Research in Automatic Profile Generation and Passage-Level Routing 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 Fourth Text REtrieval Conference (TREC-4).

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 entire document, and of those found in fixed-length overlapping passages within the document.

Performance figures and TREC-4 results are included. An appendix describes a modification to the TREC-4 algorithm, made since the conference, which has produced significant improvements in both recall and precision.

1. Introduction

LMDS is a commercial off-the-shelf (COTS) product, designed specifically for high-speed document routing [1] 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 "query by example," the capability 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-4 training documents provide 50 such sample sets, our objective for TREC-4 was to use these sample sets to refine and enhance the automatic profile generation and relevance ranking prototype system [2] that we developed for TREC-3.

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 not only be effective, but must also be compact and fast. The TREC-4 prototype system was designed with these goals in mind, and a hardware configuration was

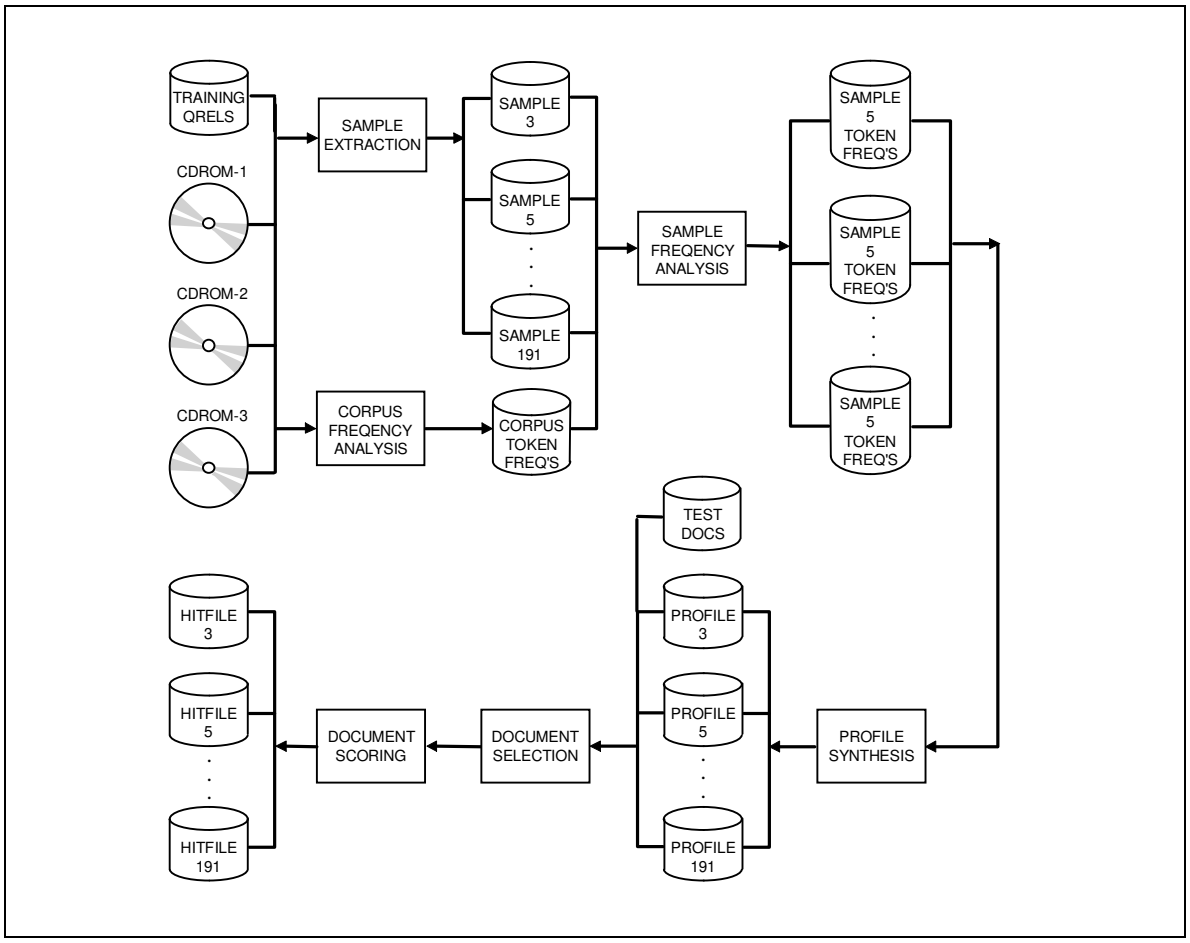


Figure 1: System Data Flow

used which reflected typical real-world resource constraints:

Processor:	Dedicated SPARCstation 10
CPU Clock:	50 MHz
RAM:	32 MB
Hard Disk:	5 GB

2. Problem Description

Routing participants in TREC-4 were provided with 1,078,000 training documents, 3.2 gigabytes on three CDROM disks. Documents were selected from the *Wall Street Journal*, *AP Newswire*, *Computer Select* disks, *Federal Register*, *San Jose Mercury News*, DOE abstracts, and US patents, and were stored with embedded SGML formatting tags.

Participants also received 50 routing topics, each a one-page description of a user's information needs, together with 50 sets of document numbers (prior TREC retrieval relevance judgments, or "QRELS") 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 329,000 test documents (0.9 gigabytes) selected from the *Federal Register*, *Computer Select* disks, *IR Digest*, *Virtual Worlds*, and Internet newsgroups.

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 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-4 judges.

3. System Overview

As shown in Figure 1, the prototype system consisted of six separate processes, each with 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 training QRELS to extract all documents relevant to each routing topic from CDROMs 1 through 3, 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 of the test documents. 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-4 evaluation.

The details of each of the above processes are discussed below in the context of TREC-4. Performance figures for each process are provided.

3.1. Corpus Frequency Analysis

This preliminary process analyzed all training documents on CDROMs 1 through 3. (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, a slightly augmented version of the list used by SMART [3].
4. Served as SGML tags.
5. Occurred <= twice on each CDROM.
6. Were in an area of the document deemed non-searchable by TREC-4 rules.

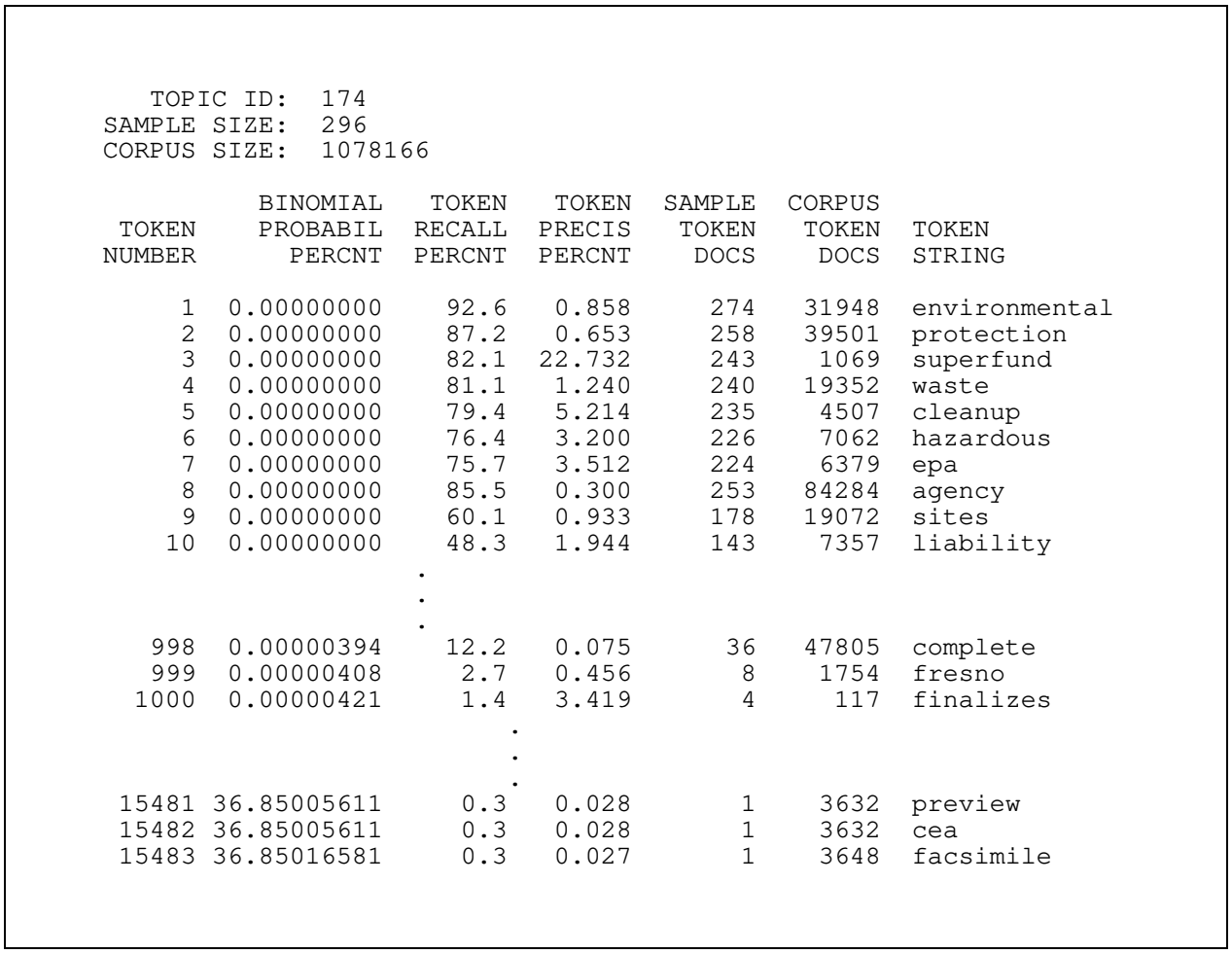
The process next counted the number of documents in which each token occurred, then

```
CORPUS SIZE: 1078166

CORPUS
 TOKEN  TOKEN
  DOCS  STRING

   712  aa
   201  aaa
     2  aaaa
     2  aaac
        .
        .
        .
     9  superfonts
     4  superforms
     4  superfos
    56  superframe
     5  superfunction
  1069  superfund
     4  superfundlist
     7  superfused
     6  superfusion
    18  superguage
    10  supergen
        .
        .
        .
   114  zz
     6  zzz
    88  zzzz
     7  zzzzbest
```

Figure 2: Corpus Analysis Results



```
   TOPIC ID:  174
SAMPLE SIZE:  296
CORPUS SIZE:  1078166

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

     1  0.00000000    92.6   0.858     274   31948  environmental
     2  0.00000000    87.2   0.653     258   39501  protection
     3  0.00000000    82.1  22.732     243    1069  superfund
     4  0.00000000    81.1   1.240     240   19352  waste
     5  0.00000000    79.4   5.214     235    4507  cleanup
     6  0.00000000    76.4   3.200     226    7062  hazardous
     7  0.00000000    75.7   3.512     224    6379  epa
     8  0.00000000    85.5   0.300     253   84284  agency
     9  0.00000000    60.1   0.933     178   19072  sites
    10  0.00000000    48.3   1.944     143    7357  liability
                      .
                      .
                      .
   998  0.00000394    12.2   0.075      36   47805  complete
   999  0.00000408     2.7   0.456       8    1754  fresno
  1000  0.00000421     1.4   3.419       4     117  finalizes
                          .
                          .
                          .
 15481 36.85005611     0.3   0.028       1    3632  preview
 15482 36.85005611     0.3   0.028       1    3632  cea
 15483 36.85016581     0.3   0.027       1    3648  facsimile
```

Figure 3: Sample Analysis Results for Topic 174

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:

1,078,000	Documents analyzed
580	Stopwords used
249,000	Tokens identified
0.03	Wall-clock seconds/document, average

3.2. Sample Extraction

This preliminary process used the training QRELS to extract all relevant documents for each topic from the corpus, and to store these sets of documents into 50 separate directories. 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 803 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

```
! Profile:  Q-174-999-1.DIS
! Topic:    174

! 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/D-174-999
SOURCE:  all

TERMS:

1     DOC CO      environmental     (H   858)
2     DOC CO      protection        (H   653)
3     DOC CO      superfund         (H 22732)
4     DOC CO      waste             (H  1240)
5     DOC CO      cleanup           (H  5214)
6     DOC CO      hazardous         (H  3200)
7     DOC CO      epa               (H  3512)
8     DOC CO      agency            (H   300)
9     DOC CO      sites             (H   933)
10    DOC CO      liability         (H  1944)
        .
        .
        .
998   DOC CO      complete          (H    75)
999   DOC CO      fresno            (H   456)
1000  DOC CO      finalizes         (H  3419)

LOGIC:  ANY 2 OF 1 THRU 10
```

Figure 4: Automatically Generated Profile for Topic 174

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:

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

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 indicating that a document was to be selected if it contained any two of the top ten 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
5.4	Wall-clock minutes/profile, average
0.8	Wall-clock seconds/document, average

3.5. Document Selection

Document selection was performed with the LMDS 2.1.4 routing engine, which enabled each profile to perform a free-text scan of each test document.

Whenever the Boolean logic statement in any profile was satisfied by a document, LMDS sent the

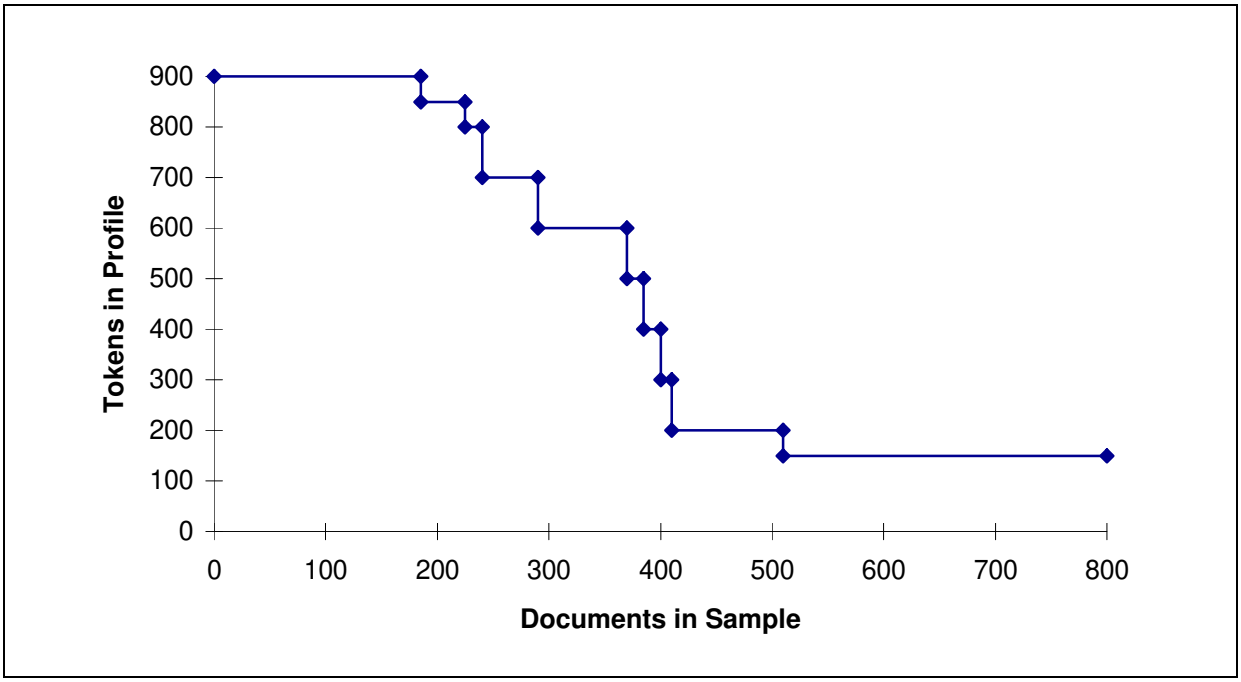


Figure 5: Observed Optimum Relationship of Sample Size to Profile Size

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-4. 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 next divided the document into a set of "passages," where each passage consisted of 100 contiguous tokens, and each new passage began 50 tokens after the beginning of the previous passage. The process then calculated a set of intermediate scores, consisting of:

1. An intermediate score for each passage:

$$s_p = (1-(1-w_1)(1-w_2)...(1-w_n))$$

2. An intermediate score for the entire document:

$$s_d = (1-(1-w_1)(1-w_2)...(1-w_n))\frac{u}{t}$$

where:

n = *unique profile tokens in passage or document*
w_i = *weight of each unique profile token in passage or document*
u = *unique profile tokens in document*
t = *unique tokens in document*

The passage formula calculated the probability of relevance to a topic, given the presence of a particular subset of the topic's most descriptive tokens. The document formula did the same, except that it multiplied by the ratio of *u/t* to correct 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 TREC-3 testing that as the sample size increased, the accuracy of the profile usually decreased [2]. 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 each intermediate score 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 all intermediate scores had been calculated, the process then combined the scores in two different ways to produce two final document scores, one for each TREC-4 run (losPA2 and losPA3):

1. For losPA2, the process used the highest passage score for each document to determine the top 1000 documents for each topic, then assigned the intermediate document score as the final document score.

2. For losPA3, the process multiplied the intermediate document score by two, then added the highest passage score to produce the final score.

For each run, the process then stored the document number and its score into the hitfile designated by the profile. When all test documents 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-4 judges.

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

329,000	Documents processed
0.45	Wall-clock seconds/document, average
27,000	Documents scored per topic, average

4. TREC-4 Results

The detailed results for runs losPA2 and losPA3 are provided in Figure 6. Run losPA3 achieved better results, and its recall-precision scores are summarized in the graph in Figure 7.

Except for the first section of Figure 6, all scores are averages of the corresponding scores for each of the 50 topics. While most sections of Figure 6 are self-explanatory, several concepts may require additional explanation:

Run ID:	losPA2	losPA3
Total docs over all topics:		
Retrieved:	50000	50000
Relevant:	6576	6576
Ret AND Rel:	4993	4957
Interpolated recall-precision:		
At 0.00:	0.7848	0.7553
At 0.10:	0.5105	0.5174
At 0.20:	0.4436	0.4505
At 0.30:	0.3738	0.3796
At 0.40:	0.3262	0.3234
At 0.50:	0.2817	0.2820
At 0.60:	0.2293	0.2354
At 0.70:	0.1781	0.1874
At 0.80:	0.1303	0.1293
At 0.90:	0.0598	0.0668
At 1.00:	0.0114	0.0122
Non-interpolated precision over all relevant documents:	0.2793	0.2834
Precision at:		
5 docs:	0.5160	0.5160
10 docs:	0.4940	0.4960
15 docs:	0.4840	0.4840
20 docs:	0.4700	0.4670
30 docs:	0.4433	0.4527
100 docs:	0.3578	0.3550
200 docs:	0.2757	0.2828
500 docs:	0.1651	0.1651
1000 docs:	0.0999	0.0991
R-Precision:	0.3108	0.3161

Figure 6: Routing Results: losPA2 and losPA3

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.

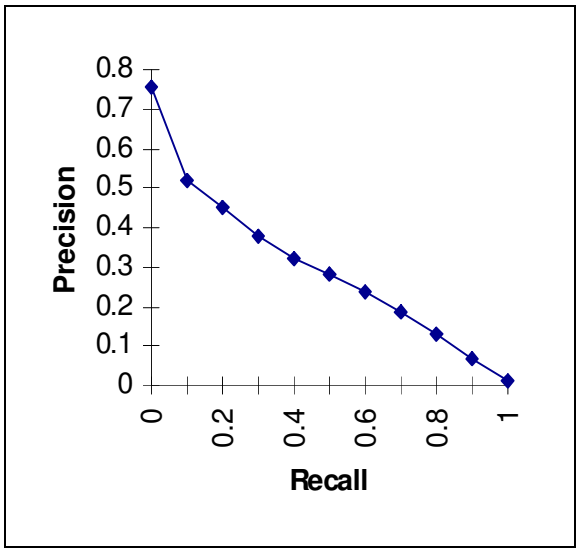


Figure 7: Recall-Precision Curve: losPA3

5. Analysis

To judge the effect of the improvements incorporated into our TREC-4 prototype, we performed a benchmark after the conference, using our TREC-3 algorithm [2] with the TREC-4 training QRELS, test documents, and scoring software. The benchmark revealed modest improvements in both recall (2.1%) and precision (5.0%) over TREC-3, and its results are summarized in the first three rows of the table in Figure 8.

The fourth row of this table contains the results of an experimental run, also performed since the conference, using two-token phrases for search terms. This phrase-based approach achieved substantial improvements both in recall (9.1%) and precision (16.9%) over the TREC-3 approach, and is discussed in more detail in the Appendix.

The columns in Figure 8 contain the following data:

1. Algorithm identifies the approach which produced the results. "TREC-3" here refers to the benchmark using the Logicon TREC-3 algorithm with the TREC-4 training QRELS and test documents. "Phrases" here indicates the experimental algorithm described in the Appendix.

2. Recall is the total number of relevant documents retrieved divided by the total number of relevant documents.

Algorithm	Recall	Average Precision	Recall Improvement Over TREC-3 Baseline	Precision Improvement Over TREC-3 Baseline
TREC-3*	73.81%	26.99%		
losPA2	75.93%	27.93%	2.87%	3.48%
losPA3	75.38%	28.34%	2.13%	5.00%
Phrases*	80.58%	31.55%	9.17%	16.90%
*Unofficial results, using TREC-4 QRELS, documents, and scoring software				

Figure 8: Comparison of Logicon Results

3. Average Precision is the non-interpolated average precision over all relevant documents.

4. Recall Improvement Over TREC-3 Baseline is the percentage difference between Recall and "TREC-3" Recall.

5. Precision Improvement Over TREC-3 Baseline is the percentage difference between Average Precision and "TREC-3" Average Recall.

6. 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.

Processing logs show that, for any given topic, the Boolean-based Document Selection process sent an average of only 8.2% of the test documents to the Document Scoring process (27,000 of 329,000), yet this small set still contained an average of 96% of the documents relevant to the topic. Moreover, the average wall-clock time required to completely process each document against all topics was only 0.45 second per document.

Passage-based routing has improved our scores compared to TREC-3, and phrase-based experiments promise substantial improvements still to come.

7. References

[1] J. Yochum. A High-Speed Text Scanning Algorithm Utilizing Least Frequent Trigraphs. In *Proceedings of the IEEE International Symposium on New Directions in Computing*, pages 114-121, Trondheim, Norway, 1985.

[2] J. Yochum. Research in Automatic Profile Generation and Relevance Ranking with LMDS. In D. Harman, editor, *Overview of the Third Text REtrieval Conference (TREC-3)*, pages 289-297. National Institute of Standards and Technology Special Publication 500-225, 1995.

[3] C. Buckley. *Implementation of the SMART Information Retrieval System*, Technical Report TR 85-686, Computer Science Department, Cornell University, Ithaca, NY, 1985.

APPENDIX

An Experiment with Phrase-Based Routing

A1. Introduction

This appendix describes a set of extensions to the TREC-4 algorithm, made since the conference, to incorporate the use of two-token phrases as search terms.

The primary effort required by this experiment involved the development of a program to identify the phrases in a document. Minor modifications were then made to several of the processes of the TREC-4 prototype to utilize this program. The program and the modifications are described below.

A2. Phrase Identification Program

To locate the phrases within a document, this program first translated each non-alphanumeric character in the document to a blank, leaving a set of alphanumeric tokens separated by strings of one or more blanks. The program then translated to a blank any token which:

1. Was only one character long.
2. Contained one or more numeric characters.
3. Was in the TREC-4 algorithm stopword list.
4. Served as an SGML tag.
5. Was in an area of the document deemed non-searchable by TREC-4 rules.

When this process was complete, the program treated as a phrase any two tokens separated by only a single blank.

A3. System Overview

The phrase-based approach used the same six processes as the TREC-4 algorithm. Accordingly, each process is described below only as it differed from its associated TREC-4 process.

TOPIC ID: 174
SAMPLE SIZE: 296
CORPUS SIZE: 1078166

TOKEN NUMBER	BINOMIAL PROBABIL PERCNT	TOKEN RECALL PERCNT	TOKEN PRECIS PERCNT	SAMPLE TOKEN DOCS	CORPUS TOKEN DOCS	TOKEN STRING
1	0.00000000	84.1	2.670	249	9326	environmental protection
2	0.00000000	83.4	3.139	247	7869	protection agency
3	0.00000000	60.1	7.639	178	2330	hazardous waste
4	0.00000000	34.8	21.458	103	480	waste sites
5	0.00000000	32.1	20.085	95	473	environmental response
6	0.00000000	31.8	20.302	94	463	liability act
7	0.00000000	31.8	19.831	94	474	comprehensive environmental
8	0.00000000	19.3	58.163	57	98	superfund law
9	0.00000000	23.0	16.915	68	402	cleanup costs
10	0.00000000	25.0	9.451	74	783	toxic waste
.						
.						
.						
998	0.00000008	4.1	0.323	12	3710	state law
999	0.00000008	2.0	1.770	6	339	environmental regulations
1000	0.00000008	1.4	9.302	4	43	meet epa

Figure A1: Sample Analysis Results (Phrases) for Topic 174

1. Corpus Frequency Analysis was augmented to create a database of phrase-frequency data in addition to its database of token-frequency data.

2. Sample Extraction was not modified.

3. Sample Frequency Analysis was augmented to also determine the set of phrases that were statistically most descriptive of the sample, and to assign a weight to each phrase based on its observed frequencies of occurrence in the sample and in the corpus. Sample phrase-related output from this process is shown in Figure A1.

4. Profile Synthesis was modified to merge both the most descriptive tokens and the most descriptive phrases into a LMDS profile, and to append a Boolean logic statement indicating which tokens and phrases would be used for document selection. A sample profile produced by this process is shown in Figure A2.

5. Document Selection was not modified.

6. Document Scoring was modified to calculate only the intermediate score for each passage:

$$s_p = (1-(1-w_1)(1-w_2)...(1-w_n))$$

where:

w_i = *weight of each unique profile token or phrase in the passage*

The process then assigned the highest passage score as the final document score.

A4. Phrase-Based Results

To judge the effectiveness of the phrase-based modifications described above, we ran a benchmark using the TREC-4 training QRELS, test documents, and scoring software. The detailed results of this run are provided in Figure A3.

```
! Profile:  Q-174-999-1.DIS
! Topic:    174

! 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/D-174-999
SOURCE:  all

TERMS:

1      DOC CO       environmental             (H   858)
2      DOC CO       protection                (H   653)
3      DOC CO       environmental protection  (H  2670)
4      DOC CO       protection agency         (H  3139)
5      DOC CO       superfund                 (H 22732)
6      DOC CO       waste                     (H  1240)
7      DOC CO       cleanup                   (H  5214)
8      DOC CO       hazardous                 (H  3200)
9      DOC CO       epa                       (H  3512)
10     DOC CO       hazardous waste           (H  7639)
         .
         .
         .
998    DOC CO       resolved                  (H   234)
999    DOC CO       public notices            (H  7407)
1000   DOC CO       ross                      (H   331)

LOGIC:  ANY OF 1 THRU 10
```

Figure A2: Automatically Generated Profile (Phrases) for Topic 174

```
Total docs over all topics:
  Retrieved:   50000
  Relevant:     6576
  Ret AND Rel:  5299

Interpolated recall-precision:
    At 0.00:  0.7511
    At 0.10:  0.5383
    At 0.20:  0.4668
    At 0.30:  0.4172
    At 0.40:  0.3719
    At 0.50:  0.3339
    At 0.60:  0.2680
    At 0.70:  0.2131
    At 0.80:  0.1693
    At 0.90:  0.1239
    At 1.00:  0.0204

Non-interpolated precision
over all relevant documents:
              0.3155

Precision at:
     5 docs:  0.6160
    10 docs:  0.5340
    15 docs:  0.5133
    20 docs:  0.4920
    30 docs:  0.4613
   100 docs:  0.3600
   200 docs:  0.2927
   500 docs:  0.1757
  1000 docs:  0.1060

R-Precision:
              0.3410
```

Figure A3: Routing Results: Phrases

A5. Summary

As was noted above in Section 5 and was shown in Figure 8, the phrase-based approach achieved significant improvements both in recall (9.1%) and in precision (16.9%) over the baseline TREC-3 algorithm.

These results are highly encouraging, and make it clear that phrases will be an important addition to future versions of the algorithm.