Fusion of Information Retrieval Engines (FIRE)

S.Alaoui Mounir, N. Goharian, M. Mahoney, A. Salem, O. Frieder

Computer Science Department
Florida Institute of Technology
Melbourne, FL 32901

Abstract

We examine the feasibility of fusing the outputs of multiple text retrieval engines to improve accuracy. We tested three Web-based search engines (Excite for Web Servers, Infoseek's Ultraseek Server, and Sony Search Engine) over a 74,520 document collection (TREC Wall Street Journal articles from 1990-92) with a set of 125 natural-language queries with relevance judgements (TREC topics 51-175). We show that a weighted combination of scores produces higher precision over the top 5, 105 20, and 30 documents than any single engine over the same data set. We also compare favorably against the state-of-the-art heuristics in merging search engines. Our results suggest that fusing the results from the most dissimilar engines (those with the least overlap in the retrieved sets) is a more effective strategy than simply weighting the best engines more heavily.

Keywords: Information Fusion, Distributed Applications, Web Search Engines, Information Retrieval, TREC.

1 Introduction

Information overload is no longer a threat but a daily reality. Whereas, in the past, the difficulty to find relevant information stemmed from the inability of the user to locate and access the desired information, today, the difficulty rests in the filtering of the desired nuggets of information from the sea of chaff.

Daily users retrieve distributed information from the World-Wide-Web (WWW) via the use of search engines. Many engines exist including Excite, Infoseek, Yahoo, Alta Vista, Sony Search Engine, and Lycos. Each engine boasts of a set of features and processing characteristics that differentiate it from the rest. The common goal of each engine is to yield a highly accurate set of results, particularly in terms of precision. Precision is the percentage of the number of relevant documents retrieved from the total number of documents retrieved. Both a recent unpublished study by Excite Corporation [Wu97] and a recent historical perspective on web search engines [Hahn98] imply that users are mostly interested in accuracy for only the top screen or two of retrieved results and seldom look beyond the first few screens. Therefore, in such applications, high precision may come at the expense of recall. Recall is the percentage of the number of relevant documents retrieved from the total number of relevant documents available collection-wide.

In spite of each vendor's claims, one best engine does not exist. The motivation behind our work is to fuse the results of multiple engines, capitalizing on the advantages of each with the hope that the weaknesses of each engine will be masked by the other engines. We developed a heuristic system, called FIRE (Fusion of Information Retrieval Engines) that merges the results from multiple parallel search engines, where all engines access the same data set.

To meet user expectation and match typical user request patterns, we emphasize on high precision, and focus on short queries. We evaluate our results in terms of the top 5, 10, 20, and 30 documents retrieved. We, also require our fusion heuristic to be simple, thus introducing low computational overhead. We favorably compare FIRE against the results of the individual engines as well as against prior fusion results techniques.

2 Background

The idea of combining results from multiple engines or from multiple runs to yield better overall results is not new. Kantor [Kant94] combined the results of an individual search engine using different fusion rules. However, his results demonstrated that it is not easy to get better results using multiple engines as compared to only a single search engine. Shaw and Fox [Shaw94] used the combination of results from several different operational paradigms generated by a single engine. By summing the similarity values obtained, they demonstrated better overall accuracy than using a single similarity value.

More recently, Gauch and Wang [Gauc96] developed a system, called ProFusion, that incorporates multiple parallel Web-based search engines to process queries. Using a small set of queries, they evaluated results obtained by merging rankings from multiple independent parallel searches against the results of each individual engine. Their results did indeed demonstrate an overall accuracy improvement of the fused multiple engine search results over any of the individual engines. Since each search engine used had different input data, i.e., indexed different portions of the web, with some degree of overlap, and since no standard data or query sets were used in the evaluation, it is difficult to accurately assess and compare against the obtained results.

In the latest ACM SIGIR Conference, Lee [Lee97] also proposed and analyzed improvements obtained by combining the results of multiple search engines. Unlike the Web search engines used by Gauch and Wang in their study, Lee relied on a set of engines that had recently participated in the National Institute of Standards and Technology (NIST) Text Retrieval Conference (TREC). Lee developed several heuristics, the main two, Comb-SUM and Comb-MNZ. Lee concluded that his fusion technique Comb-MNZ provided better retrieval effectiveness over the individual engines and previous methods. FIRE improves on these results.

Finally, an information retrieval system, called SENTINEL, that supports both the fusion of multiple engines and a three-dimensional, interactive, visualization user interface is described in [Fox98]. Accuracy assessments using TREC-6 data are presented, but not for the individual engines that are fused. Since the underlying search engines used are proprietary and their individual scores are not presented, it is not possible to independently evaluate the actual fusion algorithm.

Other fusion experiments focused on combining separate, non-overlapping data sets to yield results for collections that are not integrated. Both Voorhees, Gupta, and Johnson-Laird [Voor94] and Baumgarten [Baum97] focus on merging results from such separate non-overlapping document collections. Voorhees, Gupta, and Johnson-Laird's work was predominantly experimental and was conducted on the TREC-3 data sets. Baumgarten focused on developing a probabilistic ranking principle over different sub-collections, each with different indexing. The work was predominantly theoretical, and no experimental evaluation of the derived model and the selection criterion were presented. Finally, in 1995, an entire track within the TREC activities was devoted to the concept of merging separate databases [Harm95]. For a recent overview of the NIST TREC activities, see [Harm98].

Smeaton & Crimmins [Smea98] created a Java based user interface for multiple search engines. Individual search engine results are merged and displayed. No accuracy measurements were presented.

For a general overview of information retrieval, the readers are referred to general texts and articles such as those by Salton [Salt89], Kowalski [Kowa97], Grossman and Frieder [Gros98] and Gudivada, et. al [Gudi97]. Grossman and Frieder devote an entire chapter to the topic of distributed information systems, and describe, in somewhat detail, two sample Web-based information retrieval systems. As of December 1997, the best commercial search engines have been found to index only a third of the estimated 180 million Web pages [SCIE981.

3 Search Engines Descriptions

We combined the results of three search engines designed for use on local web sites. The engines are:

- EWS - Excite for Web Servers [EWS981
- SEEK - Infoseek's Ultraseek Server [SEEK981
- SONY - Sony Search Engine [SONY981

All of these engines, and text retrieval systems in general, work by matching terms in the query to terms in the document. They assign a score to each document based on the number of matching terms and other criteria, sort the documents by score, and present the highest ranking documents to the user. The engines differ mostly in how the scores are computed and how the terms are parsed.

3.1 Sony Search Engine

SONY is an extended Boolean engine that, by default, inserts an AND operation between terms. Because every query term must be present, it does poorly on long queries. Documents are scored and sorted by counting the number of matches to each query term and multiplying the counts together. SONY defines a term as any sequence of characters except spaces, quotes, and parentheses. It does no stemming (suffix removal), but counts a match if the query term matches a prefix of the document term. There is no stop-word removal for common terms, and all terms are weighted equally.

3.2 Infoseek's Ultraseek Server

SEEK scores documents by adding rather than multiplying term frequencies; so, not all query terms must be present. Terms are weighted by inverse document frequency; matches to words that appear in many documents, such as "the", are considered less significant. SEEK uses sophisticated language dependent rules to match equivalent terms such as "Move" to "moving" or "Oracle8" to "oracle-8".

3.3 Excite for Web Servers

EWS, like SEEK, uses a weighted summation of term frequencies, but it takes the square root of each term count, since the first match is likely to be more significant than subsequent matches. EWS also makes an adjustment for document length. It ignores common words from a list of 199 stop-terms. EWS does no stemming but uses a "concept-based" retrieval system to match related terms, such as "intellectual property rights" to "software piracy" but not to "real estate". The system is proprietary, however, techniques for doing this based on relevance feedback or automatic thesaurus generation are well known.

4 Test Documents

We tested 125 queries from TREC 1, 2, and 3 (topics 51 through 175) on the 74,520 Wall Street Journal articles from TREC disk 2 (1990-92). Because the search engines are designed for the Web, we translated the documents into HTML. We strived to use a format suitable for display. We kept only the text that would normally appear in the original printed version, and removed additional codes such as the document numbers and manually assigned keywords. The headline appears twice in the HTML file: once in the <TITLE> section and again as an <Hl> heading. Here is an example (document WSJ900402-0195):

<html> <head> </title>
Who's News: Timken Co. </title> </head>
<body><h1>
Who's News:Timken Co.</h1> <P>
04/02/90 <P>
WALL STREET JOURNAL (J), NO PAGE CITATION
<P>
<P>
TIMKEN Co. (Canton, Ohio) Larry R. Brown, managing partner of the law firm Day, Ketterer, Raley, Wright & Rybold of Canton, Ohio, was named vice president and general counsel, a new post at this specialty steels and bearings company.
<P>
</body></html>

The queries were taken directly from the titles of TREC topics 51 through 175. Here are some examples as they were entered into each engine (minus the query number):

052: *South African Sanctions*
076: U.S. *Constitution - Original Intent*

090: Data on Proven Reserves of Oil & Natural Gas Producers

141: Japan's Handling of its Trade Surplus with the U.S.

All of the queries have from I to *19* terms (average 4.68), and 34% contained non-alphabetic characters.

5 Test Results

5.1 Data Collection

We tested EWS, SEEK, and SONY on our data, measuring precision using relevance judgements provided to us by TREC. We took at most the top 30 documents returned by each engine for each query. Out of 9212 relevant documents collection-wide, 5325 were retrieved by at least one engine (57.8% recall). Of these, 1172 were relevant (22.0% precision).

Engine	Average Retrieved	Average Relevant
EWS	30.00	7.68
SEEK	27.84	6.20
SONY	12.12	3.66

Table 1: Engine performance (limit 30 per query)

The results, by engine are shown in Table 1. There, we present the average number of documents retrieved per query by each engine, and the average number of relevant documents among retrieved top 30. The last column (average relevant) divided by 30 is the precision at 30 documents, which is 0.256 for EWS, 0.207 for SEEK, and 0.122 for SONY.

Among the engines, we note that EWS had the best performance (greatest number relevant in the top 30). SONY performed poorly because it expects the user to manually stem the query terms and remove stop words, which we did not do.

To study the correlation between engines, we counted how often a document picked by one engine was picked by the others. We found (as did Lee [Lee97]) that engines are more likely to agree when the document is relevant (Table 2).

Table 2 shows the overlap between each pair of engines. The first number in each group is the number of documents retrieved by both engines. The second is the number of those that are retrieved by either engine. The ratio of the two is the overlap. Overlap is highest for EWS and SEEK (55.7%), and lowest for EWS and SONY (13.2%). Among relevant documents, the overlap is higher in all cases, up to 64.8% for EWS and SEEK.

5.2 Engine Scores

Each engine assigns a score to each retrieved document, then ranks them from highest to lowest. For EWS and SEEK, the score is an integer from 0 to 100, with 100 meaning the best possible match.

	Average Retrieved	Average Relevant
EWS and SEEK	20.73	5.46
EWS or SEEK	37.22	8.42
Overlap (percent)	55.7	64.8
EWS and SONY	4.90	2.30
EWS or SONY	37.11	9.04
Overlap (percent)	13.2	25.4
SEEK and SONY	5.79	2.36
SEEK or SONY	34.17	7.50
Overlap (percent)	16.9	31.4
ALL	4.07	1.96
ANY	42.60	9.38
Overlap (percent)	9.6	20.9

Table 2: Engine Overlap among top 30 documents

SONY assigns a score which is the product of the term frequencies. The score can grow exponentially with the length of the query. To remove this effect, we applied the following transformation to the SONY score:

$$\text{score} = \frac{N \times \log(\text{prod} - \text{tf})}{|Q|}$$

where prod-tf is the raw SONY score (product of term frequencies), N=10 is the engine ranking normalizer, and JQJ is the number of query terms, using SONY's method of parsing terms (delimited by spaces, quotes, or parenthesis, thus "U.S.-U.S.S.R." is one term). Using this normalization, we found that the scores have the characteristics described in table 3. We measured:

- MIN and MAX, namely the lowest and highest scores (restricted to integers in the range 1 to 99).
- TOP 30 is the average score among the top 30 documents, using 0 for non retrieved documents when less than 30 documents are retrieved.
- RETR is the average score among documents retrieved, which is always 30 for EWS, but may be less for SEEK or SONY.
- REL is the average score among relevant documents retrieved.
- N-REL is the average score among non-relevant documents retrieved.

It is interesting to note that SONY scores non-relevant documents slightly higher than relevant ones. This merely indicates that comparing scores between queries is less meaningful than comparing scores within a single query.

5.3 Combining Engines - Results

We measured the precision (average number relevant) for the top 5, 10, 20, and 30 documents for each query. Then, we combined the outputs of the three engines using six different heuristics, and again measured the precision. (Table 4).

EWS, SEEK, and SONY are the three engines. CSUM and CMNZ are the combining heuristics CombSUM and CombMNZ described by Lee [Lee97]. F5 through F30 are the combining heuristics FIRE-5 through FIRE-30 that we developed. "Top n" shows the precision averaged over the queries. Precision is calculated as:

$$\text{Precision} = \frac{\text{(number relevant in top n)}}{n}$$

For all engines and combinations, we sorted the returned documents by their raw scores and assigned a ranking, I for the highest, 2 for the next highest, and so on. In cases of ties, we used an arbitrary but fixed ordering of the documents throughout our experiments to allow fair comparisons.

Tables 5, 6, and 7 compare EWS, CombMNZ, and the four FIRE heuristics, taking data from table 4. Table 5 compares EWS to FIRE-5 in the top 5, FIRE10 in the top 10, and so on. In each case, we show an improvement in precision at the indicated level. Each heuristic FIRE-n is optimized to maximize precision at level n, i.e., FIRE-5 has the best precision of the four FIRE heuristics in the top 5. Table 6 compares EWS with CombMNZ. In each case CombMNZ does

Score	MIN	MAX	TOP 30	RETR	REL	N-REL
EWS	24	98	76.44	76.44	80.04	75.20
SEEK	1	99	60.15	64.82	67.54	64.04
SONY	3	99	7.97	19.72	19.68	19.74

Table 3: Engine score characteristics

Precision	EWS	SEEK	SONY	CSUM	CMNZ	F5	F10	F20	F30
Top 5	**.357**	.285	.232	.355	.349	**.371**	.365	.366	.360
Top 10	**.329**	.266	.190	.327	.323	.333	**.334**	.322	.326
Top 20	**.288**	.235	.149	**.281**	.284	.286	.286	**.200**	.283
Top 30	**.256**	.208	.122	.244	.245	.256	.256	.256	**.259**

Table 4: Precision for three engines and 6 combining heuristics

worse. Table 7 compares CombMNZ with FIRE-n at level n. In each case, FIRE-n does better.

5.4 Heuristics for Combining Engines

CombSUM (CSUM) - Following Lee, we assigned a score of 31 - i to the i'th ranked document from the top 30 from each engine, i.e., the top document is scored 30, the second is scored 29, and so on. Any document not ranked in the top 30 is scored 0. We then added the scores together for the three engines. For instance, if a document is ranked 10'th by EWS, 25'th by SEEK, and 40'th by SONY, then its combined score is (31 - 10) + (31 - 25) + 0 = 21 + 6 + 0 = 27.

CombMNZ (CMNZ) - We assigned a total score as in CombSUM, then multiplied by the number of nonzero scores. In the above example, the document was given two nonzero scores, so its combined score is 2(21 + 6 + 0) = 54.

We found that we could improve on CombSUM and CombMNZ by using the raw scores reported by the engines, rather than a score derived from their rankings (an effect also noted by Lee). We used the raw scores from EWS and SEEK, and applied normalization to the SONY score as described in section 5.2. **The FIRE heuristic is:**

$$FIRE = W(ews) \times S(ews) + W(seek) \times S(seek) + W(sony) \frac{N \times \log S(sony)}{|Q|}$$

where S(x) is the raw score reported by engine *X*, W(x) is an experimentally determined weight, N is the engine ranking normalizer, here set to 10, and |Q| is the number of terms in the query as counted by SONY.

We experimentally found four sets of weights that maximized precision in the top 5, 10, 20, and 30 documents, and call these heuristics FIRE-5 through FIRE-30. In the last case, the weights depend on the number of query terms (again using SONY's method of counting terms) (Table 8).

Table 8 shows the weights applied to each engine for FIRE-5, 10, and 20. For FIRE-30, three sets of weights are used: one if there are less than three query terms, another if there are exactly three, and a third set if there are more than three.

In adjusting the weights to maximize precision, we found that the 3-dimensional weight space appears to be fairly smooth, without local maxima. This simplifies the search process. The weights are simply adjusted until any increase or decrease in any of the weights reduces the precision

Precision	EWS	FIRE	Diff
Top 5	.357	.371	+3.9%
Top 10	.329	.334	+1.5%
Top 20	.288	.290	+0.7%
Top 30	.256	.259	+1.2%

Table 5: Comparison of EWS and FIRE

Precision	EWS	CombMNZ	Diff
Top 5	.357	.349	-2.2%
Top 10	.329	.323	-1.9%
Top 20	.288	.284	-1.4%
Top 30	.256	.245	-4.3%

Table 6: Comparison of EWS and CombMNZ

Precision	CombMNZ	FIRE	Diff
Top 5	.349	.371	+6.3%
Top 10	.323	.334	+3.4%
Top 20	.284	.290	+2.1%
Top 30	.245	.259	+5.7%

Table 7: Comparison of CombMNZ and FIRE

	EWS	SEEK	SONY	Terms
FIRE-5	1	0	3	
FIRE-10	1	0	2	
FIRE-20	1	0	13	
FIRE-30	1	1	12	$\|Q\| < 3$
	0.5	1	12	$\|Q\| = 3$
	2	1	12	$\|Q\| > 3$

Table 8: Experimentally Determined Weights

The FIRE-30 heuristic weights the engines as a function of query length. The procedure is to partition the queries by the number of terms, optimize the weights for each set, and combine them. The rationale is that some engines do better on shorter queries, while others do better on longer queries. Without such partitioning, the optimal weights were found to be EWS = 1, SEEK = O, SONY = O, i.e. no improvement over EWS. Partitioning results in a 1.2% improvement.

Conclusion

We show that it is possible to combine multiple search engines to produce a result that is superior to any one engine, even when the individual results from the engines are good. The key difference is that we weight the engines not just by how well they perform individually, but by their dissimilarity. There is little to be gained by combining two very good engines if they both produce the same results.

Lee's CombSUM and CombMNZ heuristics work well when the top 1000 documents are available from each engine, but fail when only the top 30 axe available. (The top 1000 documents were a TREC constraint and typically require additional processing as compared to processing only the top 30 hits.) The FIRE heuristics, which use a weighted summation of similarity measures instead of an unweighted summation of rankings, produce results superior not only to the best engine, but to the best known heuristics as well. We made further improvements by making the weights a function of query length to take advantage of the strengths and weaknesses of individual engines. The superiority of combining document scores rather than rankings confirms previous work, but the relationship between optimal weights, engine performance, and engine correlation is a new result.

Acknowledgment: This work was supported in part by NSF under contract # IRI-9357785.

References

[Baum97] Christoph Baumgarten, "A Probabilistic Model for Distributed Information Retrieval", in proceeding of 20th annual ACM SIGIR conference (1997).

[EWS98] Excite for Web Servers, http://www.excite.com/navigate /home.html (Mar. 3, 1998).

[Fox98] Kevin L. Fox, Ophir Frieder, Margaret M. Knepper, and Eric J. Snowberg, "SENTINEL: A Multiple Engine Information Retrieval and Visualization System" , submitted for publication in theJournal of the American Society of Information Science, 1998.

[Gauc96] Susan Gauch and Guijun Wang, "Information Fusion with ProFusion" , in Webnet96 Proceedings, 1996.

[Gros98] David Grossman, Ophir Frieder, "Information Retrieval: Algorithm and Heuristics" Kluwer Academic Publishers, 1998.

[Gudi97] Venkat N. Gudivada, Vijay V. Raghavan, William 1. Grosky, Rajesh Kasanagottu, "Information Retrieval on the World Wide Web", IEEE Internet Computing, Sept/Oct 1997, p.58-68.

[Hahn98] Trudi Bellardo Hahn, "Text Retrieval Online: Historical Perspective on Web Search Engines", Bulletin of the American Society for Information Science, April/May 1998.

[Harm98] Donna Harman, "The Text Retrieval Conferences (TRECs): Providing a Test Bed for Information Retrieval Systems", Bulletin of the American Society for Information Science, April/May 1998.

[Harm95] Donna Harman, "Overview of the Fourth Text REtrieval Conference (TREC-4) TREC-4 Proceedings, 1995.

[Kant94] Paul B. Kantor, "Decision Level Data Fusion for Routing of Documents in the TREC3 Context: A Best Case Analysis of Worst Case Results", TREC-3 Proceedings, 1994.

[Kowa97] Gerald Kowalski, "Information Retrieval Systems, Theory and Implementation", Kluwer Academic Publishers, 1997.

[Lee97] Joon Ho Lee, "Analysis of Multiple Evidence Combination", 20th annual ACM SIGIR Conference Proceedings, 1997.

[Salt89] Gerald Salton and Michael McGill, Introduction to Modern Information Retrieval", McGraw-Hill Book Company, 1989.

[SCIE98] --- , "Web Searches Fall Short", Science News (153) p. 286, May 2, 1998.

[Shaw94] Joseph A. Shaw and Edward A. Fox, "Combination of Multiple Searches", TREC-3 Proceedings, 1994.

[SONY98] Sony Search Engine, http://src.sony.co.jp (Mar. 3, 1998).

[SEEK98] Ultraseek Server, http://software.infoseek.com/products /ultraseek/ultratop.htm (Mar. 3, 1998)

[Smea98] Alan F. Smeaton and Francis Crimmins, "Using a Data Fusion Agent for Searching the WWW", http://Iorca.compapp.dcu.ie/fusion/papers/fusion-WWW6.html (1998).

[Voor94] Ellen M. Voorhees, Narendra K. Gupta and Ben Johnson-Laird, "The Collection Fusion Problem", TREC-3 Proceedings, 1994.

[Wu97] Jack Wu, Excite Corporation, personal communication with Ophir Frieder and David Grossman, November 21, 1997.