TREC 2004 Web Track Experiments at CAS-ICT

Zhaotao Zhou[1,2], Yan Guo[1], Bin Wang[1], Xueqi Cheng[1], Hongbo Xu[1], Gang Zhang[1]

(1.Institute of Computing Technology,Chinese Academy of Sciences,Beijing, China, 100080

2.Graduate School of the Chinese Academy of Science , Beijing, 100039)

http://lcc.software.ict.ac.cn/

zzt@software.ict.ac.cn, guoy@ict.ac.cn

Abstract: This report presents CAS-ICT's experiments on the Mixed query task of the TREC2004 Web track. Our work focused on combining different Web page evidences together to improve the overall retrieval performance. Four kinds of evidences, including body content(C), anchor texts (AT), basic structural information (S0) and extended structural information (S1) were considered for retrieval. Six combination functions were investigated in our experiments. The experimental results show that most functions can improve the retrieval performance. Some heuristic re-ranking techniques were also introduced and tested in the task. No query classification was made during the experiments.

Keywords: Web retrieval, TREC 2004, the Mixed query task, information fusion.

1. Introduction

This year we only participated in the Mixed query task of the Web track. From 2001 to 2003, different Web tasks, including homepage finding, topic distillation, named page finding, and known item finding, have been defined for the Web track. Each of them is a separate task with different type of topics. This year, all these tasks are combined together with only one set of mixed topics. "Mixed" here means topics for the above different tasks are blent together without knowing which one belongs to which task.

A natural idea is to classify the topics into different types and then to use different model or method for each type. Such kind of work on query classification has been done in [6]. We also implemented this idea in our work. However, the results are not as good as we expected. And this experiment is not included in this report. Our experiments mainly focused on Web evidences combination and re-ranking techniques.

As we know, a Web page has much more useful information than a plain text. For instance, besides the body text, the title, Meta data, some tags, anchor texts and links in a Web page can provide rich information to improve the performance of Web retrieval. All these information are called Web evidences. This year, we focused on combining different Web evidences together to improve the overall performance.

Another work in our experiments is to re-rank the retrieval results according to some heuristic methods.

Our work was still based on our enhanced version of SMART[1], which we developed for the previous Web tracks. Lnu-Ltu weighting scheme and pivoted document length normalization (PDLN) technique were used. However, the parameters of PDLN for different Web evidences were set to different values.

---

[1] ftp://ftp.cs.cornell.edu/pub/smart

The queries from TREC 2003's home/name page task and topic distillation task were combined together to a mixed query set (called MIXED03) for parameters estimation and training.

The rest of the report is organized as follows: Section 2 describes the concepts and formulas used in our experiments. Section 3 introduces our experiments and the results in detail. Finally, we give our conclusions in Section 4.

2. Concepts and formulas

2.1 Web evidences and their combination

As we pointed out before, Web pages have more information than plain texts. Besides body content texts, they have some other potentially useful information that can be used for Web retrieval. Such information include URL, title, anchor texts, tags, Meta data and structural information such as links. All these information are called Web evidences. In our experiments, a "pseudo-text" was constructed for each page. It consists of not only some information of current page, but also all the anchor texts that link to the current page and all the titles of the pages that the current page links to. The in-link information are called backward propagation information while the out-link information called forward propagation information. We guessed such a "pseudo-text" could provide richer information for Web retrieval. And our later experiments confirmed this thought.

A combination process of different evidences is as follows: given a query, for each page, first a score is computed according to each evidence, and then based on these scores, a combination score is computed as the unique final score. A combination function is the most important in the combination process.

Formally, a combination score of page p and query q is defined as:

$$CS(p,q) = \underset{i=1}{\overset{n}{com}}(s(e_i,q)) \tag{1}$$

Where ei is the ith evidence of total n evidences of page p, $s(ei,q)$ is the score based on ei and q, com is a function that combines the n scores. Some com functions are defined in table 1.

Table 1: Some Combination functions

Name	Meaning
CombMIN	Minimum of individual similarities
CombMAX	Maximum of individual similarities
CombSUM	Summation of individual similarities
CombANZ	CombSUM/number of nonzero similarities
CombMNZ	CombSUM* number of nonzero similarities

Fox& Shaw [2] have worked on the above methods for combining multiple retrieval runs and obtained improvements over any single retrieval run. Lee [7] analyzed why improvements can be achieved with evidence combination and showed that different runs retrieve similar sets of relevant documents while retrieve different sets of nonrelevant documents. Lee found that in his evaluation CombMNZ got the best retrieval performance. In our submitted run ICT04MNZ3, we used the CombMNZ to combine 3 retrieval results from anchor text, structure info and content. In our system and training set, CombSum performed better than CombMNZ, at the same time CombANZ and CombMIN had worse performance.

In addition, a linear combination method (LM) is also introduced in our work, which is actually a weighted summation of all individual evidences. The formula is:

$$CS(p,q) = \sum_i w_i * s(e_i, q) \tag{2}$$

Where w_i is the weight for evidence e_i. Each w_i can be learned from training set.

Table 2 defines the evidences that were used in our experiments.

Table 2 Some evidences' name and their meaning

Name	Meaning
CII or C	Body content text of current page
AT	Anchor texts link-in (backward propagation information)
S0	Information including title and h1 tag information
S1	pseudo-text defined in Section 2.1

2.2 Similarity computation for each evidence

Given a query q, for each evidence e_i, a cosine similarity is computed as the score between q and e_i, where both q and e_i are represented as weighted term vectors in traditional Vector Space Mode (VSM).

$$s(e_i, q) = \cos(\vec{e_i}, \vec{q}) = \frac{\vec{e_i} \bullet \vec{q}}{\left\|\vec{e_i}\right\| \left\|\vec{q}\right\|} \tag{3}$$

Where "•" means the inner product of two vectors. In our experiments, the *Lnu-Ltu* weighting scheme with pivoted normalization technique is used. Concretely, the *Lnu* weight of a term in a document is defined as

$$\frac{\dfrac{1+\log(tf)}{1+\log(average\ tf)}}{(1.0 - slope) \times pivot + slope \times \#\ of\ unique\ terms} \tag{4}$$

Where *slope* and *pivot* are two parameters for pivoted normalization, which can also be learned through training. *tf* is the term frequency in the document. *# of unique terms* is the number of different terms that occur in the document.

Pivoted normalization can be used to modify any normalization function thereby reducing the gap between the relevance and the retrieval probabilities [1]. Its effectiveness was confirmed in our previous results[5]. In our baseline run ICT04basic, the parameters' value is set as: *pivot*=196.661 for both text and query, *slope*=0.04 for content and *slope*=0.85 for query. Other runs using different parameters for different Web evidence.

Table 3 Parameters for different evidences (Refer to Table 2)

Evidence's Name	C	CII	AT	S0	S1
Pivot	196.661	196.661	196.661	192	196.661
Slope	0.04	0.18	0.04	0.49	0.19

2.3 Re-ranking strategies

To re-rank the first retrieval results may improve the final performance. Kiduk Yang's PPT[2] proposed a re-ranking strategy that to "keep top 5 static, boost potential homepages and file type page which contains at least 2 query terms".

Some heuristic strategies based on URL depth, URL words, anchor text and site compression were investigated in our experiments. The strategies are as follows:

For those results that have same similarities, first sort them according to their URL depth, and then tune their orders according to the following formula. If the page is a PDF file then improve its rank, and on the other side if its URL contains words such as "news", then decrease its rank.

$$ts(p) = \sum_{i \in \{URL, title, anchor\ text\}} w_i * n_i$$

Where n_i is the number of query terms that occur in field i, w_i the corresponding weight. In our experiments, the weights for URL, title and anchor texts are respectively set as 0.5, 0.25 and 0.25.

3. Experiments

3.1 Training

The queries from TREC 2003's home/name page task and topic distillation task were combined together to a mixed query set (called MIXED03) for training. .Gov is used as the Web collection.

Different combination functions were investigated and the parameters were learned through the training experiments. The training results are listed in Table 4.

Table 4 Training results

Run	Average Precision(AP)	Rel_Rt/Total query number	Description
M03_C	0.2656	321/350	Only using body content texts
M03_S0	0.2288	280/350	Only using some basic structure information such as h1, title
M03_S1	0.2995	320/350	Using more structure information mentioned in Section 2.1
M03_AT	0.4131	304/350	Using anchor texts
M03_CombSum	0.4869	341/350	CombSum of C, S0, AT
M03_CombMNZ	0.4415	342/350	CombMNZ of C, S0, AT
M03_CombMin	0.1659	309/350	CombMin of C, S0, AT
M03_LM	0.5082	339/350	Linear combination of C, S0, AT

Where C, AT,S0 and S1 are defined in Table 2. Rel_Rt is the number of the queries whose relevant pages are in the return list (1000 pages returned per query)among the total queries. It can be regarded as some kind of recall measure.

Several observations can be found from Table4:

[2] http://elvis.slis.indiana.edu/docs/widit_trec2003_files/frame.htm#slide0006.htm

- The result using S1 (M03_S1) outperforms the result using S0(M03_S0). It may mean that S1 provide more useful information that S0 does.
- Among all individual evidences, the AP result using AT(M03_AT) outperforms the results using any other individual evidence. However, the Rel_RT of M03_AT is not very high. This may mean the AT evidence can be used to improve the precision while losing some recall rate. On the contrary, the result using body content texts (M03_C) has the highest Rel_RT, while its AP is very low.
- Except CombMin, any result with other combination function outperforms any individual result in the sense of either AP or Rel_RT. And the linear combination function is the top-performance function in our training.

After training, some parameters can be learned to get best performances. And then the results and the above observations were applied in this year's task.

3.2 Official Runs

In this part we give the description and the results of the 5 official runs submitted, which are listed in Table5.

Table 5 Official runs

Run	Average Precision	Average S@10	Rel_Rt	Description
ICT04basic	0.3976	0.7467	219/225	Retrieval using a linear merging of C, S0 and AT introduced in 3.2
ICT04CIILC	0.3958	0.7600	210/225	CII instead of C in ICT04basic, others the same as ICT04basic
ICT04CIIS1AT	0.4250	0.8044	223/225	Linear merging of CII, S1, AT, the same as ICT04CIILC except S1 instead of S0
ICT04RULE	0.4219	0.8044	211/225	Re-rank the result of ICT04CIIS1AT using our re-rank algorithm
ICT04MNZ3	0.4279	0.7689	223/225	CombMNZ of CII, S1, AT

Here both CII and C mean that the runs use the body content text as the evidence for retrieval. The difference between them is that different slope and pivot parameters are used (see Table 3).

ICT04basic is the basic run for all our submitted runs, it can be regarded as the baseline run. ICT04basic achieves the Average Precision 0.39, the average S@10 0.7469, and Rel_RT 219 among 225. There are 6 queries missed the relevant documents in the 1000 submitted documents and these 6 queries are all known-item queries.

The parameters for *slope* and *pivot* used for content retrieval in ICT04basic and CT04CIILC are different, but their performance show that different content retrieval results have little effect on merging results. The results of ICT04basic, ICT04CIILC and ICT04CIIS1AT show that richer structure information such as S1 can provide better retrieval results in the mixed query task.

We used some heuristic strategies (see Section 2.3) to re-rank the retrieval results and intend to improve the retrieval performance. But the comparison of ICT04RULE and ICT04CIIS1AT shows the strategies don't work. But it doesn't mean that re-ranking methods are useless for the mixed query. We believe better re-ranking algorithms should be found.

ICT04MNZ3 is the run using CombMNZ combination method. It also improves the performance as it did in the training experiments.

4. Conclusion

Mixed query task is an interesting task for the Web track. Its queries are those that real search engines face. Our experiments show that richer structure information and a good combination method (such as CombMNZ) can provide better Web retrieval performance. Further work may include query classification or developing more precise re-ranking strategies.

References:

[1] A. Singhal, C. Buckley. M. Mitra. *Pivoted Document Length Normalization*. ACM SIGIR, 1996. http://citeseer.ist.psu.edu/singhal96pivoted.html

[2] E.A. Fox, J.A. Shaw. *Combination of Multiple Searches*. Proceedings of the 2nd Text REtrieval Conference (TREC-2), National Institute of Standards and Technology Special Publication 500-215, pp. 243-252, 1994. http://citeseer.ist.psu.edu/fox94combination.htm

[3] Joon Ho Lee. Analyses of Multiple Evidence Combination. Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval, 1997.

[4] Nick Craswell, David Hawking, Trystan Upstill. TREC12 Web and Interactive Tracks at CSIRO, TREC Report,2003.

[5] Hongbo Xu, Zhifeng Yang, Bin Wang, Bin Liu, Jun Cheng, Yue Liu, Zhe Yang, Xueqi Cheng, Shuo Bai. TREC-11 Experiments at CAS-ICT: Filtering and Web. TREC Report, 2002.

[6] In-Ho Kang, Kim . Query Type Classification for Web Document Retrieval. SIGIR '03.