Next: Results and Discussion **Up:** A Case Study in **Previous:** Collection of Pages

TREC Ad-Hoc Algorithm

Different groups participating in TREC have developed several ad-hoc algorithms over years. Most groups have their own expertise built into these algorithms. An analysis of some of the best performing TREC algorithms shows that the top ad-hoc algorithms at TREC have the following two common features: [23,24]

1. Most of the top performing systems use a modern term weighting method developed in either the Okapi system [12,13] or the SMART system [19,20].

2. Most groups use a two-pass pseudo-feedback based query-expansion approach. In this approach a first pass retrieval is done to find a set of top (say) 10 or 20 documents related to the query, the query is expanded by adding new words/phrases from these documents using relevance feedback [14,15], and this expanded query is used to generate the final ranking in a second-pass retrieval.

Each participating group has its own twist on the above two components. For example, several groups use collection enrichment [10], in which a much larger document collection is used in the first pass (instead of the target collection) to locate documents for use in the query-expansion process. In yet another enhancement, several groups assume that poorly ranked documents from the first pass are not relevant to the query and use this evidence of non-relevance in the query expansion process [20].

Table 1: Term Weighting Schemes

d tf factor:

$$1 + ln(1 + ln(tf)) \qquad 0 \ if \ tf = 0$$

t idf factor:

$$log(\frac{N+1}{df})$$

b pivoted byte length normalization factor:

$$\frac{1}{0.8 + 0.2 \times \frac{length\ of\ document\ (in\ bytes)}{average\ document\ length\ (in\ bytes)}}$$

tf is the term's frequency in text (query/document)
N is the total number of documents in the collection
df is the number of documents that contain the term, and
the average document length depends on the collection.

dnb weighting: **d** factor × **b** factor
dtb weighting: **d** factor × **t** factor ×**b** factor
dtn weighting: **d** factor × **t** factor

We implement an algorithm which is a scaled-down version of the ad-hoc algorithm used by Singhal et al. in [20]. As described in [23], this algorithm was one of the best performing ad-hoc algorithms at

TREC-7. Here are the steps implemented in our algorithm.

- **Pass-1**: Using *dtn* queries and *dnb* documents, a first-pass retrieval is done (see Table 1 for an explanation of this term-weighting jargon).
- *Expansion:* Top ten (distinct) documents retrieved in the first pass are *assumed* to be relevant to the query. Rocchio's method (with parameters $\alpha = 1.0,\ \beta = 0.5$, and the γ factor is not needed here since we do not assume any documents as non-relevant) is used to expand the query by adding twenty new words with highest Rocchio weights [14]. To include the *idf*-factor in the expansion process, documents are *dtb* weighted.
- **Pass-2:** The expanded query is used with *dnb* documents to generate the final ranking.

Since web collections do have a reasonable number of duplicate documents, to do the query expansion well for the web collections, we have observed that we need to eliminate duplicate documents from the top ten documents used for query expansion. To do this, we retrieve top 100 documents in the first pass, and starting from rank 2 we test if a retrieved document is a duplicate of a previously ranked document. If it is, we remove it from the list. We do this until we get ten distinct documents. Two documents are considered duplicates of each other if they share more than 70% of their vocabulary. We have found this to be a reasonable heuristic for web pages.

To test that our implementation of this TREC ad-hoc algorithm is not broken and is indeed state-of-the-art, we run our system on two recent TREC ad-hoc tasks and compare its precision to the best performing systems at TREC. Since most web queries are short, we want to evaluate the system performance for short queries, and only use the 2-3 words *title* portion of the TREC queries. Our objective in this study is to do a precision oriented evaluation, we only compare systems based the precision in top ranks. We compare the precision of our system at rank 10 and at rank 20 to corresponding values for the five best performing systems at TREC using title-only queries (these values are available from the detailed results presented in the TREC proceedings, see Appendix A in [21] and [22]). The results are shown in Tables 2 and 3.

Table 2: Precision at 10 and 20, TREC-7 ad-hoc task

System	**P@10**	System	**P@20**
ok7as	48.6%	ok7as	42.5%
OUR Implementation	**46.8%**	LNaTit7	39.1%
LNaTit7	46.2%	**OUR Implementation**	**38.8%**
pirc8At	44.8%	pirc8At	37.7%
att98atc	44.2%	FLab7at	37.5%
FLab7at	42.8%	att98atc	36.3%

Table 3: Precision at 10 and 20, TREC-8 ad-hoc task

System	**P@10**	System	**P@20**
ok8asxc	48.8%	pir9At0	44.1%

FLab8at	48.6%	FLab8at	42.6%
uwmt8a1	48.2%	uwmt8a1	42.5%
OUR Implementation	**48.0%**	**OUR Implementation**	**42.4%**
pir9At0	48.0%	att99ate	42.0%
att99ate	47.6%	ok8asxc	41.6%

Tables 2 and 3 show the precision value for the best TREC systems ordered by decreasing performance. Inserted in that order, is the corresponding precision value for our system. These results show that our system, motivated by a state-of-the-art TREC ad-hoc algorithm is quite competitive with the top performing TREC systems. This is especially true considering the performance gap between the best and the fifth-best system is not very significant. For example, Table 2 indicates that for the TREC-7 ad-hoc task, the best performing system `ok7as` retrieves on an average 4.86 relevant documents in top 10 for a query, whereas the fifth-best performing system retrieves 4.28. That difference is not very large from a user's perspective.

In summary, these results verify that our implementation of a modern TREC ad-hoc algorithm is not broken and is indeed state-of-the-art. It will be reasonable to say that this system, when run over our fresh web collection, would produce results that will be quite comparable to the results produced by any other top TREC ad-hoc system.

Next: Results and Discussion **Up:** A Case Study in **Previous:** Collection of Pages
Amit Singhal 2001-02-18