IIT TREC-9 - Entity Based Feedback with Fusion

A. Chowdhury, S. Beitzel, E. Jensen, M. Sai-lee, D. Grossman, O.Frieder
Information Retrieval Laboratory
Department of Computer Science
Illinois Institute of Technology
Chicago, IL 60616
{abdur, beitzel, jensen, lee, grossman, frieder} @ ir.iit.edu

M.C. McCabe
Office of Advanced Analytical Tools
U.S. Government

D. Holmes
NCR Corporation
Rockville, Maryland
David.Holmes@WashingtonDC.NCR.COM

Abstract

For TREC-9, we focused on effectiveness in the web track. The key techniques we employed were information fusion, entity-based relevance feedback, Wordnet-based query parsing and a user interface designed to assist with web-based manual queries. Our initial results are positive. For the manual task, forty of fifty queries are over the median. In the adhoc, title-only task, thirty-four of fifty queries are over the median.

1. Introduction

For TREC-9 we focused on the web track, especially on in improving our effectiveness on large volumes of data. The past few TREC's we have focussed on scalability by treating the IR problem as an application of a parallel, relational database [Grossman97]. To focus on effectiveness this year, we built a new Java-based IR system called AIRE (Advanced Information Retrieval Engine). AIRE is designed to be a flexible, modular IR engine capable of state-of-the art retrieval techniques and easily modified to incorporate new proven or experimental techniques.

The keys to our effectiveness included the following approaches:

- Fusion using probabilistic (both traditional and models involving self-relevance [Robertson98, Kwok96], and vector space model with pivoted document length normalization [Singhal96].

- Entity-based relevance feedback [McCabe00].

- Improved parsing techniques of both the query and the documents.

- New user interface for manual queries.

Section 2 describes each of these approaches. Our initial results are positive. For the manual task, forty of fifty queries are over the median. In the adhoc, title-only task, thirty-four queries of fifty are over the median. More details on these results are given in Section 3. Section 4 describes directions for future work.

2. Approach to TREC-9

Our basic approach was to focus solely on the manual and adhoc parts of the web track. Given that our previous work focused on scalability, this year was a significant challenge as we started to really focus on effectiveness. We calibrated our techniques using the TREC-8 adhoc and web document collections. Early in the TREC-9 year, our best average precision was around 0.23 (this was about the average for effectiveness at TREC-8) and after applying a variety of fusion techniques we improved to roughly 0.28. At this point, we incorporated collection enrichment and then re-calibrated our fusion techniques. This left us at around 0.30. Finally, we improved our parsing and stemming algorithms using a combination of our own modified Porter stemmer and the U-Mass conflation classes [Xu96, Xu98, Pickens]. At this point we were at around .31 for the adhoc (disks 4 and 5) and .36 for the TREC-8 small web track. At about this time, the TREC-9 queries came out and we ran our best calibrated system against the TREC-9 collection. Details of our fusion techniques are given in Section 2.1, details of our parsing techniques are given in Section 2.2.

Additionally, we experimented with a means by which we could integrate information extraction with information retrieval. The idea of this technique is to use entities identified by an information extractor (we used SRA's) system and then *only* add entities in the feedback process. We submitted one adhoc run without the use of entities: *iit00td* (title + description), and *iit00t* (title) and one run with the use of entities *iit00tde* (title + description + entities). Such feedback with extraction is described in Section 2.3.

For the manual track we built a new user interface to facilitate manual query processing. Details of this user interface are given in Section 2.4.

Fusion

Prior work in fusion combined results from disparate retrieval systems [Fox94, Bartell94, Lee97]. Our approach was to provide fusion via one common system. Using a common parser, stoplist, inverted index, etc, we implemented a variety of retrieval algorithms within our framework. Thus, we avoid confusing fusion improvements with simple parsing or other system differences. We conducted numerous calibrations using the vector space model [Singhal96], Robertson's probabilistic retrieval strategy [Robertson98], and a modified vector space retrieval strategy. The following equations describe those used as the foundation of our retrieval strategies.

Robertson's Retrieval Status Value (RSV)

$$RSV = \sum_{T \in Q} w \left(\frac{(k_1 + 1)tf}{(K + tf)} \frac{(k_3 + 1)qtf}{k_3 + qtf} + k_2 |Q| \frac{avdl - dl}{avdl + dl} \right)$$

where *tf* = frequency of occurrences of the term in the document
qtf = frequency of occurrences of the term in the query
dl = document length
avdl = average document length
k_n are parameters set based on the nature of the queries and the collection.

$$\sum \log \left(\frac{N - n + .5}{n + .5} \right) \left(\frac{(2.2)tf}{.3 + (.75 * dl / avdl) + tf} \right) qtf$$

Our implementation with constants as specified by Robertson

Singhal's Similarity Coefficient

$$SC(Q, D_i) = \frac{\sum_{j=1}^{t} q_{qj} d_{ij}}{((1.0 - s)p + s(| d_i |))}$$

where $|d_i|$ is the number of elements in the vector, or the number of distinct terms in the document, s is the *slope* which Singhal calculated for a variety of test corpuses (mostly TREC subsets) and found that 0.20 works well across most collections. The pivot, *p,* is the point, or document length at which the probability of relevance equals the probability of retrieval. This is estimated to be the average document length of the collection.

In addition to fusion of various retrieval strategies, AIRE permits the fusion of different *query representations*. Also, each input run has a scalar weight that indicates the relative importance of the run.

For our first pass retrieval, we focused on finding one retrieval strategy that did better for high recall and another strategy that performed well for high precision (at 30 documents). Our hypothesis was that the combination would perform better in terms of average precision than either input run. Our initial results showed a slightly modified Vector Space did well for high recall and that Robertson's probabilistic model did the best at for precision at 30. The combination we ultimately settled on was: Modified Vector Space title-only (weighted at 1.0), Robertson title-only (weighted at 0.1), Robertson description when applicable (weighted at 0.7). This fusion combination effectively emphasized title terms as most important while still benefiting from high-recall description terms.

For our second pass, we selected the top fifteen feedback terms from the top ten documents using the fused pass one run. In order to select the top fifteen terms, we first weighted each term found in the top documents using Robertson's term weighting. We then calculated the ultimate rank of the candidate term using Rocchio's relevance feedback formula [Rocchio71]. In addition to finding the top fifteen terms and phrases, a check is made to a list of nouns obtained from Wordnet to filter candidate terms and phrases so that only nouns are selected. The new query terms are then used in pass two as one of the query representations for a fusion input run. We found a scalar weight of .5 and the Roberston retrieval strategy to work well with this query representation.

We also used a collection enrichment representation for a pass two fusion input run. This query run consisted of terms selected from a pass one retrieval executed against the TREC disks 4 and 5. The fifteen top ranked terms are then used as a query against the search collection (10GB web) with Robertson retrieval strategy and a weight of .5 (same as relevance feedback terms).

Finally, two additional runs are included in the pass two fusion. The original title terms are used with modified vector space retrieval weighted at 1.0. The original description terms are used with Robertson weighted at 1.0.

Interestingly enough, for our final TREC submission, we did not normalize the fusion runs. Thus our scalar weights represent actual multipliers rather than relative importance in pass two. This choice was made based on prior calibrations.

In summary, our *iit00t*, *iit00td* and *iit00tde* submissions were fusions of the following four different representations of a single query:

- Title words only
- Description words only (this was only used for runs involving the description)
- Relevance feedback terms obtained from running the title (and description when applicable)
- Relevance feedback terms (and entities for tde) obtained from a collection enrichment run (TREC disks 4 and 5)

Information Extraction

In previous years, our manual runs did well when the user added person and place names to queries. For example *Kuhn Sa* was very helpful on the query regarding drug triangle. This year, we propose entity-based feedback as a method to automatically select such person names, as well as place names and organization names and add them to the query. The technique required modifications to the inverted index in order to include term-type (term, phrase, person, location, etc.). Secondly, the document preprocessing was modified to include SRA's Name Tagger to identify entities within the text. Then, the original query of only terms and phrases was run for pass one. Pass two selects entities from the top documents returned and adds these terms to the query as in relevance feedback.

For our calibrations, we isolated the entities and added only person names, only locations, and only organizations to the query. In order to understand the real impact of each entity, we ran many calibrations where only one new word was added to the query. We found that good improvement is possible when we adding only a single organization, a single location, or a single person name. Details of these calibrations may be found in [McCabe00]. For example, query *Ireland Peace Talks* added the organization *Sinn Fein*, the person *Jerry Adams*, and the location *Northern Ireland.* Each of these improved the query effectiveness by over 100%. While more queries improved than degraded with each entity type, several queries degraded badly. Names that were associated with more than the query topic were harmful. For example, the query *Estonian Economics* selected the Estonian Prime Minister Mart *Laar* as the name to add. This degraded performance because the prime minister is in many documents having nothing to do with economics. In addition, ambiguous names were harmful. It turns out there are many individuals with the name *Stirling*. So that addition to the query *Stirling Engine* brought back documents about a California senator, a minister, etc.

For TREC-9 we selected entities from our collection enrichment corpus rather than our search corpus. That is simply because we already had our collection enrichment corpus (TREC disks 5 and 6) tagged and indexed with entities, while we had not yet tagged the web 10GB. In order to reduce the chances of a bad entity being selected, we added entities into the mix of potential feedback terms and only selected those that ranked in the top 15 terms or phrases. Our entity-based feedback run performed about the same as our title plus description run. This is because many queries did not select entities and of those that did some improved and some degraded, mostly canceling out the overall effect.

Parsing

We improved our parsing algorithms in TREC-9. Previously, we did not use any stemming. This year, we indexed terms with a modified Porter stemmer that does both prefix and suffix stemming. In addition, we use equivalence classes based on term co-occurrence to further restrict the stemming [Xu96,Pickens, Xu98]. If the term is found in the conflation file (a set of terms with their equivalence classes), we use the first occurrence of the term as the root form. If the term is not found, the modified porter stemmer is used. This technique corrects for over-stemming of common words. For example the standard Porter stemmer would conflate *policy* and *police* while these equivalence classes would not.

In addition to single terms, our parser indexes standard statistical two-term phrases. Like numerous groups over the years, a sliding two-term window is used to detect these phrases. Any span punctuation or stop term prevents a phrase. We also eliminate phrases that do not occur more than 25 times.

In order to update our parser to accommodate web-type queries such as misspelled and mis-spaced terms, we incorporated a “find a real query term” algorithm using Wordnet. Our algorithm finds the longest common sub-string match in the query that is also a noun in Wordnet. If the initial query found no documents, we use this algorithm as an automatic best guess approach.

Manual Query Processing

Our previous years at TREC have shown that a user who is given the ability to add related terms to a “concept” for a query is able to improve effectiveness. Our query expert now has six years of experience with TREC queries and is quite comfortable with defining terms for a query. About 5-10 minutes are spent on each query in which two different concepts are defined for “inclusion” into the query and one is defined for “exclusion”. The ultimate query may be expressed such that if terms in the set C1 {c_{11}, c_{12}, …., c_{1n}} and the set C2 {c_{21}, c_{22}, …, c_{2n}) are included and terms in the set X {x_1, x_2, …, x_n} are excluded, the following Boolean processing is done on the query: ((c_{11} OR c_{12} OR …., OR c_{1n}) AND (c_{21} OR c_{22} OR ….OR c_{2n})) NOT (X_1 or X_2 or X_n). Additionally, for other related terms that are not used to filter a document a scoring concept is used. For these terms S {s_1, s_2, …, s_n}, only the similarity measure is affected – these terms are not used to filter the document. Once the Boolean filters are incorporated, standard tf-idf VSM is employed to rank documents.

A Java servlet is used to provide quick feedback to the user. For each initial request, the user quickly views documents obtained in response to the request. Additionally, relevance feedback terms and phrases are suggested. Overall, our test user was quite pleased with the new user interface (only command line SQL processing has been available in previous years).

3. Results

We describe our adhoc results first, then our manual results, and finally we give some initial failure analysis.

Adhoc

Our title-only run was called *iit00t* and our title with description run was named *iit00td*. The run which used named entities as part of the collection enrichment was entitled *iit00tde*. The table below gives a summary of our results. The columns indicate the average precision for the median of all groups, IIT's average precision, the number of queries at or above the median, the number of queries below the median, the number of queries that gave the best results, and the number of queries that gave the worst results.

Run	**Avg. Median**	**IIT Avg. Precision**	**# Above Median**	**# At Median**	**# Below Median**	**# Best**	**# Worst**
iit00t	.1212	.1627	30	2	18	5	4
iit00td	.1554	.2227	38	1	11	3	0
iit00tde	.1554	.2293	37	2	11	4	1

Manual

For the manual query track, we had promising results. With all but seven queries over the median and twenty-five queries listed as achieving the highest average precision, we are pleased with this run. Unfortunately, one query was found to be the worst.

Run	**Avg. Median**	**IIT Avg. Precision**	**# Above Median**	**# At Median**	**# Below Median**	**# Best**	**# Worst**
iit00m	.1350	.3519	40	3	7	25	1

Failure Analysis

In failure analysis, we review poor performing queries and analyze the cause of failure. We started some failure analysis for TREC-9. The manual query with the worst average precision (0.000) was topic 485 which simply contained the terms "*gps clock*". For this query, there were only two relevant documents and we did not find either of them in our top 100. The reason is that we added numerous synonyms for gps and for clock and they overshadowed the basic phrase "gps clock." Worse, our adhoc system stemmed "gps clock" to "gp clock." Once we stemmed "gps" to "gp" we found documents about "general permit" (Document wtx082-b37-24) and *grand* *prix,* hockey statistics for *games played*, etc. A simple rule that precluded stemming three character terms would have improved this run tremendously. It is not clear how we would know that our manual run could be improved, but one approach might be to simply run the manual query and fuse it with the entire original query.

For topic 495, *Where can I find information on the decade of the 1920's?*, our user tried to think of events in the 1920's that would be of interest (e.g.; Charles Lindbergh, Calvin Coolidge, etc.). Unfortunately, he missed many useful events in the 1920's and missed some relevant documents.

In addition, our analysis indicates that our use of a scoring concept in the manual runs hurt us for some queries. It may well be helpful to use fusion to combine a query run with the scoring concept and without the scoring concept so as to ensure that high scoring documents without the scoring concept are included in the final result set.

4. Summary and Future Work

Overall, we are pleased with our work on effectiveness this year. We plan to spend more time on failure analysis. Additionally, more cleanup of our parser is needed. More importantly, the potential of entity-based relevance feedback needs more research.

5. References

[Bartell94] Bartell, B. T., G.W. Cottrell, and R.K. Belew. Automatic combination of multiple ranked retrieval systems. *SIGIR '94: Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, 1994.

[Croft95] W. B. Croft and Jinxi Xu. Corpus-specific stemming using word form co-occurence. *In Proceedings for the Fourth Annual Symposium on Document Analysis and Information Retrieval*, pages 147--159, Las Vegas, Nevada, April 1995.

[Fox94] Fox, E. and J. Shaw. "Combination of Multiple Searches, *Proceedings of the 2nd Text Retrieval Conference (TREC2),*National Institute of Standards and Technology Special Publication 500-215, 1994.

[Gross97] Grossman, D., O. Frieder, D. Holmes and D. Roberts, Integrating Structured Data and Text: A Relational Approach, *Journal of the American Society for Information Science*, January 1997.

[Kwok96] Kwok, K.L (1996). A new method of weighting query terms for ad-hoc retrieval. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* pp.187-195.

[McCabe2000] McCabe, M.C. *Improving Information Retrieval Effectiveness with Databases, Fusion and Entity-Based Feedback*. GMU PhD thesis. Aug.2000.

[Lee97] Lee, J.H. Analysis of multiple evidence combination. *SIGIR '97: Proceedings of the Twentieth Annual InternationalACM-SIGIR Conference on Research and Development in Information Retrieval*, 1997.

[Porter80] Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, 14(3):130--137.

[Robertson98] Robertson S., S. Walker and M. Beaulieu, Okapi at TREC-7: Automatic ad hoc, filtering, VLC and Iinteractive. *In Proceedings of the Seventh Text Retrieval Conference (TREC)*7, 1998.

[Singhal96] Singhal, A., C. Buckley, and M. Mitra, Pivoted Document Length Normalization. In *Proceedings of the Nineteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1996.

[Xu98] Jinxi Xu and W. Bruce Croft. Corpus-based stemming using co-occurrence of word variants. Technical Report TR96-67, Dept. of Computer Science, University of Massachusetts/Amherst.

[Pickens] Jeremy Pickens, "Stemming and Cooccurrence on a Larger Corpus"

[Rocchio71] J. Rocchio. Relevance Feedback in Information Retrieval. *Smart System - Experiments in Automatic Document Processing*, pages 313--323. Prentice Hall, Englewood Cliffs, NJ, 1971.

[Xu96] J. Xu and W. Croft, Query Expansion Using Local and Global Document Analysis. In *Proceedings of the Nineteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 4--11, 1996.