National Taiwan University at Terabyte Track of TREC 2005

Ming-Hung Hsu and Hsin-Hsi Chen

Department of Computer Science and Information Engineering

National Taiwan University

Taipei, Taiwan

E-mail: mhhsu@nlg.csie.ntu.edu.tw; hhchen@csie.ntu.edu.tw

Abstract

There are three tasks in the Terabyte track of TREC 2005, i.e. Efficiency, Ad hoc and Named page finding. We participated in all the tasks and use different retrieval methods to deal with each task, aiming to vary the retrieval method according to the different characteristics of different tasks. In Ah hoc task, we adopt the technique of query-specific clustering. In Named page finding task, we cared more about the information of document title and anchor text of out-links.

1 Introduction

This is the first year we participated in Terabyte Track. The primary goal of this track is to develop an evaluation methodology for terabyte-scale document collections. Besides, efficiency and scalability issues are also concerned. Some different criteria are used in evaluation because the information needs of the web users often vary a lot. There are three tasks in this track, i.e., Efficiency, Ad hoc, and Named page finding. Our retrieval methods for each task are mainly based on Okapi [6], with some variants according to the characteristics of different tasks. In this paper, we focus on Ad hoc task and Named page finding task since we simply used Okapi to retrieve documents based on content words in Efficiency task.

In Ad hoc task, we employed clustering technique to improve retrieval performance. A scoring function was adopted to rank clusters that were generated from the top *N* documents retrieved by Okapi, i.e., query-specific clustering [1].

In Named page finding task, the information from the document title and anchor text is very useful and important to identify the named pages, so that we increased the weight of document title and anchor text when computing the relevance score of a document to a query.

2 Preprocessing and Indexing

All the documents in the corpus were stemmed using Porter's algorithm [3], and all words except stop words were indexed. Titles (i.e., the words within <title> and </title> in html format) of documents are extracted and indexed additionally, so as anchor text of out-links of documents. The two additional indices will be used in Named page finding task.

3 Ad Hoc Task

3.1 Motivation and Description of Our Method

In a typical ad hoc retrieval task, the IR system is requested to retrieve as many relevant documents to some topics (queries) as possible. For most topics in Ad Hoc tasks of TREC, there are usually tens of relevant documents. For example, topics of TREC9 (451-500) have 2,617 relevant documents, about 52 relevant documents per topic. Intuitively, as the number of documents increases, the number of relevant documents increases, too. Under this situation, we tried to cluster relevant documents together in the ranked list retrieved by Okapi (so called query-specific clustering), to improve retrieval performance. Clustering hypothesis [2] has been verified to be held in the manner of query-specific clustering [1].

Our method is a two-stage approach. In the first stage, we used Okapi to retrieve 10,000 documents. In the second stage, the top N ($N \leqq 10{,}000$) documents retrieved in the first stage were clustered. After that, those clusters were scored and ranked by a heuristic function. In the final ranked list, all documents in a higher-scored cluster would be ranked higher than those in a lower-scored cluster, and the intra-cluster ranking of documents would follow their ranks in the first stage.

For the reason of efficiency, we used Bi-Section K-Means, which has shown to be an efficient and high-quality clustering algorithm, for query-specific clustering [4].

3.2 Ranking Clusters

In the ranked list returned by an IR system based on probability model, a document with a higher rank is more possible to be relevant to the query. According to this property, we proposed a simple cluster scoring function. This function, which is in terms of the rank of a document and the *general performance* of the IR system, estimates the probability of the document to be relevant, and further the quality of certain cluster. The general performance of Okapi here is the $P@R$ of testing the topics used in last year's terabyte track, where R is one of {5, 10, 15, 20, 30, 100, 200, 500, 1000}. The results are listed in Table 1. Different values of R represent the upper bounds of *rank levels* of document rank, so we partitioned documents into nine rank levels. Assume the function $L(r)$ maps a document with rank r into certain rank level. That is, $L(1)=1$, $L(6)=2$, $L(11)=3$, and so on. According to Table 1, we can estimate the relevant probabilities of documents in different rank levels. For example, the relevant probability of a document with rank 16, i.e., the 4th rank level, can be estimated by

$$(20 * P@20 - 15 * P@15) \,/\, (20-15) = 0.4202$$

In this way, we can derive Table 2 from Table 1. It shows the relevant probability $RP(L)$ related to rank level L.

Let C_i denotes the ith cluster, R_{ij} denotes the rank of the jth document in C_i and $|C_i|$ denotes total number of documents in C_i. The cluster scoring function $S(C_i)$ determines the average relevant probability of documents in C_i:

$$S(C_i) = \frac{1}{|C_i|}\sum_{j=1}^{|C_i|} RP(L(R_{ij})) \qquad (1)$$

After scoring clusters, all the clusters were ranked according to their scores, and the final ranked list is generated.

Table 1. The performance of Okapi on TREC2004 topics

R	5	10	15	20	30	100	200	500	1000
P@R	0.4612	0.4633	0.4694	0.4571	0.4469	0.3720	0.3073	0.1934	0.1235

Table 2. Relevant probability related to rank level

L	1	2	3	4	5	6	7	8	9
RP(*L*)	0.4612	0.4654	0.4816	0.4202	0.4265	0.3399	0.2426	0.1175	0.0536

Besides Table 1 and Table 2, we also used the topics of last year to determine the number of clusters, K, which is a parameter for query-specific clustering using Bi-Section K-Means algorithm. In the first stage of our method, 10,000 documents were retrieved and regarded as the baseline (without re-ranking), but only the top N will be clustered and re-ranked. By testing our method with the topics of last years, *K*=10 and *N*=1,000 showed a better result. However, no matter what values of *K* and *N* were used, our simple cluster scoring function didn't outperform the baseline in average precision (AP). On the other hand, interestingly, our method exhibited potential for the improvement in *P@10*. As most users browse only a few of the top-ranked documents when they search the web, the improvement in *P@10* may be useful and meaningful.

3.3 Experimental Results and Discussion

Table 3 shows the results of three runs we submitted in this task. There are three evaluation criteria, i.e., average precision, P@10, and Binary preference (Bpref). NTUAH1 is the run using BM25 on full documents. The run NTUAH2 used BM250 to retrieve passages, which are dynamically determined, so it is time-consuming. NTUAH3 is the result of our method. It is obvious that our simple two-stage approach is inferior to the baseline, NTUAH1, and passage retrieval a little outperforms full document retrieval, whatever the evaluation criterion is. There are several reasons for the out-of-expected performance of our two-stage method. The first is that our simple cluster scoring function deeply depends on the performance of the first stage, i.e., the original ranked list retrieved by Okapi. When the performance of the first stage is not good enough, our cluster scoring function usually performs worse. The second reason is that the number of truly relevant documents has effects on ranking clusters. In other words, if there are fewer relevant documents in the corpus, it is more difficult to rank clusters. For the 50 topics in this task, the median of the numbers of relevant documents is 172. Comparing with NTUAH1, our method performs better in 26 topics. In the remaining 24 topics, 17 of them have relevant documents fewer than 172. This result reflects the second reason mentioned above. The third reason is the relevant probability (refers to Table 2) estimated by last year's topics is much different from the result of this year's. Table 4 shows the performance of Okapi on this year's topics. In Table 4, it's presented that *P@R* decreases whenever *R* increases. However, it's not the same condition in Table 2. In Table 2, *P@R* does not decrease obviously between *R*=5 and *R*=30. The difference between Table 2 and Table 4 directly influences the result of our cluster scoring function.

Table 3. Our Results of Ad Hoc task

Run-ID	AP	P@10	Bpref
NTUAH1 (Okapi-Doc)	0.3023	0.59	0.3201
NTUAH2 (Okapi-Psg)	0.3233	0.6	0.3419
NTUAH3(Okapi-D+Clst)	0.2425	0.506	0.29

Table 4. The performance of Okapi on TREC2005 topics

R	5	10	15	20	30	100	200	500	1000
P@R	0.6520	0.5900	0.5533	0.5400	0.5180	0.4282	0.3459	0.2174	0.1383

4 Named Page Finding Task

4.1 Motivation and Description of Our Method

Named page finding task is much different from traditional ad hoc task in the aspect of the number of "correct answers". The goal of Named page finding is to find a specific page or its "near duplicates" with near rank one, so using only content of the document to identify the relevance between the document and the query is obviously not sufficient for this task. In the environment with terabyte-scale corpus and without support of extremely expensive hardware, we tried to utilize titles and anchor texts of out-links of documents to improve the search result which is retrieved based on only document contents, since document title and anchor text are commonly considered as informative [5]. Besides the original index of document contents, two additional indices are produced for the titles and the anchor text of out-links, respectively. Our method is described as follows:

For each query, we perform BM25 retrieval on all the three indices (i.e., content, title, and anchor text) and merge the three ranked lists. The results are merged by a linear combination of scores (which have been normalized) of the documents, hence the relevant score value (RSV) of document Di is

$$RSV(D_i) = C_C \cdot S_C(D_i) + C_T \cdot S_T(D_i) + C_{AT} \cdot S_{AT}(D_i) \qquad (2)$$

where $S_C(D_i)$, $S_T(D_i)$ and $S_{AT}(D_i)$ represent the score of D_i in the list retrieved on index of content, title, and anchor text, respectively. C_C, C_T, and C_{AT} are their weights and $C_C + C_T + C_{AT} = 1$.

4.2 Experimental Results and Discussion

Table 5 shows our results of three runs. NTUNF1 is the baseline, i.e., document retrieval using BM25. NTUNF3 is the result of passage retrieval using BM250. NTUNF2 is our method that utilizes the information of document title and anchor text of out-links. Passage retrieval did not clearly outperform document retrieval. That indicates the information of document content was insufficient to deal with Named page finding. Our method got slightly improvement in the percentage of named pages retrieved at top 10, but results of the three runs exhibited no significant difference in MRR. However, it didn't mean that document titles and anchor text had no influences on the performance. By comparing the results of individual topics in NTUNF1 and in NTUNF2, we found that many topics got much different performance in the two runs. For example, the named page of topic 619 was ranked at 2 in NTUNF1 but was ranked at 21 in NTUNF2. This example indicated that document title and anchor text were intuitively informative but also very noisy, especially for this task. How to utilize the information reliably and robustly is necessary for future work.

Table 5. Our results of Named page finding task

Run-ID	ARR	%top10	%fail
NTUNF1 (Doc)	0.385	52	20.2
NTUNF2 (D＋T＋AT)	0.387	51.6	20.2
NTUNF3 (Psg)	0.388	51.2	19.4

Figure 1 shows the performance difference of RR (reciprocal of rank) between NTUNF1 and NTUNF2 for each topic. The *x*-axis stands for topic ID and the *y*-axis stands for the performance difference, i.e. for the performance of each topic, its RR value in NTUNF1 subtracting that in NTUNF2. For the points in Figure 1, if the *y*-coordinate is smaller than 0, it means our method performs better than baseline for that topic, otherwise it means that our method performs worse for that topic. It's showed that our method improves the baseline for many topics but there are also some topics got worse result at the same time.

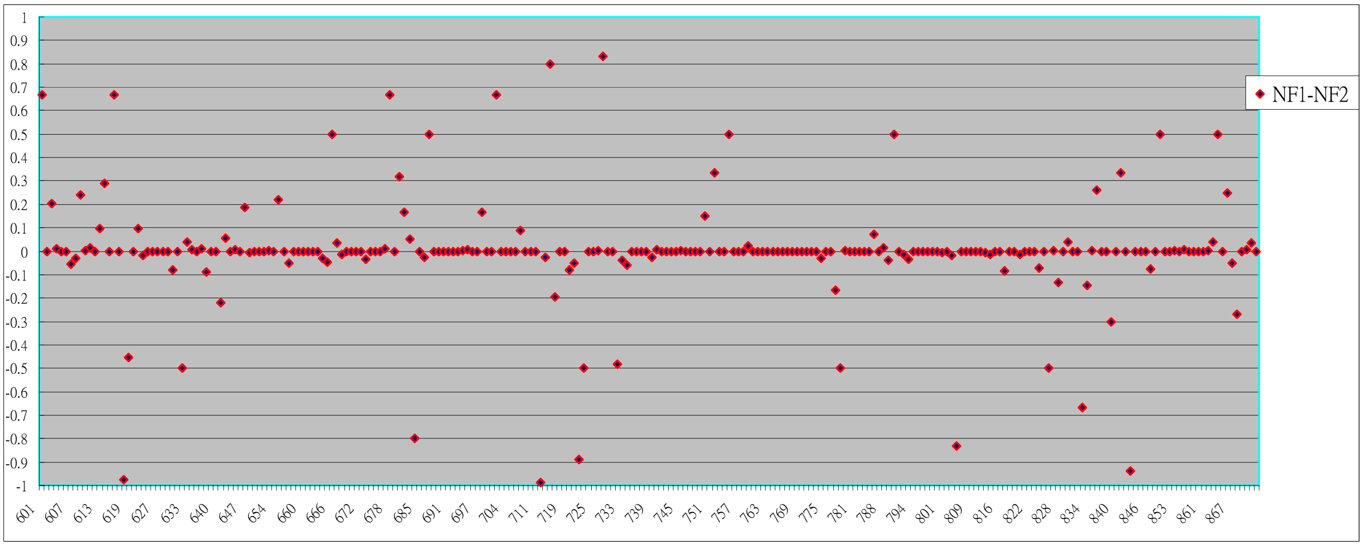

Figure 1. The difference of RR values between NTUNF1 and NTUNF2

References

[1] A. Tombros, R. Villa and C. J. Van Rijsbergen (2002). The effectiveness of query-specific hierarchic clustering in information retrieval. *Information Processing and Management*, 38, pp. 559-582.

[2] E. M. Voorhees (1985). The cluster hypothesis revisited. In *SIGIR* 1985, pp.188-196

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

[4] M. Steinbach, G. Karypis, and V. Kumar (2000). A comparison of document clustering techniques. In *KDD Workshop on Text Mining*, 2000.

[5] P. Bailey, N. Craswell and D. Hawking (2001). Engineering a multi-purpose test collection for Web Retrieval experiments. *Information Processing & Management*, 2001.

[6] S. E. Robertson et al (1995). Okapi at TREC-3. In *Overview of the Third Text REtrieval Conference* (*TREC-3*). Edited by D. K. Harman, Gaithersburg.