SMART in TREC 8

Chris Buckley*, Janet Walz*,

Abstract

This year was a light year for the Smart Information Retrieval Project at SabIR Research and Cornell. We officially participated in only the Ad-hoc Task and the Query Track. In the Ad-hoc Task, we made minor modifications to our document weighting schemes to emphasize high-precision searches on shorter queries. This proved only mildly successful; the top relevant document was retrieved higher, but the rest of the retrieval tended to be hurt. Our Query Track runs are described here, but the much more interesting analysis of these runs is described in the Query Track Overview.

Basic Indexing and Retrieval

In the Smart system, the vector-processing model of retrieval is used to transform both the available information requests as well as the stored documents into vectors of the form:

$$D_i = (w_{i1}, w_{i2}, \ldots, w_{it})$$

where D_i represents a document (or query) text and w_{ik} is the weight of term T_k in document D_i. A weight of zero is used for terms that are absent from a particular document, and positive weights characterize terms actually assigned. The assumption is that t terms in all are available for the representation of the information.

The basic "tf*idf" weighting schemes used within SMART have been discussed many times. For TREC 8 we made a slight modification to Lnu-ltu weights we have used in the past 4 years in TREC 4–7. We noticed that the pivoted byte-length document normalization used by Singhal et al in TREC 7([5]) seems to favor high precision searches when used with short queries. It is a bit more biased towards shorter documents than our previous "u" scheme which uses number of unique terms in the document, thus a good short document containing all the query terms will be ranked highly. We hoped that this would enable our blind feedback query expansion to be based on more relevant documents and thus improved.

The same phrase strategy (and phrases) used in all previous TRECs (for example [2, 3, 4, 1]) are used for TREC 8. Any pair of adjacent non-stopwords is regarded as a potential phrase. The final list of phrases is composed of those pairs of words occurring in 25 or more documents of the initial TREC 1 document set. Phrases are weighted with the same scheme as single terms. Note that no human expertise in the subject matter is required for either the initial collection creation, or the actual query formulation.

When the text of document D_i is represented by a vector of the form $(d_{i1}, d_{i2}, \ldots, d_{it})$ and query Q_j by the vector $(q_{j1}, q_{j2}, \ldots, q_{jt})$, a similarity (S) computation between the two items can conveniently be obtained as the inner product between corresponding weighted term vectors as follows:

$$S(D_i, Q_j) = \sum_{k=1}^{t} (d_{ik} * q_{jk}) \tag{1}$$

Thus, the similarity between two texts (whether query or document) depends on the weights of coinciding terms in the two vectors.

*SabIR Research, Inc.

The Cornell TREC experiments use the SMART Information Retrieval System, Version 13.3, and most were run on a dedicated Intel 350 Mhz Pentium running Linux, with 128 Megabytes of memory and 54 Gigabytes of local disk.

SMART Version 13 is the latest in a long line of experimental information retrieval systems, dating back over 30 years, developed under the guidance of G. Salton. The new version is approximately 48,000 lines of C code and documentation.

SMART is highly flexible and very fast, thus providing an ideal platform for information retrieval experimentation. Documents for TREC 8 are indexed at a rate of about 2 Gigabytes an hour, on hardware costing under $2,000 new. Retrieval speed is similarly fast, with basic simple searches taking much less than a second a query.

Ad-hoc Task

The basic approach we used for this year's TREC ad-hoc task is almost identical to our TREC 5 approach. We only used one algorithm (no experimental algorithm this year!), and ran it on 4 different topic lengths. Our TREC 5 paper [1] gives the details and rationale for the approach. The only important difference is that we used the Lnb weighted documents described above instead of the Lnu documents of TREC 5.

The basic algorithm is

1. Retrieve 1000 documents using the initial query (using *Lnb.ltu* weights).
2. Generate cooccurrence information about the query terms from the top 1000 documents.
3. Rerank the top 50 documents as in TREC 5, using correlation between query terms and other terms, as well as proximity information of the query terms.
4. Assume the top 20 documents relevant, documents ranked 501–1000 non-relevant.
5. Expand the query by 25 words and 5 phrases using Rocchio expansion with $\alpha = 8$, $\beta = 8$, and $\gamma = 8$.
6. Retrieve the final set of 1000 documents using the expanded query.

Ad-Hoc experiments and analysis

We submitted four runs in the ad-hoc category, all using the same algorithm. Sab8A1 used only the title field of the topic, Sab8A2 used only the description field, Sab8A3 used all three fields, and Sab8A4 used the title and description field (this last run being the requested "official" ad-hoc run).

Table 1 shows the results for the various runs across 50 queries. Results are all quite close to each other, with noticeable disagreements between the various evaluation measures.

Run	Average precision	Total rel retrieved	R precision	Precision @100 docs	Precision @Rcl(0)
Sab8A1(t)	.2553	3006	.2901	.2376	.7360
Sab8A2(d)	.2407	2829	.2829	.2228	.7988
Sab8A3(tdn)	.2546	2957	.3088	.2316	.8514
Sab8A4(td)	.2608	2986	.3021	.2384	.7860

Table 1: Ad-Hoc results (50 queries)

Table 2 shows that our runs are reasonably mediocre when compared with other runs using Average Precision. This is a disappointing performance; it is clear that the attempt to increase emphasize high precision in the top retrieved documents by the use of Lnb document weights did not end up helping our expansion and therefore overall results.

Our approach does have a positive effect at the very high-end. Using Precision at Recall (0), which basically measures precision of the top retrieved document, our runs do considerably better than they do when evaluated using Average Precision. The .8514 figure for Sab8A3 in Table 1 is the second best figure among all automatic ad-hoc TREC runs, a mere 0.3% worse than the leading run. However, the Average Precision for Sab8A3 is a whopping 23.7% worse than the Average Precision for that same run. Given our high-end performance, there may be environments where it would be appropriate to use the algorithms here, but it is clear they should not be used for general retrieval.

Run	Task pool	Best	$\geq$ median
Sab8A1	title	4	28
Sab8A2	title-desc	0	30
Sab8A3	entire	0	25
Sab8A4	title-desc	0	36

Table 2: Comparative automatic ad-hoc results (Av-Prec 50 queries)

Query Track

The Query Track is a bit different from other TREC tracks in that the individual participants all contribute data into a common pool for later analysis, rather than be evaluated separately. It is an attempt to examine the variability of queries by getting multiple query variations of past TREC topics and running them on different systems. The participants contribute their query variations, then run the variations from all other participants, and finally attempt to analyze the results.

The Query Track Overview in this proceedings gives the overall goals and procedures for the track, and gives all the analysis of the pooled results done so far. This section merely goes into the details of our contributions to the common pool of data and assumes the reader has background knowledge of the task itself.

Query Variations

We contributed 4 versions of each of the 50 topics of the TREC 1 Ad-hoc Task. All versions were created by the same person; an expert system designer of SMART. For each topic

1. Sab3a: The user looked at the results of a retrieval (done with the SMART TREC 4 algorithm, see below) using that topic as a query. In general, 3 or 4 relevant documents were looked at. Occasionally, some of the non-relevant retrieved documents were also examined. Just given these documents, a one sentence query was formed. The topic itself was not examined.
2. Sab1b: After constructing the sentence above, the user then constructed a 2-3 word short query, again before looking at the topic.
3. Sab1a: The user constructed a 2-3 word short query after looking at the topic. The user had memories of the relevant documents seen in the construction of the above two query sets.
4. Sab1c: An automatic blind feedback run was made on the original topics. The 2-3 most highly weighted terms/phrases in the expanded query were manually de-stemmed and formed into a query.

Including everything, it took 5 hours to construct the 4 query sets above. About 2 hours of that was setting up and running the indexing and retrievals on the Query Track learning and test sets.

Retrieval Variations

There were a total of 23 query sets (each composed of 50 queries) submitted by the participating groups. 21 of those were natural language and 2 were expanded lists of weighted terms.

We ran three different retrieval algorithms on each of the 23 query sets. Each algorithm used the basic SMART TREC 4 approach:

- Documents indexed with adjacency phrases and weighted with "Lnu" scheme.
- Queries indexed with adjacency phrases and weighted with "ltu" scheme (The two weighted term query sets used the given weights and did not have any phrases added.)
- If expansion terms needed, perform blind relevance feedback assuming the top 20 retrieved documents are relevant. Add and reweight terms.
- Run (possibly reweighted) queries against test documents, retrieving 1000 documents, and submit run to NIST.

The three algorithms were deliberately kept simple to aid in the later analysis, and so there would be no problems running all of the needed runs. None of the algorithms represent the best that we can do. The three run approaches are:

1. Saba: Index the query using terms from the query plus any adjacency phrases occurring in it.
2. Sabm: Moderate blind feedback expansion. Add 5 single terms and 2 adjacency phrases from the top documents.
3. Sabe: Blind feedback expansion. Add 50 single terms and 10 adjacency phrases from the top documents.

We turned in a total of 69 retrieval run results to NIST (23 query sets * 3 run approaches). In September we got the results from all the participants (about 450 Mbytes of results) and started our analysis! Analysis from all the groups is presented in the Query Track Overview.

Comparison with past TREC's

We performed our annual comparison of how TREC and our systems have varied over the years. We ran our 8 TREC SMART systems against each of the 8 TREC ad-hoc tasks.

Table 3 gives the results. Note that the indexing of the collections has changed slightly over the years so results may not be exactly what got reported in previous years. In the interest of speed, we ran our current implementation of the query and document indexing and weighting.

Comparing the columns of Table 3 gives an indication of how the difficulty of TREC task has changed over the 8 years of TREC. For example, eight different versions of the same system all do from 45% to 65% worse, in absolute numbers, on the TREC 7 task as compared to the TREC 1 task. The TREC 1 and TREC 2 figures are about the same. Performance starts to drop in TREC 3 and 4 when the queries get progressively shorter. The short high-level queries of the last 4 TRECs prove more difficult for all versions of SMART. All the methods agree that the TREC 8 task is a bit easier than any of the TREC 5–7 tasks.

Looking at other evaluation measures (not given here) run on the same 8x8 grid, we get confirmation that our TREC 8 approach does well at the high end of the retrieval ranking while not doing well overall. For the TRECs with short queries, TREC 5–8, the TREC 8 approach had the highest evaluation numbers for measures Precision(5), Precision(10), Precision at Recall(0), Precision at Recall(.10), beating all other 7 approaches. But Average Precision as given in Table 3) is the lowest in the past 4 years for all tasks except TREC 6. Other measures such as the total relevant retrieved agree with Average Precision

Methodology and Run	TREC 1 Task	TREC 2 Task	TREC 3 Task	TREC 4 Task	TREC 5 Short	TREC 6 DESC	TREC 7 DESC	TREC 8 TI-DES
TREC 1: ntc.ntc	.2442	.2615	.2099	.1533	.1048	.0997	.1137	.1412
TREC 2: lnc.ltc	.3056	.3344	.2828	.1762	.1111	.1125	.1258	.1846
TREC 3: lnc.ltc-Exp	.3400	.3512	.3219	.2124	.1287	.1242	.1679	.2102
TREC 4: Lnu.ltu-Exp	.3628	.3718	.3812	.2773	.1842	.1807	.2262	.2436
TREC 5: Exp-rerank	.3759	.3832	.3985	.3128	.2047	.1844	.2543	.2629
TREC 6: Rrk-clust	.3711	.3779	.4014	.3037	.2031	.1768	.2512	.2654
TREC 7: Rrk-clust	.3779	.3837	.4002	.3137	.2116	.1804	.2543	.2679
TREC 8: Lnb	.3563	.3623	.3647	.2836	.1997	.1857	.2282	.2608

Table 3: Comparisons of past SMART approaches with present

Cross-Language

We ended up submitting one unofficial run to the Cross-language Track. The NIST organizers asked us to submit an Italian query on Italian documents run in order to augment the Italian language pool for the full Cross-language task (there were extremely few Italian documents retrieved by any of the groups.) We discuss it here just to document where those relevant documents came from.

Words were stemmed with our multi-lingual stemmer, which includes a very few Italian specific rules:

- Removes initial "all'", "d'", "dall'", "dell'", "l'", "nell'", "quest'", "sull'", "un'".
- Removes final "a", "e", "i", "o", "mente".

Documents and queries were indexed with single terms, but no phrases.

The retrieval algorithm used is exactly the same as the TREC4 algorithm used for the Query Track Sabe run described above, except no adjacency phrases were used. Again, a simple run was deemed the best.

NIST requested the run of us since we could supply it quickly. Indeed, it took us a bit under 2 hours, which included fetching the queries via ftp, setting up and indexing the documents, doing the retrieval and ftp-ing the results to NIST.

Conclusion

We participated in TREC 8, though at a slightly lower level than in previous years. Much of the interesting work we did for TREC 8 appears in the Query Track Overview.

Our ad-hoc runs this year turned out fairly mediocre. We tried to develop a high-precision approach in the hopes that blind feedback query expansion would then perform well, but we were unsuccessful. Our approach did result in good performance at the high end, but our overall performance was poor. We did not retrieve nearly as many total relevant documents as either other TREC 8 groups did, or as we would have retrieved if we had used one of our systems from the past few TRECs.

References

[1] Chris Buckley, Mandar Mitra, Janet Walz, and Claire Cardie. Using clustering and superconcepts within SMART : TREC 6. In E. M. Voorhees and D. K. Harman, editors, *The Sixth Text REtrieval Conference (TREC-6)*. NIST Special Publication 500-240, 1998.

[2] Chris Buckley, Gerard Salton, and James Allan. Automatic retrieval with locality information using SMART. In D. K. Harman, editor, *Proceedings of the First Text REtrieval Conference (TREC-1)*, pages 59–72. NIST Special Publication 500-207, March 1993.

[3] Chris Buckley, Amit Singhal, and Mandar Mitra. New retrieval approaches using SMART : TREC 4. In D. K. Harman, editor, *Proceedings of the Fourth Text REtrieval Conference (TREC-4)*. NIST Special Publication 500-236, 1996.

[4] Chris Buckley, Amit Singhal, and Mandar Mitra. Using query zoning and correlation within SMART : TREC 5. In D. K. Harman, editor, *Proceedings of the Fifth Text REtrieval Conference (TREC-5)*. NIST Special Publication 500-238, 1997.

[5] Amit Singhal, John Choi, Donald Hindle, David D. Lewis, and Fernando Pereira. AT&T at TREC-7. In E. M. Voorhees and D. K. Harman, editors, *The Seventh Text REtrieval Conference (TREC-7)*. NIST Special Publication 500-242, 1999.