create the actual LMDS profile, this process concatenated the top 1000 tokens with their respective weights from the sorted table, and placed them in a file. It then appended a Boolean logic statement containing the top 10 tokens in an OR condition, indicating that a document was to be selected if it contained any one of the top 10 most descriptive tokens. (Once a document was selected, however, the scoring algorithm would assign a document score based on the weights of all profile tokens that appeared in that document.) The process then sent the new profile to LMDS to be activated. A portion of one such profile is shown in Figure 4.

The following figures summarize the combined performance of the Sample Analysis and Profile Synthesis processes:

50 Profiles generated
6.1 Wall-clock minutes/profile, average

3.5. Document Selection

Document selection was performed with the LMDS 2.13 routing engine, which enabled each profile to perform a free-text scan of each document on CDROM-3.

Whenever the Boolean logic statement in any profile was satisfied by a document, LMDS sent the document to the Document Scoring process, together with the token and weighting information needed by that process to score the document.

The LMDS program product itself was not modified in any way for TREC-3. All prototype functionality was added via externally-generated LMDS profiles and via the LMDS API.

3.6. Document Scoring

For each document and set of profile tokens and weights received from LMDS, this process first tokenized the document, using the same tokenizing rules as Corpus Frequency Analysis. The process then counted the number of unique tokens in the document, and calculated the score s for the document, according to the following formula:

$$s = (1-(1-w_1)(1-w_2)\ldots(1-w_n))\frac{u}{t}$$

where:

w_i = *weight of each unique profile token in doc*
u = *unique profile tokens in doc*
t = *unique tokens in doc*

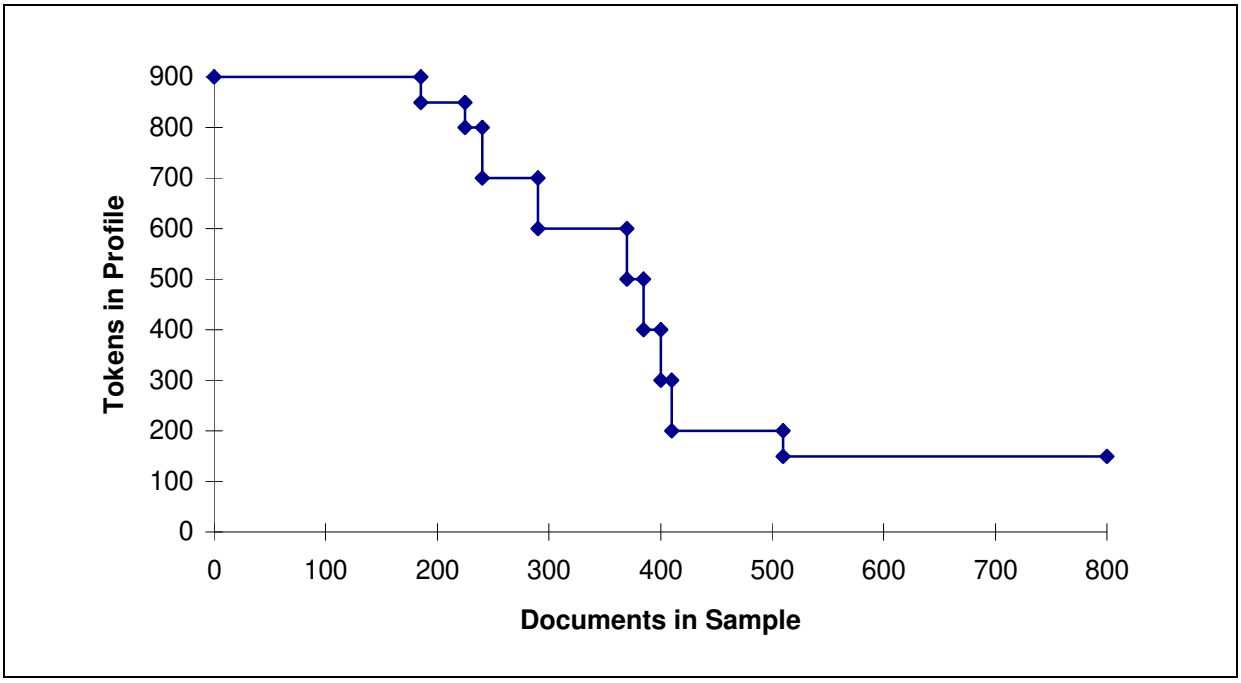


Figure 5: Observed Optimum Relationship of Sample Size to Profile Size

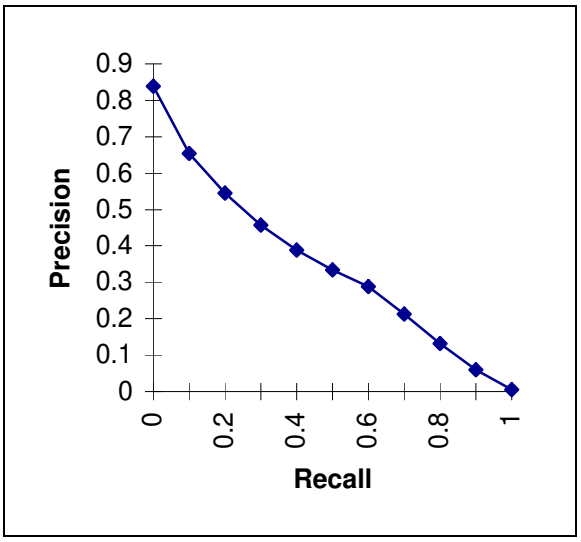


Figure 6: Recall-Precision Curve

The first part of this formula calculates the probability that this document will be relevant to a topic, given that the document contains a particular subset of the topic's most descriptive tokens. Multiplication by the ratio of *u/t* then corrects for the tendency of larger documents to achieve higher scores simply by having more tokens.

Although each profile contained 1,000 tokens, it was discovered during testing with trial documents that as the sample size increased, the accuracy of the profile usually decreased. Further testing suggested that the optimum number of profile tokens indeed was inversely proportional to the size of the sample. Accordingly, the Document Scoring process was modified to calculate *s* with only the top *x* tokens in the profile, where *x* was the value of a step function on the sample size, as shown in Figure 5.

When scoring was complete for the profile, the process stored the document number and its score into the hitfile designated by the profile. When all documents on CDROM-3 had been processed in this fashion, the 50 hitfiles were sorted on document score, and the top 1,000 entries in each hitfile were sent to the TREC-3 judges.

The following figures summarize the combined performance of the Document Selection and Document Scoring processes:

336,000	Documents processed
0.99	Wall-clock seconds/document, average

```
Total number of documents
over all queries:
   Retrieved:   50000
   Relevant:     9353
   Ret AND Rel:  6756

Interpolated recall-precision:
   At 0.00:  0.8387
   At 0.10:  0.6539
   At 0.20:  0.5450
   At 0.30:  0.4576
   At 0.40:  0.3885
   At 0.50:  0.3344
   At 0.60:  0.2877
   At 0.70:  0.2122
   At 0.80:  0.1316
   At 0.90:  0.0606
   At 1.00:  0.0055

Non-interpolated precision
over all relevant documents:
   0.3373

Precision:
   At    5 docs:  0.5840
   At   10 docs:  0.5760
   At   15 docs:  0.5747
   At   20 docs:  0.5560
   At   30 docs:  0.5253
   At  100 docs:  0.4288
   At  200 docs:  0.3540
   At  500 docs:  0.2168
   At 1000 docs:  0.1351

R-Precision:
   0.3725
```

Figure 7: Routing Results for Prototype

4. TREC-3 Results

The recall-precision scores for the prototype are summarized in the graph in Figure 6, and a table of detailed scores is provided in Figure 7.

Except for the first section of Figure 7, all results are averages of the corresponding scores for each of the 50 topics. While most sections of Figure 7 are self-explanatory, several concepts may require additional explanation:

1. Interpolated precision is the maximum precision over a range of recall points. Thus, the interpolated precision at recall 0.10 (i.e., after 10% of relevant documents have been retrieved

for a query) is the maximum precision at all recall points >= 0.10.

2. Precision at X docs is the precision after X documents have been retrieved, whether or not the documents are relevant.

3. R-Precision is the precision after R documents have been retrieved, where R is the number of relevant documents possible for a topic. Thus, if a topic has 50 possible relevant documents, precision is measured for that topic only at 50 documents.

5. Analysis

As stated in the Introduction, to qualify as workable enhancements to LMDS, any algorithms for automatic profile generation and relevance ranking must be not only effective, but also compact and fast. The prototype is evaluated below with regard to those criteria:

1. Effectiveness -- The prototype achieved aggregate scores (Figures 6 and 7) slightly above average across all measures, a strong showing for a first-time participant in TREC-3. These scores can now be used a baseline for developing improved versions of the prototype.

2. Compactness -- The only element of the prototype of any notable size is the 5-megabyte file containing the results of Corpus Analysis. Even allowing for some growth in a production environment, its costs in disk storage would remain insignificant. Moreover, the entire file does not need to be in RAM at any one time.

3. Speed -- Average wall-clock times are recapped below for those processes which could be ongoing in a production environment:

Corpus Frequency Analysis:	0.79 seconds/doc
Sample Analysis plus Profile Synthesis:	6.1 minutes/prof
Document Selection plus Document Scoring:	0.99 seconds/doc

Of these, only the figure of 6.1 minutes for profile generation is unacceptable. The profile generation software, however, is quite inefficient, and a few simple code modifications should produce a substantial increase in speed.

6. Anticipated Improvements

Substantial opportunities exist for improving the prototype. A number of these are discussed below:

1. Word-Stemming Algorithm -- Since the weight of a token is directly proportional to the number of times it occurs in a sample, use of a word-stemming algorithm should allow Sample Analysis to more accurately calculate the weight of a word which might appear many times in a sample, but in slightly different forms (e.g., “privatize” in Figures 2, 3 and 4).

2. Larger Stopword List -- Additional stopwords should help exclude from profiles those tokens with no predictive value, and should also improve the *u/t* calculation used by Document Scoring to correct for the tendency of larger documents to score higher simply by having more tokens.

3. AND Logic -- The current approach of selecting documents via an OR condition on the top 10 most descriptive tokens virtually guarantees that the set of documents passed to Document Scoring will contain all relevant documents. It also guarantees the set will contain a very large number that are totally unrelated. Examination of Sample Analysis output, such as that shown in Figure 3, strongly suggests that selecting documents via an AND condition on any 2 of the top 10 tokens would substantially reduce the number of unrelated documents selected without significantly affecting the number of relevant ones.

4. Mathematically Optimized Profile Sizes -- The optimization step function graphed in Figure 5 was derived by analyzing the results of a subset of training profiles and averaging the observed optimum relationships between sample sizes and profile sizes. While a useful tool, it cannot accurately predict the optimum size for every profile. Preliminary evidence suggests that optimum profile size actually is a cumulative function of Token Recall and Token Precision values over the set of most descriptive tokens.

5. Reduced Profile Generation Time -- Sample Analysis and Profile Synthesis are the two most inefficient processes in the prototype. Speed improvements of at least an order of magnitude should result from using C code in place of UNIX shell scripts for Profile Synthesis, and from streamlining the ways in which Sample Analysis uses the database of corpus token frequencies.

The above set of improvements should result in smaller, more accurate profiles, and in reduced throughput time for Document Selection and Document Scoring. More accurate profiles, in turn, should provide noticeable improvements in both precision and recall.

7. Summary

The prototype described in this paper is a hybrid routing system, successfully coupling the LMDS Boolean search engine with a probabilistic scoring algorithm, and using the speed of LMDS to quickly reduce the set of documents that must be evaluated by the more computationally-intensive scoring algorithm. As implemented, the prototype is simple, compact and fast, and requires no special hardware.

TREC-3 scores for the prototype are highly encouraging, and will be used as a baseline against which to measure future improvements. A set of such improvements has been identified, and will be implemented and evaluated for TREC-4.