Using Normal PC to Index and Retrieval Terabyte Document -- THUIR at TREC 2004 terabyte track*

Yijiang Jin, Wei Qi, Min Zhang, Shaoping Ma
State Key Lab of Intelligent Tech. & Sys., CST Dept, Tsinghua University, Beijing 100084, China
{yjjin, z-m, msp}@tsinghua.edu.cn , qiwei00@tsinghua.org.cn

1 Introduction

This year, Tsinghua University Information Retrieval Group (THUIR) participated in the terabyte track of TREC for the first time. Since the document collection is as large as about 426G and we do not have super computers, our first and most import target is to complete the task in a reasonable low cost, both on the hardware system and time-consuming. This target was archived in such approaches: carefully data preprocess, data set reduction, optimization of algorithm and program. As the effect of the approaches, the task was completed in a normal high performance desktop PC with an indexing time not more than several ten hours and an acceptable retrieval time. Furthermore, the retrieval performance is not terrible. All experiments have been performed on TMiner IR system, developed by THUIR group last year.

2 Data preprocess

The data preprocess is mainly extracting text from html document and filter out all tags and scripts. We use a perl script that call HTML Parser to perform the extraction[1]. The .gov2 files were gunziped and directly piped to the perl script without saving to harddisk to reduce disk space use. The data size after preprocess is about 138G, less than 1/3 of original data set.

Another work of preprocess is extraction of anchor text from html document. Then the anchor text were sorted according to the link-to documents. The size of total anchor text is about 3.9G.

3 Data set reduction and reform

The data set reduction is achieved by replace the original document by its abstract. The abstract of html files were formed with some key field of the document. These fields include tile, text with bold, italic or head attribute. The abstract of PDF files, DOC files and PS files were made by simply take the first two hundred words of the document. The data set size was reduced from 138G to 22G after such process.

We used both full text and abstracts in the experiments. In some experiments, anchor text were merged to the link-to documents.

4 Indexing and retrieval

Since the hardware limit, the whole data set was divided into several parts. Individual indexes were built on each sub-collection and then merged finally. Stop-words filtering and stemming technology were applied. The index was compressed to reduce size. The size of full-text index is about 121G and the size of abstract index is about 16G.

In this retrieval, both short query and long query were used. BM2500 retrieval formula was used to

* Supported by by Chinese National Key Foundation Research & Development Plan (973) (Grant No.2004CB318108) and Chinese Natural Science Foundation (Grants No. 60223004, 60321002, 60303005).

calculate the similarity between query topic and documents.

5 Hardware and software environment

We used two normal PCs to compete the task.

	CPU	RAM	Harddisk	OS	Complier
PC1	AMD Athlong64 3000+	1.5G[1]	400G	Linux	GCC
PC2	Intel Celeron 2.0G	512M	200G	Windows	VS.Net

Figure 1: Configuration of PCs in the experiments

[1] 768M in THUIRtb2

6 Results submitted

	THUIRtb1 (unofficial)	THUIRtb2	THUIRtb3	THUIRtb4	THUIRtb5	THUIRtb6
Query fields	Title	Title description narrative	Title	Title description narrative	Title	Title
Percentage of collection indexed	15%	15%	100%	100%	100%	100%
Use of anchor?	Y	Y	Y	Y	N	N
Use of structure?	Y	Y	N	N	N	N
Indexing time (minutes)	170	170	1024	1024	1920	1920
Ave time to return 20 docs (seconds)	2	18	9	55	15	16
# CPU	1	1	1	1	1	1
RAM amount (GB)	0.75	0.75	1.5	1.5	0.5	0.5
On-disk file (GB)	16	16	121	121	41.5	41.5
Using link analysis	N	N	N	N	N	N
Using anchor text	Y	Y	Y	Y	N	N
Using doc structure	Y	Y	N	N	N	N
MAP	-	0.0557	0.2198	0.2453	0.2437	0.2036
R-prec	-	0.0983	0.2968	0.3049	0.3181	0.2864
P@5	-	0.3469	0.4857	0.5796	0.4816	0.5020
P@10	-	0.3061	0.4714	0.5327	0.4755	0.4735

Reference

[1] Comprehensive Perl Archive Network(CPAN), http://www.cpan.org/