Tianwang Search Engine at TREC 2005: Terabyte Track

YAN Hongfei, LI Jingjing, ZHU Jiaji, PENG Bo
Network and Distribution System Laboratory
School of Electronic Engineering and Computer Science
Peking University
Beijing, China, 100871
{yhf,ljj,zjj,pb}@net.pku.edu.cn

ABSTRACT

Tianwang for the first time participated in all three tasks of the Terabyte Track of TREC 2005 to explore its performance. All three tasks, including the adhoc task (find all the relevant documents with high precision), the efficiency task (find top-20 results for each of 50k-entry queries with efficiency and scalability) and the named page finding task (sometimes search a page by name), are based on a 426GB collection of 25.2 million pages taken from the .gov Web domain ("GOV2"). In the adhoc task with 50 topics, Tianwang returned at least one relevant document in top 10 for 42 topics. In the efficiency task, Tianwang returned at least one relevant document in top 20 for 44 of the 50 quires. In the named page task with 252 topics, Tianwang returned a desired page in top 10 for 99 topics; meanwhile, it failed to find a correct one for 120 topics.

Keywords

Search Engine, Evaluation.

1. INTRODUCTION

The Tianwang [Tianwang,2005] is a search engine developed and maintained by the network and distributed system laboratory of Peking University. As a retrieval system with the capability of searching billions of web pages, it is an ideal baseline system for experimenting on TREC tasks.

Tianwang works in a general model which most of search engines adopted, consisting of three components, a crawling process, an organizing process and a servicing process. The crawling process takes charge of collecting pages from the Web and stores them as an input for the following process. Then the organizing process answers for dereplicating pages fed from the previous process and creating indices for the remaining data. Finally the servicing process shows an interface to users for searching.

Our goals of this year's participating terabyte track were modest, to complete runs with the Tianwang system (software and partial idle machines), and to gain experience in the procedure of evaluations. The experience is important, so we could do better organizing the Chinese Web Tack of the CWIRF [CWIRF,2005] with the CWT100g collection [CWT100g,2004].

Actually we built a search engine based on the GOV2 data. Its searching results are shown on Figure 1.

Figure 1. The GOV2 Search Engine

The rest of the paper gives out an experimental work for terabyte retrieval including the adhoc task, the efficiency task and the named page finding task. We first describe the collection and tasks, then give the indexing environment, servicing process, experimental results, and finally discuss the results.

2. COLLECTION AND TASK SUMMARY

In this section, we conduct terabyte retrieval on GOV2 data to show the performance of Tianwang.

The terabyte track of TREC 2005 used the GOV2 corpus, which is made up of about 25,205,179 documents crawled from the .gov Web domain, comprising about 426 GB of document source.
There are three tasks in the terabyte track. The first one is the traditional task -- ad-hoc retrieval. There are 50 topics. Each topic is presented in the usual TREC format, as a tagged document with three fields that summarize the information need at different levels of detail: a short 'title' consisting of just a few terms, a longer, more detailed 'description' field, and a multi-sentence 'narrative' field.

For each topic, participants create a query and submit a ranking of the top documents (no more than 10,000) for that topic.

The efficiency task is to find top-20 results for each topic of 50,000 entries which are mined from query logs of an operational search engine. Queries must be created automatically from these topics; manual runs are not permitted for this task.

The named page finding task is to search for a page by name. In such cases, an effective search system will return that page at or near rank one. Roughly 150 new topics will be created for this task. A run consists of the top 1000 documents for each topic. No manual or interactive query modification is permitted in this task.

Furthermore, for each run, groups should record and report the system characteristics, such as indexing time in minutes, total number of CPUs in system and total processing time for all topics in seconds.

3. INDEXING ENVIRONMENT

We use eight idle machines/nodes of Tianwang system, which are available at that moment, to index all of the GOV2 corpus documents. Each machine has dual Xeon 2.8GHz CPUs with 2GB RAM running Red Hat Enterprise Linux AS release 4.

As far as TREC tasks are concerned, substituting the crawling process with the GOV2 corpus is a straightforward way to fulfill the TREC tasks with Tianwang. Due to different storage formats of TREC and Tianwang [TianwangStorageFormat,2003], it is necessary to transform the GOV2 into the data which can be accepted by the organizing process of Tianwang. Because tasks require returning all correct documents, we should bypass the dereplicating step and build indices directly. Then using the servicing process generates all the required results according to the topics.

Tianwang, as a full text retrieving system, is built with technologies of word-level segmentation and word position information. The organizing process first distributes all its input documents among different nodes in terms of site domain names. Nodes are independent each other, and indices are built on each of them. The two times in-memory inverted indexing algorithm is applied on each node. First we construct many inverted indices for small scale document set which is suitable for putting into the available memory. Then we execute multi-merging and construct the full indices of each node. The key steps of indexing process are listed as follows:

- Parsing pages. According to the HTML grammar, we analyze each page’s tags and structures, apply Chinese word segmentation and English grammar analyzer and extract index terms. During the procedure of analyzing, we record document frequencies (DF) of each index term and term frequencies (TF) of each index term per document, and get a lexicon file. Meanwhile, we save the parsing results to an analyzing result file.
- Building temporary inverted files. In terms of statistic DF and TF, it is easy to estimate the length of each index term. Then we reserve memory according to the length and reload the analyzing result file produced last step. The corresponding inverted index is done in memory and saved the result to the disk.
- Merging small inverted indices. The step begins with the multi-lane merging algorithm, compress index term coding, and generate the final inverted index file.

The index references documents for 21,593,661 of 25,205,168 documents of the GOV2 corpus. Total indexing time of eight machines is 1,514 minutes. The size of on-disk file structures/the final index file is 45.5GB. The reasons of missing 3,611,507 documents is still unknown, maybe lying in the process of transforming data format (TREC format to Tianwang format) or indexing documents due to some invalid documents or Tianwang software bugs.

4. SERVICING PROCESS

Figure 2 illustrates the servicing framework of Tianwang [Peng,2004].

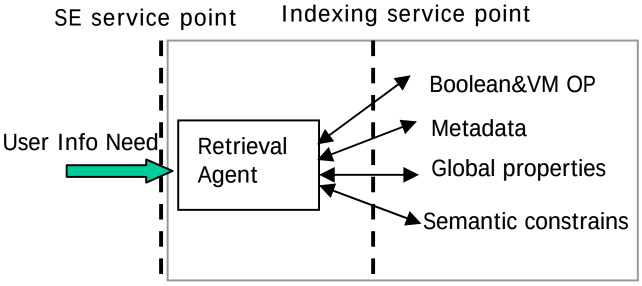

Figure 2. The Servicing Framework of Tianwang

The user first specifies a *User Information Need* which is parsed, at *SE Service Point*. Then *Retrieval Agent* collects retrieving results from all nodes and shows the ranked documents to the user. Relevant ranking is a linear combination of many ingredients [Lei, et al.,2001]. Among them, *Boolean&VM OP* is the most important ranking policy which is constructed on the Boolean model and the vector model[Salton,1971]. *Metadata* can be date, document format, site name, class label, etc. *Global properties* can be page rank[Page, et al.,1998], authoritative site lists, relevance feedback, manual compilation etc. Based on technologies of the natural language processing, *Semantic constrains* recognizes special semantic relations in each document.

In the experiments of terabyte track, our ranking policy is the combination of *Boolean&VM OP* and *Global properties* (page rank). We directly input title fields of topics to Tianwang without any changes.

5. EXPERIMENTAL RESULTS

We submitted 2 runs for the adhoc task, without and with page rank (TWTB05AD01 and TWTB05AD02 respectively), 1 run for the efficiency task (TWTB05EF01), and three runs for the named page finding run, without and with different applying methods of page rank (TWTB05NP01, TWTB05NP03 and TWTB05NP03). All submission runs are illustrated in the Table 1. If a run using technologies of link analysis, anchor text or other document structure, there will be a ‘yes’ tag in the corresponding filed.

Table 1. Submitted Runs

Run	**Link Analysis**	**Anchor Text**	**Other Doc Structure**
TWTB05EF01	-	-	-

TWTB05AD01	-	-	-
TWTB05AD02	yes	-	-
TWTB05NP01	-	-	-
TWTB05NP02	yes	-	-
TWTB05NP03	yes	-	-

In the adhoc task with 50 topics, Tianwang returned at least one relevant document in top 10 for 42 topics. In the efficiency task, Tianwang returned at least one relevant document in top 20 for 44 of the 50 quires. In the named page task with 252 topics, Tianwang returned a desired page in top 10 for 99 topics; meanwhile, it failed to find a correct one for 120 topics. More details are showed in Table 2, 3, 4.

Table 2. Submitted Runs for the GOV2 using TREC topics 751-800 and top 10,000 documents

Run	**Ranking policy** **(Bleean&VM OP *a +(1-a)* pagerank*100)**	**P@10**	**MAP**
TWTB05AD01	a=1	0.351	0.1128
TWTB05AD02	a=0.2	0.350	0.1118

Table 3. Submitted Runs for the GOV2 using TREC efficiency_topics 1-50,000 and top 20 documents

Run	**Ranking policy** **(Bleean&VM OP *a +(1-a)* pagerank*100)**	**P@10**
TWTB05EF01	a=1	0.285

Table 4. Submitted Runs for the GOV2 using TREC np_topics 601-872 and top 1,000 documents

Run	**Ranking policy** **(Bleean&VM OP *a +(1-a)* pagerank*100)**	**MRR**
TWTB05NP01	a=1	0.262
TWTB05NP02	a=0.2	0.261
TWTB05NP03	a=0	0.100

6. CONCLUSION

We archived the goal for the terabyte track, submitting all runs for three tasks of the track, and gained much experience in the procedure of evaluations.

The problems we encounter are TREC format and part of not indexed documents.

Each document in the GOV2 corpus is identified by a DocNo, which is not consisted with identifies in the Tianwang data. So we have to map the URLs of runs to corresponding DocNos. In the organizing process, Tianwang transformed the URLs into a decoding format, such as “%20” becoming “ ”. To guarantee the mapping process, we transformed the “url2id” file, provided with the GOV2 corpus, into the decode format – “url2id.decode”. Unfortunately, there are still a few entries in the submitting runs which could not find corresponding DocNos. We have not find where the bug lies. Additionally, there are 3,611,507 documents of the GOV2 corpus were not indexed due to unknown reasons.

7. ACKNOWLEDGMENTS

This work described therein is supported in part by PRC Ministry of Education grant 20030001076 and by NSFC grant 60435020.

8. REFERENCES

[CWIRF,2005] Chinese Web Information Retrieval Forum. http://www.cwirf.org/.

[CWT100g,2004] Chinese Web test collection. http://www.cwirf.org/SharedRes/DataSet/cwt100g.html.

[Lei, et al.,2001] M. Lei, J. Y. Wang, B. J. Chen, and X. M. Li, "Improved relevance ranking in WebGather," *Journal of Computer Science and Technology*, vol. 16, pp. 410-417, 2001.

[Page, et al.,1998] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank Citation Ranking: Bringing Order to the Web," Stanford Digital Library Technologies Project 1998.

[Peng,2004] B. Peng, "On Efficiency Optimization and Effectiveness Evaluation of Search Engine Retrieval System," Peking University,PhD, 2004, pp. 106.

[Salton,1971] G. Salton, *The SMART Retrieval System - Experiments in Automatic Document Processing*: Prentice Hall Inc., Englewood Cliffs, NJ, 1971.

[Tianwang,2005] Tianwang Search Engine. (http://www.tianwang.com).

[TianwangStorageFormat,2003] Tianwang storage format of raw web pages. http://net.pku.edu.cn/~webg/src/twformat/.