TREC-5 Ad Hoc Retrieval Using K Nearest-Neighbors Re-Scoring

Ernest P. Chan, Santiago Garcia and Salim Roukos
<epchan|sgarcia|roukos>@watson.ibm.com
IBM T. J. Watson Research Center,
POB 718
Yorktown Heights, NY 10598

April 22, 1997

1 Introduction

In our first participation in TREC, we focus on improving on baseline results obtained from another search engine by means of automatic query expansion. We call the specific formula we used for query expansion “Knn re-scoring”, where “Knn” stands for “K nearest-neighbors”. The first-pass ranking is done using Okapi system’s basic scoring formula[1]. The documents are then rescored using the same formula with the top-ranked K documents as queries, weighted according to their first-pass scores. As we shall see in Sec. 5 below, the formula is motivated by viewing the rescoring process as a Markov process. This approach improves the precision substantially outside the topK retrieved documents.

We have tested a variety of other techniques in trying to improving the system. These include word-sense disambiguation, passage retrieval, and document length suppression. Although they do not yield substantial or consistent improvements, some insights into search techniques can nevertheless be extracted.

Our experiments are done using the short version of the ad-hoc TREC-5 queries with just the description field retained. The offical entry is submitted

	disk2
ap	HEAD, TEXT
fr	TEXT
wsj	HL, LP, TEXT
ziff	ABSTRACT, SUMMARY, TEXT
	disk4
cr efiles	CENTER, H2, SO, TEXT, TI, TTL, UL
cr hfiles	H2, SO, TEXT, TI, TTL, UL
fr94	ACTION, ADDRESS, AGENCY, DOCTITLE, FOOTNOTE, FURTHER, SIGNJOB, SUPPLEM, TABLE, TEXT, USBUREAU, USDEPT
ft	CN, CO, HEADLINE, IN, PE, TEXT , TP

Table 1: Fields retained in TREC-5 documents

as *ibms96a*. For comparison purposes, performance on TREC-4 data and other smaller corpora are also reported here.

2 Preprocessing, Morphological analysis and Sensing

As mentioned in the Introduction, we utilize only the description field of the queries, since we anticipate users of our system will not enter more than one sentence. For the candidate documents, we discard a number of fields which contain noisy or useless content. A list of the retained fields can be found in Table 1.

The words in both the queries and the documents are stemmed using a morphological analyzer such that only the root of a word remains. The morphological analyzer also tokenizes the text such that punctuations are stripped, and common bigrams are joined together as single tokens (e.g. "because of" becomes "because_of"). This analyzer is built upon a part-of-speech tagger [3]. Once the words are tagged, a table lookup enables one to extract the root of any morphs.

We have previously studied the efficacy of the morphological analyzer and

	AveP	P5	P10	P30	P100
words	0.2488	0.4800	0.4460	0.3513	0.2274
cased morphs	0.3319	0.5920	0.5340	0.4127	0.2766
uncased morphs	0.3507	0.5920	0.5540	0.4373	0.2910
manual cased morphs	0.3560	0.6200	0.5540	0.4367	0.2914

Table 2: Effects of morphological analysis and true-casing. First-pass (Okapi with unigrams) results on WSJ portion of TREC-3

	AveP	P5	P10	P30	P100
words	0.5800	0.3040	0.1960	0.0760	0.0264
cased morphs	0.8083	0.3520	0.2240	0.0867	0.0264

Table 3: Similar to Table 2: Results on radio broadcasts.

its true-casing facility on the WSJ[1] portion of TREC-3 data (see Table 2), and also on a small database of 140 radio broadcasts stories[2] (see Table 3). We use the Okapi ranking formula for this (see Section 4.) While we found that the analyzer does help retrieval accuracy, true-casing does not. This is because the true-caser does not have sufficient accuracy. The capitalization of some non-proper nouns in the TREC-3 queries are particularly problematic. We have manually removed these capitalization in the queries, and re-run the ranking program. In this case, cased morphs do perform slightly better than non-cased morphs (see last row in Table 2). In producing the official TREC-5 results, we use only uncased morphs.

We have also studied the effects of word-sense disambiguation[4] on the retrieval accuracy, although we have not incorporated it into the official system. Scores resulting from sensed queries[3] are combined linearly with scores from unsensed queries (which include both unigram and bigram features, as explained in Sec.3) in a 9 to 1 ratio. The effects on TREC-4 results are quite obvious: even though the average precision with sensed words is lower, the precision at top10 is higher.(See Table 4.) This is an effect that we will observe again and again: the more specific a query, the higher the precision at

[1]The queries in this case are the TREC-3 ad-hoc queries.
[2]We made up some 80 short queries with about 6 words each.
[3]Note that only a selected number of words in each query are sensed.

the top few scores, at the expense of a lower precision at the tail end of the ranked list. The opposite effects happen when we expand a query. We will elaborate on this point in Section 9. We note that the effects of sensing on TREC-5 is less encouraging. Top5 precision did not improve although top10 does. (See Table 5.) This may be because far fewer words in the TREC-5 queries were sensed. We will examine this in detail in [4].

Before extracting the term frequency information from the queries and documents, we filter them through a standard stopword list of 940 words.

3 Bigrams extraction

We extract all pairs of words from the text that are within a running 4-word window and form bigrams. (Stopwords do not count in this window.) To economize on storage space, however, only bigrams with both words in the queries vocabulary are retained. The bigrams are not ordered: i.e. (w_1, w_2) is the same as (w_2, w_1).

4 First-Pass Ranking

For first-pass ranking, we adopted the basic Okapi [1] formula from TREC-3. Using their notation, each (non-stop) word in the query and document contributes a weight of

$$w = \frac{tf}{0.5 + 1.5 \times \frac{dl}{avdl} + tf} \times w^{(1)} \times qtf, \tag{1}$$

where

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

is the usual inverse document frequency (*idf*) factor. We can use the same scoring formula for the bigrams also, with two minor qualifications: (1) $w^{(1)}$ is set to 1.0 and (2) dl and $avdl$ refer to the document lengths in bytes of the original text, not the number of bigrams in that text. After the official evaluation, we have tried an experiment where the bigrams have true $w^{(1)}$ obtained from the corpus. We found indeed that there is a small benefit in using idf's in bigrams scoring.

The unigram and bigram scores from Eq.(1) are combined linearly such that the unigram scores have a weight of 0.6 versus 0.4 for the bigrams. The result of these experiments are reported in Table 4 and Table 5. It is clear that the greatest benefit of adding the bigrams is in improving the precision at top5 retrieved documents.

5 Knn Re-Scoring

In order to expand the original query, we select the topK ($K = 10$ in our runs) documents from the first pass and use these as queries to re-rank the topN documents ($N = 1000$). Let $P(d_k|q)$ be the first-pass Okapi score of the k-th document, and $P(d_i|d_k)$ be the Okapi score of the i-th document using the k-th document as query. Then we define the second-pass score of document i to be

$$s_i = \sum_{k=1}^{K} P(d_i|d_k)P(d_k|q). \tag{2}$$

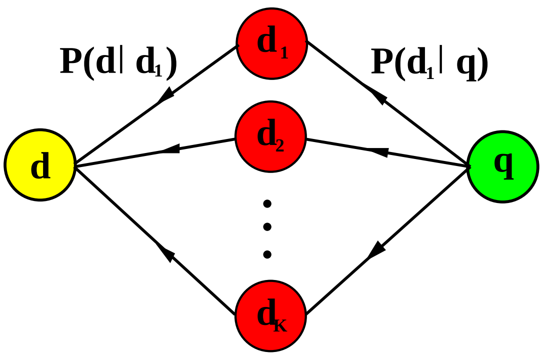

Figure 1: Knn Re-scoring

A graphical representation of this is shown in Figure 1. The notation is designed so that the formula resembles that of a Markov model, which was the original motivation for it.

We can just use the words in the expanded query k to compute s_{ik}, or we can include the bigrams in k. As mentioned above, the bigrams are restricted to those with words in the original (first-pass) query vocabulary.

Again, the unigram and bigram scores from Eq.(2) are combined linearly such that the unigram scores have a weight of 0.05 versus 0.95 for the bigrams. The addition of bigrams improves the second-pass scores slightly. (See Table 5.)

The overall improvement due to Knn re-scoring has been modest. In fact, there is a decrease in precision at the top10 documents. Our previous experience with Knn re-scoring in TREC-4 and other corpra has been much more positive. In Sec. 9, we shall discuss a possible cause of this.

6 Passage Retrieval

One may hypothesize that some documents may be viewed as relevant to a topic simply because of one particular passage within it, while other passages discuss irrelevant matters. If this view is correct, segmenting the documents into passages and scoring each of them separately, taking the maximum passage score as representative of the whole document may improve retrieval performance.

We have performed this experiment on TREC-4 data by segmenting documents into overlapping passages of about 100 words. The overlap is about 50 words. The result of using the first-pass "best passage" score is substantially worse than taking the document as a whole (with aveP=0.1803 vs. 0.1980 without passages), as can be seen from Table 4, seemingly disprove the hypothesis.

We have also tried using passage retrieval on the second pass. This can be done in 2 ways: using passages in the expanded queries to re-score the documents, and/or scoring the best passage in the document. We have tried various combinations of these 2 ways on the TREC-4 data. Furthermore, we can have a weighted average of first-pass passages used as expanded queries using the following weighting formula:

$$\begin{aligned} w_b &= 1/n^{\xi} \\ w_o &= \tfrac{1-1/n^{\xi}}{n-1}, \end{aligned} \qquad (3)$$

where w_b is the weight of the <u>b</u>est first-pass passage, and w_o is the weight of the other passages, n is the number of non-overlapping passages[4], and ξ ($0 \leq \xi \leq 1$) is the parameter that let us control the weights distribution. If $\xi = 0$, we use only the best passage as expanded query; if $\xi = 1$, all passages contribute equally, and we are back to using the whole first-pass document as expanded query. We again find that the best combination is obtained when we score the whole document (not its passages). In the future, one can combine the best-passage score and the whole-document score to see if this is better than each alone. Whether we use weighted passages as queries does not seem to have much effect.

Because of this discouraging result, we did not attempt passage retrieval in TREC-5.

7 Length Suppression

We believe that long documents have an unfair advantage over short ones because they often have a variety of heterogeneous subtopics which increases the chance of matching the query words. However, the occurrence of these subtopics does not necessarily make the document relevant to the query. Hence we have applied a sigmoidal suppression factor for document length, such that the new 2nd pass score s_{new} is related to the old 2nd pass score s_{old} by

$$s_{new} = \frac{1}{1 + e^{(dl - meanDl)/width}} s_{old},$$

where $meanDl = 10000$ and $width = 8000$. Scores of documents with length much larger than $meanDl$ will be exponentially decreased.

It turns out that length suppression hurts precison for the top5 documents but is beneficial beyond that. In the future, we will adopt the technique advanced by the SMART group [5] to discover the bias (with respect to document length) that our system suffers.

[4]If we happen to pick a best passage that is offset by an odd-multiple of 50 words, we would simply ignore either the first 50 words or the last 50 words of the document for simplicity.

8 Combining 1st and 2nd pass Scores

Motivated by the idea of using a "prior probability" in a Bayesian probabilistic retrieval framework, we can combine the second-pass scores linearly with the first-pass scores, thus viewing the first-pass scores as the log of a prior probability. We use a (unnormalized) weight 0.01 for the first-pass vs. 0.99 for the second-pass scores. This generally produces a small improvement in both TREC-4 and TREC-5. The results are listed also in Table 4 and Table 5. The larger improvement in TREC-5 is expected because, as we discuss in the Conclusion, the initial retrieval there is so poor that using these documents for query expansion often hurts precision. Adding the first-pass scores ensures that movement in document ranks is reduced.

A more radical method to ensure that Knn re-scoring does not degrade the topK precision is to fix the rankings of the top10 or so documents from first pass and simply re-score the rest in the second pass. The results of not re-scoring the top10 in second pass are shown in Table 4 and Table 5. In TREC-5, it does show a substantial improvement over the official second pass score; however, this result does not hold for TREC-4. Hence it is still possible that Knn re-scoring not only improves the aveP, but can improve the topK precision as well.

9 Conclusion

We have implemented a query expansion scheme using all the words and some of the bigrams in the topK documents. An improvement in average precision is achieved by this scheme. When we tested this scheme on TREC-4 topics, the percentage improvement in aveP is more than 9 times bigger than the TREC-5 improvement (see Table[4]). The unimpressive performance in TREC-5 may be due to the fact that the initial retrieval has so poor accuracy that 2/3 of the top10 documents used as expanded queries are irrelevant. Nevertheless, we achieved an above-median average precision in 34 out of 50 topics compared with other TREC-5 (short queries) participants.

Compared to other methods of query expansion, such as that used by the SMART TREC-4 engine [2], we have used far more words and bigrams for expansion. Apparently this has hurt our performance, perhaps by overly generalizing the very short queries. In particular, the topK precision in

1st pass	2nd pass	AveP	P5	P10	P30	P100
u	–	0.1980	0.5240	0.4660	0.3447	0.2314
u, b	–	0.2014	0.5160	0.4440	0.3487	0.2356
u, b-idf	–	–	–	–	–	–
u, b, s	–	0.1987	0.5320	0.4740	0.3540	0.2310
u, b	u	0.2150	0.5120	0.4800	0.3960	0.2664
u, b	b	0.1629	0.4000	0.3600	0.2800	0.1938
u, b	b-idf	0.1678	0.4280	0.3700	0.2820	0.1992
u, b	u, b	0.2362	0.5040	0.4920	0.3980	0.2774
u, b	u, b, c	0.2378	0.5080	0.4880	0.3980	0.2806
u, b	u, b, l	0.2369	0.5080	0.4940	0.4013	0.2758
u, b	u, l, f	0.2315	0.5160	0.4440	0.4007	0.2734
u, b	u, b, l, f	0.2409	0.5160	0.4440	0.3980	0.2768
u, p	–	0.1803	0.4560	0.4040	0.3293	0.2194
u, b	u, pq(1), p	0.1523	0.3640	0.3480	0.2760	0.1912
u, b	u, pq(0.3), p	0.1542	0.3800	0.3600	0.2813	0.1956
u, b	u, pq(0), p	0.1542	0.3920	0.3680	0.2873	0.1968
u, b	u, pq(0.3)	0.2197	0.5000	0.4760	0.3933	0.2660
u, b	u, pq(0)	0.2198	0.5040	0.4780	0.3933	0.2662

u= unigrams included.
b= bigrams included.
s= sensed unigrams used.
b-idf = use idf in Okapi scoring of bigrams.
c= combining 1st pass and 2nd pass scores in a 99 to 1 ratio.
l= length suppression used.
f= fixing top10 first pass rankings.
p= use best passage score as representative.
pq(ξ)= use weighted passages as queries with exponent ξ.

Table 4: Results of experiments on TREC-4.

1st pass	2nd pass	AveP	P5	P10	P30	P100
u	–	0.1466	0.3640	0.3160	0.2340	0.1528
u, b	–	0.1557	0.4120	0.3320	0.2373	0.1560
u, b, s	–	0.1522	0.4120	0.3480	0.2280	0.1518
u, b	u	0.1563	0.3520	0.3100	0.2420	0.1708
u, b	b	0.1139	0.2680	0.2240	0.1733	0.1214
u, b	b-idf	0.1164	0.2960	0.2200	0.1807	0.1266
u, b	u, b	0.1587	0.3840	0.3180	0.2453	0.1718
u, b	u, b, c	0.1643	0.3600	0.3300	0.2593	0.1762
*u, b	u, b, l	0.1585	0.3440	0.3240	0.2600	0.1726
u, b	u, l, f	0.1719	0.4280	0.3320	0.2627	0.1772
u, b	u, b, l, f	0.1685	0.4280	0.3320	0.2573	0.1736

* official TREC-5 result.
Otherwise identical to Table[4].

Table 5: Results of experiments on TREC-5.

TREC-5 was degraded by Knn re-scoring, even though it helps the aveP there. In the future, a far more restrictive selection of terms may be tried. In addition, combining first- and second-pass scores as we did in Section 8 also helps to alleviate this over-expansion problem.

Here is a summary of other results we obtain:

- Morphological analysis produces significant improvements over tokenized words. Sensing produces slight improvement in topK precision, but not aveP.
- Addition of bigrams produces overall improvement in aveP, but surprisingly, it lowers topK precision in TREC-4. Using idf for bigrams produces additional improvement.
- Using best passage scores by themselves degrade performance. Using best passages as expanded queries do not have much effect.
- Length suppression has positive effect on TREC-4, but negative effect on TREC-5.

In the future, we hope to experiment with many of the possible improvements suggested above.

10 Acknowledgements

This work is supported by NIST grant no. 70NANB5H1174. We thank Satya Dharanipragada for assistance in the pre-processing of the corpus, and Robert T. Ward for stimulating discussions and various aspects of computing.

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] C. Buckley, G. Salton, J. Allan, A. Singhal 1995. Automatic query expansion using SMART: TREC 3. In D. K. Harman, editor, *Proceedings of the Third Text REtrieval Conference (TREC-3).* NIST Special Publication 500-225, 1995.

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

[4] E. P. Chan, S. Garcia, S. Roukos, T. Ward. (To be submitted to ACL '97) Bilingual Sensing applied to Information Retrieval

[5] A. Singhal, G. Salton, M. Mitra, C. Buckley 1995 Document length normalization. Technical Report TR95-1529, Cornell University