THU TREC2002 Web Track Experiments*

Min Zhang, Ruihua Song, Chuan Lin, Shaoping Ma, Zhe Jiang, Yijiang Jin, Yiqun Liu, Le Zhao
State Key Lab of Intelligent Tech. & Sys., CST Dept, Tsinghua University, Beijing 100084, China
zhangmin99@mails.tsinghua.edu.cn

1 Introduction

Anchor text has been proofed efficient in former TREC experiments on homepage finding task[1] and somewhat useful to ad hoc retrieval by result combination[2]. In this year, our conclusion was consistent with formers. Besides, the use of the URL and links inside the webpage were also observed. Again, results on training set are encouraging.

We made an assumption that a key resource is more likely to link to multiple relevant documents. Then the out-degree of the page and the similarities of the documents the page point to were used as the two factors for key resource selection. Experimental results were quite good, showing their ability of finding key resource on one server.

Two site uniting (SU) approaches have been studied to select proper pages as the representation of one server. (1) The document which has index characteristic and has a high enough similarity is reserved as key resource. (2) Documents of the same server in result list are given different reliability factor which is decaying by decreases of similarities. Both are useful for given examples (using as training set) in this year's Web track, especially the latter one. Better results were got by combing SU approach and out-degree factor mentioned above to find key resource.

All the experiments we performed were run on Okapi system. There are quite a few parameters to tune, which affect the performance greatly. Therefore, we also proposed and implemented a genetic algorithm based dynamic parameter learning approach to all the tasks.

2 Data preprocessing

2.1 Word and document pruning

Odd characters are meaningless to users and may bring on exception in processing. We cleaned characters that are unprinted and unrelated with formats. Also, the words containing more than 20 characters were deleted as they were deemed to incorrect words that affect collection statistics.

As we discovered that two bunches of documents had no content, we pruned them. One bunch was the set of files with postfix of “jpg” or “gif” (totally 151 files). The other bunch was most redirect html documents (17,086 files) as they only acted as gangways to destined documents that contained detailed content. The exceptions were those documents that redirected to themselves in order to refresh periodically. In all, we removed 17,235 documents (note there’s documents overlap between two bunches). No relevant documents were lost.

2.2 Html parsing

As non-html pages (doc, ps and pdf) were much longer than html pages and had no html tags, we divided the collection into two, non-html (153,775 files) and html (1,076,743 files), and indexed

* Supported by the Chinese National Key Foundation Research & Development Plan (Grant G1998030509), Natural Science Foundation No.60223004, and National 863 High Technology Project No. 2001AA114082.

these two subsets respectively.

Html pages were parsed for two goals. One is to convert the file to XML format. For example, text in the tags of <b>, and <u> was extracted and marked with a new XML tag <srhB>. It was preparation for using html structure to improve retrieval. The other goal is to remove invisible text, such as comment text and codes in scripts, because they are meaningless to users.

3 Using of document structure

HTML document structure is studied in our experiments. We found that using keywords, bold text and title fields of the in-link pages do help on named page finding task. We use these three parts of the in-link webpage and in-link anchor text to build a new document of current page, and then index and retrieve on this new dataset. Although result of the new dataset is not good, by combing this result and result on original dataset, we got some improvement, shown in Table 3.1.

Table 3.1 Effects of using document structure on named page finding task

Method	Content based retrieval	Anchor + special fields	Combined
MRR	0.690	0.530	0.717

To topic distillation task, fields of bold font (<B>) and keywords in <meta> are also useful to the retrieval. It does help by giving a different term weight from the other full text.

4 Using link structure

4.1 Re-Ranking based on link analysis

Intuitively, counting the links to a document has been used to estimate the document's quality. However, the concept of key resource is different from the concept of quality. Therefore, we used some other features, such as Kleinberg's hub score, Kleinberg's authority score + hub score and out degree, to estimate whether the document is a key resource. The experiment on the training examples showed some improvement (see Table 4.1), but the result was disappointing to 50 topics of web track. The training set included seven topics in track guidelines.

Table 4.1 Finding key resources with link analysis on training topics

	Baseline	By linke analysis based re-ranking
Average precision of top 20 results	0.3827	0.4395

4.2 Site Uniting

The definition of key resource implied that it was most possible that only one page was key resource among pages from an identical site. As the list of content-based retrieval contains all the relevant pages, some might be ranked adjacently. Even worse, all of them were ranked high and thus pressed a possible key resource from another site lower. Therefore, we re-ranked the list by enhancing the page with highest rank from each site. The approach is called *Site Uniting* (SU). The algorithm can be shown as following, where F_1=1.03, F_2=1.01, F_3=1.005 in our experiments:

1) Divide the list to sub-lists, all pages in one sub-list come from an identical site.
2) To one sub-list, give the first, second and third highest similarity the weight F_1, F_2 and F_3, respectively.
3) Merge all the sub-lists into one and re-rankit.
The additional condition is that $F_1 > F_2 > F_3$.

4.1

The p@10 was lower than the base although 11-point precision was a little higher.

5 Using URL

In topic distillation task, the URL is also used to the retrieval. There are two functions can be provided by URL: (1) searching and shrinking; (2) scoring and selecting. On searching and shrinking, we give a "right level" to the return results within a server. The shrinking is based on three principles: i. Pages with more keywords matched in the URL is more important. ii. The location of the match effects the importance of the page, the righter the better; iii. To pages with the same conditions i and ii, shorter URL is better. On scoring and selecting, a keyword search is performed on URLs and got a result list which is useful to re-rank content-based retrieval result.

On named page finding tasks, we tried the URL classification. Using URL types (TNO-UTwente TREC 10 report [3]) proved to be a success in Entry Page Finding task last year. Unfortunately, it doesn't help this year. We analyzed 100 of the 150 correct answers. Table 5.1 shows how Named Pages distribute over the 4 URL types (root, sub root, path and file). Compared with Table 5.2, we conclude that Named Pages have almost the same distribution over this kind of URL classification as ordinary web pages. That means URL type is not a useful character for this year's task.

Table 5.1 URL type distribution in qrels

URL type	#page	percent
Root	2	2%
Sub root	1	1%
Path	6	6%
file	91	91%

Table 5.2 URL type distribution in corpus

URL type	#page	percent
Root	11680	0.6%
Subroot	37959	2.2%
Path	83734	4.9%
file	1557719	92.1%

6 Combination of distributed Retrieval results

As described in former section, the corpus has been divided into two data sets: one is for html document called html database, another one is for the remaining, call extra database. Retrieval has been done separately on these two distributed databases. How to combine the two result list is one of the interesting issues. The algorithm we used is to find a start rank point in html results, and insert the extra results from this point with some interval. The selection formula of start-point is:

$$Start = (2 - Sim\_extra/Sim\_html) * k + b \qquad 6.1$$

where k and b are constants. In our experiments, k=150 b=14.

7 Unsupervised dynamic parameters learning

The similarity between queries and documents is computed by BM2500[4]. There are quite a few parameters to be set, such as b, k_1, k_3, $avdl$. Especially b and k_1 play an important role in the performance. Parameters by training data, however, are always not suitable and helpful when dataset or queries change. At the same time, relevance judgments are not available while retrieving, thus supervised learning algorithms do not help. In this section, an unsupervised dynamic parameters (b and k_1) learning algorithm is described.

We use Genetic Algorithm (GA) for learning process. The fitness function in GA determines whether each set of parameters is good or not and the survival probability of each set[5]. According

to the fact of lacking relevance judgment, it is required to find appropriate fitness functions which are oriented from the data and the retrieval themselves. The one we used is the summation of the similarity scores of top n relevant documents, shown in Eq(7.1).

$$fit\_fun_1 = \sum_{i=1}^{50}\sum_{j=1}^{1000} sim_{i,j} \qquad 7.1$$

Figure 7.1 and 7.2 show the correlation of using summation of similarities and the 11-point average precision, and the correlation of P@10 and summation of similarities on TREC10 and TREC2002 dataset, respectively.

Figure 7.1 Correlation between the two fitness functions in TREC2001 data

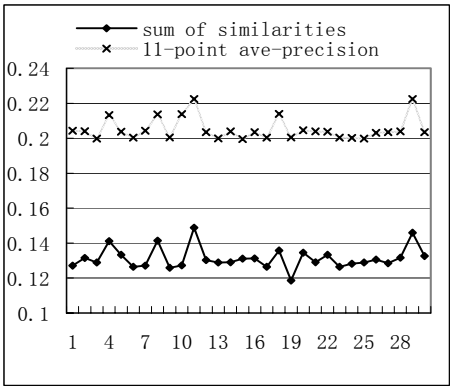
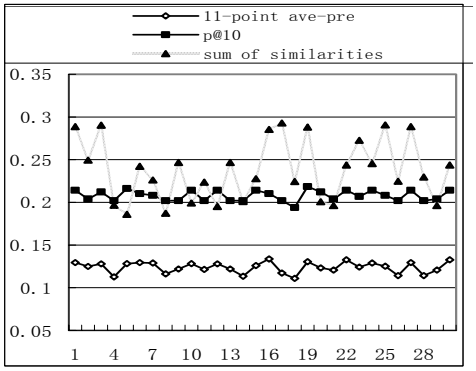


Figure 7.2 Correlation between three fitness functions in TREC2002 data



8 Runs Submitted and Evaluation Result

Table 8.1 Runs submitted on Topic distillation task

Run	Description	P@10
Thutd1	Thutd4 + outdegree	19.80%
Thutd2	Thutd4 + anchor re-ranking	22.45%
Thutd3	Thutd4 + site uniting	23.06%
Thutd4	Thutd5 + anchor combination + extra db result	21.43%
Thutd5	On Html db, long query	25.10%

Table 8.2 Runs submitted on Named page finding task

Run	Description	MRR
Thunp1	Content method	0.690
Thunp2	Combining inverse rank of Content and special fields results	0.530
Thunp3	Thunp5 + URL hierarchy	0.719
Thunp4	Thunp1 + URL hierarchy	0.687
Thunp5	Re-ranking of content and special fields results	0.717

Reference

[1] Craswell, N., etc., Effective site finding using link anchor information. In *SIGIR-01*.

[2] Gaojianfeng, etc., TREC-10 Web Track Experiments at MSRCN, in TREC-10.

[3] Thijs Westerveld, etc., Retrieving web pages using content, links, URLs and anchors, in TREC-10.

[4] Robertson, S. E., and Walker, S., Okapi/Keenbow at TREC-8. In *TREC-8*.

[5] Y. S. Chen, C. Shahabi, Automatically Improving the Accuracy of User Profiles with Genetic Algorithm, International Conference on Artificial Intelligence and Soft Computing, 2001.