Dartmouth College at TREC 2007 Legal Track

Wei-Ming Chen and Paul Thompson
Department of Computer Science
Dartmouth College

Abstract

This report describes Dartmouth College's approach and results for the 2007 TREC Legal Track. Our original plan was to use the Combination of Expert Opinion (CEO) algorithm [1], to combine the search results from several search engines. However, we did not have enough time to build the index for more than one search engine by the time for submission for official runs. The official results described here are based only on the Lemur / Indri [2] search engine.

Introduction

This year, all Legal Track participants were required to submit at least one automatic run in which only the RequestText field was used with no human intervention. In other runs any fields could be used, e.g., processing the Boolean query sentences to find synonyms of keywords or parsing the complaints text to find more context of the request.

Our plan was to use the CEO algorithm [1] to combine the results returned from several search engines. This algorithm uses a probability model to compute a final score for a document and then sorts the retrieved documents according to this score. This algorithm can also receive relevance feedback to train the weights given to different search engines. In our experiments to date we have not used this feature due to lack of time. Our official results are only based on using the Indri [2] search engine.

Steps

After downloading the xml files from the TREC ftp site, we used a program written by the University of Massachusetts to convert the whole data set from xml to TREC format. Since the Lemur toolkit supports TREC documents, we ran the program provided along with the toolkit to build the index for the Indri [2] search engine.

Another search engine we planned to use was Lucene [3], an open source project in Apache. Unfortunately, it does not support TREC format. Therefore a simple parser was written to extract documents which were then passed to Lucene [3]. In addition to the DOCNO and the content text of a document, other meaningful fields such as titles, authors, document type, and organization were extracted. Others fields that use numbers such as page count were not indexed.

The CEO algorithm expects scores to be combined to be probabilities. Because the scores returned by Indri [2] are the logarithms of probabilities of relevance, we normalize them to be within the range of zero and one. By passing document IDs and normalized scores as arguments, the function returns a list of documents sorted by the final score.

Runs

Our results are all automatic runs. The first run, which is also our only official run, is the result returned by the Indri [2] search engine. The second run is the unofficial result returned by the Lucene [3] search engine, while the third one combines the previous two runs using the CEO algorithm.

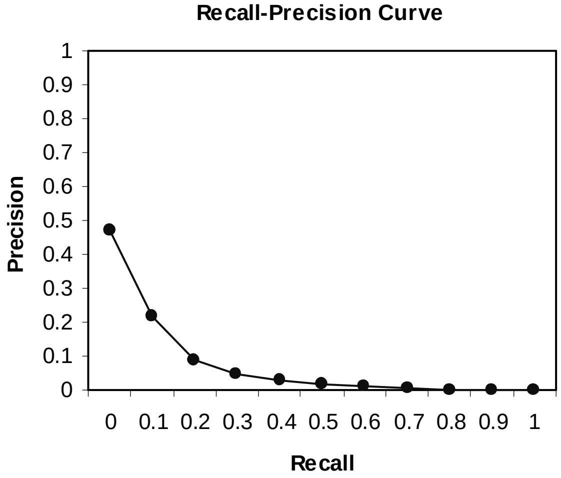

Figure 1.a: Recall-Precision Graph (Indri)

	Precision
At 5 docs	0.2791
At 10 docs	0.2395
At 15 docs	0.1969
At 20 docs	0.1779
At 30 docs	0.1496
At 100 docs	0.0914
At 200 docs	0.0621
At 500 docs	0.0393
At 1000 docs	0.0266

Figure 1.b: Document Level Averages (Indri)

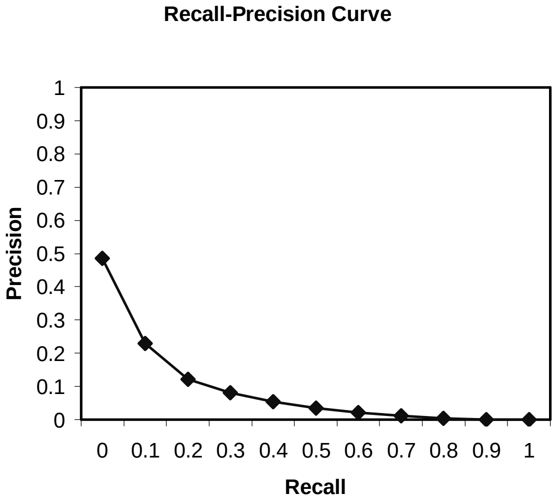

Figure 2.a: Recall-Precision Graph (Lucene)

	Precision
At 5 docs	0.2605
At 10 docs	0.2395
At 15 docs	0.2217
At 20 docs	0.2000
At 30 docs	0.1798
At 100 docs	0.1172
At 200 docs	0.0829
At 500 docs	0.0483
At 1000 docs	0.0320

Figure 2.b: Document Level Averages (Lucene)

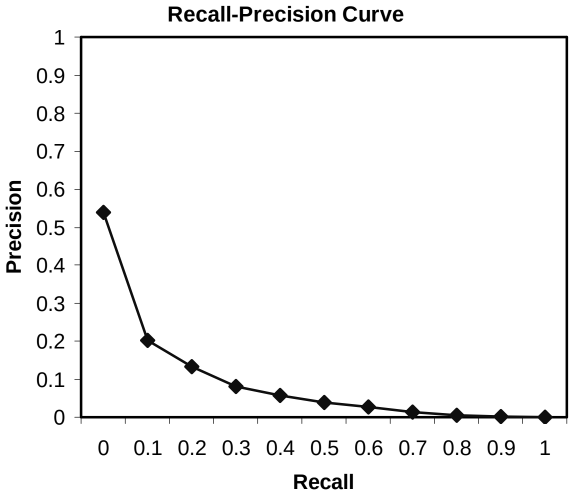

Figure 3.a: Recall-Precision Graph (CEO)

	Precision
At 5 docs	0.3023
At 10 docs	0.2512
At 15 docs	0.2217
At 20 docs	0.1965
At 30 docs	0.1705
At 100 docs	0.1151
At 200 docs	0.0821
At 500 docs	0.0502
At 1000 docs	0.0322

Figure 2.b: Document Level Averages (CEO)

Discussion

As a default the CEO algorithm gives equal weight to retrieval engines being combined to produce its final probability of relevance for a document. Since no training was done, this default mode of combination was used. As shown in the tables recall at 5 and 10 documents was slightly better with the combined run than with either search engine alone. In further experiments we plan to use relevance judgments from the relevance feedback task to train the algorithm. We expect this training to lead to better performance of the algorithm.

Future Work

From just passing the whole request text as the query this year, we plan to further process the request in the future. Through syntactic analysis we hope to find keywords in the sentence automatically and then to use query expansion to produce our final query.

We also plan to combine the results of more search engines in the future. Recall and precision might be increased if more relevant documents are retrieved, given the use of more search engines. We also plan to make more use of the CEO algorithm's relevance feedback capability. Through training the algorithm can assign more accurate weights to the engines being combined, which should lead to the final combined ranking being more accurate.

Acknowledgement

We wish to thank Afra Zomorodian for making a computer with a large memory available for this project.

References

[1] Paul Thompson. *"A Combination of Expert Opinion Approach to Probabilistic Information Retrieval."* Information Processing & Management Vol. 26, No, 3, pp. 371-394, 1990
[2] Lemur Project. http://www.lemurproject.org/
[3] Apache Lucene Project. http://lucene.apache.org/