TREC-6 Ad-Hoc Retrieval

Martin Franz, Salim Roukos
<franzm|roukos>@watson.ibm.com
IBM T. J. Watson Research Center,
POB 718
Yorktown Heights, NY 10598

1 Introduction

In TREC-6 ad-hoc experiments we used multi-pass strategy, based on improving the document scores obtained from the Okapi formula [1] by combining them with scores produced by expanded queries, constructed automatically using top ranking documents from the first pass. We have examined various ways of creating expanded sets, as well as computing the scores for words and word pairs contained in them. An application of the same algorithms in the context of TREC-6 Very Large Corpus was also tested.

2 Data Preprocessing

The description fields of the queries and content bearing fields of the documents were filtered and tokenized using a statistical tokenizer. After that the texts of both queries and documents were processed using a morphological analyzer. The analyzer uses part-of-speech tags, obtained from a statistical tagger [2]. Based on the spelling and the word tag, the canonical form of word was found by table lookup. We map 71,472 word forms into 31,434 word stems. The words not contained in the table were kept in their original form. All the words were case-folded after the morphological analysis was done. Hyphenated words were then split into their components.

Before the n-grams were collected for the query sentences, there was a filtering done to remove the common query prefixes, such as "A relevant document would discuss". The filter used a table of such prefixes collected from the previous TREC query sets. All of them and their prefixes were removed if they occurred at the beginning of a query sentence. Such filtering was done for TREC-6 experiments only.

3 Collecting n-grams

Unigram and bigram counts were used for experiments described in this paper. Bigram counts were collected only for the directly neighboring word pairs (phrases), the word order was considered significant. Counts were not collected for the words in a stopword list, containing 514 items.

4 First Pass Scoring

Standard Okapi formula [1] was applied in the first-pass ranking. Each unigram and bigram term in the intersection of the query and document term lists contributed a score of:

$$s = \frac{tf}{c_1 + c_2 \times \frac{dl}{avdl} + tf} \times w^{(1)} \times qtf, \tag{1}$$

where tf and qtf are the document and query counts for a given term, dl is the length of the document, $avdl$ is the average length of the documents in the corpus, $w^{(1)}$ is the inverse document frequency, computed as:

$$w^{(1)} = log(\frac{N - n + 0.5}{n + 0.5}),$$

where N is the total number of documents in the corpus and n is the number of documents containing a given term. In the Eq.(1) we used $c_1 = 0.5, c_2 = 1.5$ for unigram scoring and $c_1 = 0.05, c_2 = 0.05$ for the bigrams. We also decided to set $qtf = 1$ based on experiments with TREC-4. The first pass score was a linear combination of unigram and bigram scores given by Eq.(1), with the unigram scores weight set to 0.8 and bigram scores weight equal to 0.2. The results of these experiments are summarized in Table 1, lines 1 and 2.

5 Query Expansion

We assumed the top 40 documents for each query as ranked by the first pass to be relevant. The documents from these sets were used to establish the new, expanded queries, which were later applied to obtain the second pass scores.

To decide which unigrams should be included in expanded queries, we used a probabilistic model described in [3]. The set of expanded unigrams contained the words for which the summation of the probabilistic model scores for the top 40 documents was above the 20% of the total score for the highest ranking word.

We also tried a second method for query expansion by using the number of documents containing a given word as an indication whether the word should be included in the expanded query. In this case the word had to occur in at least 20 of the top 40 documents.

Bigrams in the expanded sets were the ones contained in at least 15 of the top 40 documents as ranked by the first pass.

The expanded queries were applied to the document n-grams using Eq.(1) and unigram and bigrams scores were combined the same way as in the first pass. Second pass scores were normalized and combined linearly with normalized first pass scores, using weight set to 0.8 for the first pass and 0.2 for the second pass. The results of these test runs are listed in Table 1, lines 5 and 6.

We also experimented with using the probabilistic model scores directly, both applying the scores of all the words in the document and using only the words from the original and expanded queries, using the probabilistic model scores to decide about the query expansion. The bigram scores were obtained the same way as described above. The results of these experiments may be found in Table 1, lines 3 and 4.

6 Third Pass Scoring

Based on the results of experiments on TREC-4 and TREC-5, we decided to use a three pass strategy for our TREC-6 submission. We trained the probabilistic model using the top 40 documents as ranked by the combination of the first two passes. Expanded queries were created by selecting the unigrams for which the summation of probabilistic model scores for the top 40 documents was above the 20% of the total score for the highest ranking word and bigrams contained in at least 15 of the top 40 documents (i.e. the same way as in the second pass). Third pass ranks were obtained by adding the probabilistic model scores for the n-grams from the original and expanded queries. The final score is a linear combination of the first two passes combined and the third pass, using the weight ratio 90/10. The bottom line of Table 1 summarizes the three pass rescoring results.

			TREC-4		TREC-5		TREC-6	
p1	p2	p3	AveP	P20	AveP	P20	AveP	P20
u,o	–	–	0.2049	0.4030	0.1645	0.2690	0.1697	0.2850
b,o	–	–	0.2275	0.4230	0.1769	0.3050	0.1769	0.3050
b,o	b,ps	–	0.2470	0.4470	0.1846	0.2860	0.1801	0.3030
b,o	b,px,ps	–	0.2513	0.4420	0.1904	0.2910	0.1861	0.2980
b,o	b,cx,o	–	0.2521	0.4500	0.1856	0.2854	0.1788	0.3120
b,o	b,px,o	–	0.2648	0.4660	0.1925	0.2930	0.1819	0.3040
b,o	b,px,o	b,px,ps	0.2695[1]	0.4600	0.1946	0.3050	0.1775[2]	0.2930

u: unigram terms
b: unigram and bigram terms
o: Okapi formula used for scoring
ps: probabilistic model used for scoring
px: probabilistic model used for query expansion
cx: word frequencies used for query expansion
[1] the relevant set used for this third pass run was not obtained exactly the same way as it was done for TREC-5 and TREC-6 experiments, but by using slightly different query expansion scheme, yielding AveP = 0.2652
[2] official TREC-6 result submitted as ibms97a

Table 1: Results of experiments on TREC-4, TREC-5 and TREC-6 ad-hoc.

		baseline		VLC	
pass1	pass2	AveP	P20	AveP	P20
u,o	–	0.0526	0.2670	0.1976	0.3630
b,o	–	0.0564	0.2780	0.2052	0.3620
b,o	b,px,o	0.0626	0.2930	0.2116	0.3860

u: unigram terms
b: unigram and bigram terms
o: Okapi formula used for scoring
px: probabilistic model used for query expansion

Table 2: Results of experiments on TREC-6 VLC.

7 Very Large Corpus

We applied a system similar to the second pass of the ad-hoc run in our VLC experiment, using the probabilistic model for query expansion and Okapi formula to obtain the term scores. The only difference was that the expanded queries were constructed using the documents from CD4 and CD5 (i.e. TREC-6 ad-hoc set) only. This strategy made it possible to run the scoring in a single pass fashion, collecting scores for the original and the expanded queries simultaneously. Unfortunately, the code used for VLC scoring contained an error causing some unigrams and bigrams from the original queries being dropped from the processing. VLC results, summarized in Table 2, contain the average precision and precision at top 20 results obtained using the system after correcting the above mentioned problem, with the original official relevance judgements. The numbers given should be thus viewed as the lower estimate of the real system performance.*

8 Conclusion

We have experimented with various query expansion and scoring algorithms in the context of TREC-4, TREC-5 and TREC-6 tasks. All of the multi-pass strategies improved the average precision as compared to the first pass results. The combination of a probabilistic model used to select the ex-

*We used the query description fields only (as opposed to title and description fields) in our VLC experiments.

panded query words and Okapi rescoring yielded the most significant improvement for TREC-4 and TREC-5, while using probabilistic model based query expansion with probabilistic model scoring was the best in case of TREC-6. Applying a third pass caused a slight average precision improvement in TREC-4 and TREC-5 and relatively large degradation in TREC-6. The overall effect of rescoring diminishes as moving from TREC-4 (our development test set) to TREC-5 and again between TREC-5 and TREC-6.

The Very Large Corpus results were obtained using the first two passes of our ad-hoc system.

9 Acknowledgements

This work is supported by NIST grant no. 70NANB5H1174. We would like to thank Jerome Gros for implementing the statistical tokenizer and Robert T. Ward for contributing to several phases of morphological processing.

References

[1] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, M. Gatford 1995. Okapi at TREC-3. In D. K. Harman, editor, *Proceedings of the Third Text REtrieval Conference (TREC-3).* NIST Special Publication 500-225, 1995.

[2] B. Merialdo 1990 Tagging text with a probabilistic model. In *Proceedings of the IBM Natural Language ITL,* Paris, France, pp. 161-172.

[3] E. P. Chan, S. Garcia, S. Roukos, (to be submitted) A Probabilistic Model for Information Retrieval