Overview of the TREC 2004 Terabyte Track

Charles Clarke
University of Waterloo
claclark@plg.uwaterloo.ca

Nick Craswell
Microsoft Research
nickcr@microsoft.com

Ian Soboroff
NIST
ian.soboroff@nist.gov

Abstract

The Terabyte Track explores how adhoc retrieval and evaluation techniques can scale to terabyte-sized collections. For TREC 2004, our first year, 50 new adhoc topics were created and evaluated over a a 426GB collection of 25 million documents taken from the .gov Web domain. A total of 70 runs were submitted by 17 groups. Along with the top documents, each group reported average query times, indexing times, index sizes, and hardware and software characteristics for their systems.

1 Introduction

Early retrieval test collections were small, allowing relevance judgments to be based on an exhaustive examination of the documents but limiting the general applicability of the findings. Karen Sparck Jones and Keith van Rijsbergen proposed a way of building significantly larger test collections by using pooling, a procedure adopted and subsequently validated by TREC. Now, TREC-sized collections (several gigabytes of text and a few million documents) are small for some realistic tasks, but current pooling practices do not scale to substantially larger document sets. Thus, there is a need for an evaluation methodology that is appropriate for terabyte-scale document collections. A major research goal of the Terabyte track is to better define where our measures break down, and to explore new measures and methods for dealing with incomplete relevance judgments.

Current tasks that are evaluated using large web collections, such as known-item and high-precision searching, focus on the needs of the common web searcher but also arise from our inability to measure recall on very large collections. Good estimates of the total set of relevant documents are critical to the reliability and reusability of test collections as we now use them, but it would take hundreds of different systems, hundreds of relevance assessors, and years of effort to produce a terabyte-sized collection with completeness of judgments comparable to a typical TREC collection. Hence, new evaluation methodologies and ways of building test collections are needed to scale retrieval experiments to the next level.

The proposal for a TREC Terabyte Track was initiated at a SIGIR workshop in 2003 and accepted by the TREC program committee for TREC 2004. This report describes the details of the task undertaken, the runs submitted, and the range of approaches taken by the participants.

2 The Retrieval Task

The task is classic adhoc retrieval, a task which investigates the performance of systems searching a static set of documents using previously-unseen topics. This task is similar to the current Robust Retrieval task, and to the adhoc and VLC tasks from earlier TREC conferences.

2.1 Collection

This year's track used a collection of Web data crawled from Web sites in the `.gov` domain during early 2004. We believe that this collection ("GOV2") contains a large proportion of the crawlable pages in `.gov`, including HTML and text, plus the extracted text of PDF, Word and postscript files. By focusing the track on a single, large, interconnected domain we hoped to create a realistic setting, where content, structure and links could all be fruitfully exploited in the retrieval process.

The GOV2 collection is 426GB in size and contains 25 million documents. While this collection contains less than a full terabyte of data, it is considerably larger than the collections used in previous TREC tracks. For TREC 2004, the collection was distributed by CSIRO in Australia on a single hard drive for a cost of A$1200 (about US$800).

2.2 Topics

NIST created 50 new topics for the track. Figure 1 provides an example. As in the past, the title field may be treated as a keyword query, similar to the queries stereotypically entered by users of Web search systems. The description field provides a slightly longer statement of the topic requirements, usually expressed as a single complete sentence or question. Finally, the narrative supplies additional information necessary to fully specify the requirements, expressed in the form of a short paragraph. While keywords from the title are usually repeated in the description, they do not always appear in the narrative.

2.3 Queries

For each topic, participants created a query and submitted a ranking of the top 10,000 documents for that topic. Queries could be created automatically or manually from the topic statements. As for all TREC tasks, automatic methods are those in which there is no human intervention at any stage, and manual methods are everything else. For most runs, groups could use any or all of the topic fields when creating queries from the topic statements. However, each group submitting an automatic run was required to submit an automatic run that used just the title field.

2.4 Submissions

Each group was permitted to submit up to five experimental runs. Each run consists of the top 10,000 documents for each topic, along with associated performance and system information. We required 10,000 documents, since we believe this that information may useful during later analysis to help us better understand the evaluation process.

In addition to the top 10,000 documents, we required each group to report details of their hardware configuration and various performance numbers, including the number of processors, total RAM (GB), on-disk index size (GB), indexing time (elapsed time in minutes), average search time (seconds), and hardware cost. For the number of processors, we requested the total number of CPUs in the system, regardless of their location. For example, if a system is a cluster of eight

```
<top>
<num> Number: 705

<title>
Iraq foreign debt reduction

<desc> Description:
Identify any efforts, proposed or undertaken, by world governments to seek
reduction of Iraq’s foreign debt.

<narr> Narrative: Documents noting this subject as a topic for
discussion (e.g. at U.N. and G7) are relevant. Money pledged for
reconstruction is irrelevant.

</top>
```

Figure 1: Terabyte Track Topic 705

dual-processor machines, the number of processors is 16. For the hardware cost, we requested an estimate in US dollars of the cost at the time of purchase.

Some groups may subset a collection before indexing, removing selected pages or portions of pages to reduce its size. Since subsetting may have an impact on indexing time and average query time, we asked each group to report the fraction of pages indexed.

For search time, we asked the groups to report the time to return the top 20 documents, not the time to return the top 10,000, since this number better reflects the performance that would be seen by a user. It was acceptable to execute a system twice for each query, once to generate the top 10,000 documents and once to measure the execution time for the top 20, provided that the top 20 results were the same in both cases.

2.5 Judgments

The top 85 documents of two runs from each group were pooled and judged by NIST assessors. The judgments used a three-way scale of “not relevant”, “relevant”, and “highly relevant”.

3 Submitted Runs

Figures 2 and 3 provide an overview submitted runs. The first two columns give the group and run ids. The third column lists the topic fields — Title (“T”), Description (“D”) and Narrative (“N”) — that were used to create the query. In all cases queries were generated automatically from these fields. No manual runs were submitted. The next three columns indicate if link analysis techniques, anchor text, or other document structure was used in the ranking process. The third-last column gives the average query time required to generate the top 20 results, and the second-last column gives the time to build the index in hours. The last column gives the mean average precision achieved by each run.

Group Id	Run Id	Topic Fields	Links?	Anchors?	Structure?	Query Time (s)	Index Time (h)	MAP
cmu.dir.callan	cmuapfs2500	TDN	N	N	N	600	20.0	0.284
	cmutufs2500	T	N	N	N	240	20.0	0.248
	cmutuns2500	T	N	N	N	75	20.0	0.207
dubblincity.u	DcuTB04Base	T	N	N	N	2	408.7	0.118
	DcuTB04Ucd1	TDN	N	Y	N	84	883.7	0.076
	DcuTB04Wbm25	T	N	N	Y	2	760.8	0.079
	DcuTB04Combo	T	N	Y	Y	2	906.0	0.033
	DcuTB04Ucd2	TDN	N	Y	N	15	457.5	0.070
etymon	nn04tint	T	N	N	N	25	44.8	0.112
	nn04eint	T	N	N	N	78	44.8	0.074
	nn04test	T	N	N	N	46	44.8	0.028
hummingbird	humT04l	T	N	N	Y	115	100.0	0.224
	humT04dvl	T	N	N	Y	142	100.0	0.212
	humT04vl	T	N	N	Y	119	100.0	0.221
	humT04l3	T	N	N	Y	49	100.0	0.155
	humT04	T	N	N	Y	50	100.0	0.196
iit	iit00t	T	N	N	N	23	8.0	0.210
	robertson	T	N	N	N	42	8.0	0.200
jhu.apl.mcnamee	apl04w4tdn	TDN	N	N	N	10000	0.0	0.034
	apl04w4t	T	N	N	N	10000	0.0	0.027
max-planck.theobald	mpi04tb07	T	Y	N	Y	6	42.0	0.125
	mpi04tb09	TD	Y	N	Y	9	42.0	0.123
	mpi04tb101	TD	Y	N	N	9	42.0	0.081
	mpi04tb81	TD	Y	N	N	9	42.0	0.092
	mpi04tb91	TD	Y	N	N	9	42.0	0.092
microsoft.asia	MSRAt3	T	N	Y	Y	1	11.6	0.171
	MSRAt4	T	N	Y	Y	1	11.6	0.188
	MSRAt5	T	N	Y	Y	1	11.6	0.190
	MSRAt2	T	N	N	Y	1	11.6	0.092
	MSRAt1	T	N	N	Y	1	11.6	0.191
rmit.scholer	zetbodoffff	T	N	N	N	25	13.5	0.219
	zetanch	T	N	Y	N	2	13.6	0.217
	zetplain	T	N	N	N	2	13.5	0.223
	zetfuzzy	T	N	Y	N	2	13.6	0.131
	zetfunkyz	T	N	Y	N	3	13.6	0.207

Figure 2: Summary of Submitted Runs (Part 1)

Group Id	Run Id	Topic Fields	Links?	Anchors?	Structure?	Query Time (s)	Index Time (h)	MAP
sabir.buckley	sabir04td3	D	N	N	N	18	14.0	0.117
	sabir04ta2	TDN	N	N	N	9	14.0	0.172
	sabir04tt	T	N	N	N	1	14.0	0.116
	sabir04td2	D	N	N	N	3	14.0	0.121
	sabir04tt2	T	N	N	N	1	14.0	0.118
tsinghua.ma	THUIRtb5	T	N	N	N	15	32.0	0.244
	THUIRtb4	TDN	N	Y	N	55	17.0	0.245
	THUIRtb3	T	N	Y	N	9	17.0	0.220
	THUIRtb2	TDN	N	Y	Y	18	2.8	0.056
	THUIRtb6	T	N	N	N	16	32.0	0.204
u.alaska	irttbtl	T	N	N	Y	5	30.0	0.009
u.amsterdam.lit	UAmsT04TBm1	T	N	Y	Y	90	4.3	0.044
	UAmsT04TBanc	T	N	Y	N	1	0.3	0.013
	UAmsT04TBm1p	T	N	Y	Y	90	4.3	0.043
	UAmsT04TBtit	T	N	N	Y	20	4.0	0.039
	UAmsT04TBm3	T	N	Y	Y	90	4.3	0.043
u.glasgow	uogTBQEL	TDN	N	N	N	46	200.6	0.307
	uogTBPoolQEL	TDN	N	N	N	46	200.6	0.231
	uogTBBaseS	T	N	N	N	4	200.6	0.271
	uogTBAnchS	T	N	Y	N	3	501.7	0.269
	uogTBBaseL	TDN	N	N	N	28	200.6	0.305
u.mass	indri04AWRM	T	N	N	N	39	5.9	0.284
	indri04AW	T	N	N	N	7	5.9	0.269
	indri04QLRM	T	N	N	N	26	5.9	0.253
	indri04QL	T	N	N	N	1	5.9	0.251
	indri04FAW	T	N	Y	Y	52	21.6	0.279
u.melbourne	MU04tb3	T	Y	Y	N	0.08	2.5	0.043
	MU04tb2	T	N	Y	N	0.08	2.5	0.063
	MU04tb4	T	Y	Y	N	0.36	13.0	0.268
	MU04tb1	T	N	N	N	0.08	1.7	0.266
	MU04tb5	T	Y	Y	N	0.08	2.5	0.064
upisa.attardi	pisa4	T	Y	Y	Y	3	16.0	0.103
	pisa3	T	Y	Y	Y	3	16.0	0.107
	pisa2	T	Y	Y	Y	3	16.0	0.096
	pisa1	T	Y	Y	Y	1	16.0	0.050

Figure 3: Summary of Submitted Runs (Part 2)

4 Overview of Systems

Most groups contributed papers to this notebook, and we refer the reader to the these papers for complete details about individual systems. In the remainder of this section, we summarize the range of approaches taken by the groups and highlight some unusual features of their systems.

4.1 Hardware and Software

The cost and scale of the hardware varied widely, with many groups dividing the documents across multiple machines and searching the collection in parallel. At one extreme, the group from the University of Alaska's Arctic Region Supercomputing Center used 40 nodes of the NCSA "mercury" TeraGrid cluster, which cost over US$10 million. At the other extreme, the group from Tsinghua University used a single PC with an estimated cost of US$750.

To index and search the collection, most groups used custom retrieval software develop by their own group or by an associated group. One exception is the University of Alaska, which used MySQL (finding a bug in the process). Hummingbird used their commercial SearchServer$^{\mathrm{tm}}$ system. Etymon Systems used their Amberfish package, which they have released as open source (`etymon.com/tr.html`). Both CMU and University of Massachusetts used Indri, a new indexing and retrieval component developed by the University of Massachusetts for the Lemur Toolkit.

4.2 Indexing

Overall, indexing methods were fairly standard. Most groups applied stopping and stemming methods. However, at least three groups, the University of Massachusetts, CMU, and Etymon Systems did not remove stopwords, despite the size of the collection. Several groups compressed the index to improve performance and reduce storage requirements, including the University of Glasgow, the University of Melbourne, and the University of Pisa. Sabir implemented compressed indices, but did not use them in their final runs.

Since a large portion the collection consists of HTML, many groups applied special processing to the anchor text or to specific fields within the documents. For example, Dublin City University generated surrogate anchor text documents, comprised of the anchor text of inlinks to a document. The Indri system supports the indexing of arbitrary document fields, and this facility was used to index various fields of HTML documents (title, h1, h2, etc.). The University of Pisa performed extensive preprocessing, extracting page descriptions and categories from Dmoz, collecting links and anchor texts, and identifying specific fields within HTML documents.

The most unusual approach was taken by the University of Amsterdam group, who indexed only document titles and anchor text. The resulting indexes are small: 1.4GB for the titles covering 83% of the documents, and 0.1 GB for the anchors covering 6% of the documents. This very selective indexing produced a 20 minute indexing time and a 1 second average query time without the need for special performance optimizations.

Figure 4 plots the fastest indexing times, ignoring all but the fastest time from each group. Indexing a 426GB collection in under 14 hours implies an indexing rate of over 30GB/hour. However, most of these groups parallelized the indexing process or indexed only a subset of the collection. The fastest reported "indexing" time, zero, does not appear on the figure. The group reporting this indexing time, JHU/APL, did not index the collection at all. Instead, they searched it with a DFA executed by a Perl script.

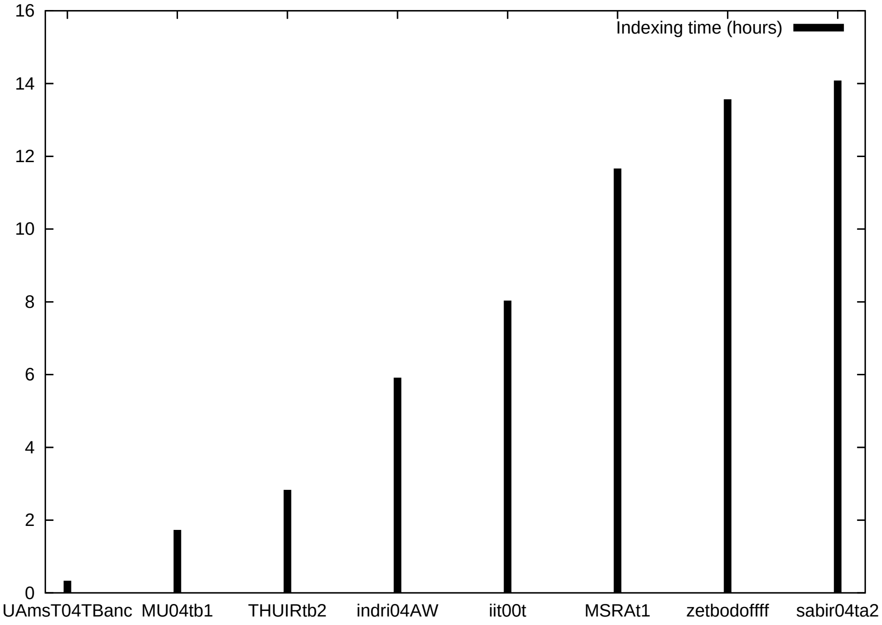

Figure 4: Indexing Time (hours) — Top 8 Groups

4.3 Query Processing

Although adhoc retrieval has been a mature technology for many years, a surprising variety of retrieval formulae were used, including Okapi BM25, cosine, and methods based on language modeling and divergence from randomness. Proximity operators were used by several groups including University of Pisa and CMU. Link analysis methods were used in 17% of the runs, anchor text was used in 37%, and other document structure (usually document titles) was used in 36%. Several groups expanded queries using pseudo-relevance feedback. This wide range of methods suggests that "best practice" for information retrieval over large Web collections may not be as well established as some believe.

Figure 5 plots the eight fastest average query times, ignoring all but the fastest run from each group. The run submission form requested the average query time in seconds, rather than milliseconds, and the impact of this error can be seen in the figure. Five groups reported an average query time of "1 second" and two groups reported a time of "2 seconds". The query time reported by the University of Melbourne, 0.08 seconds, is roughly equal to the time typically required for a single disk access.

Figure 6 plots the title-only runs achieving the best mean average precision, ignoring all but the best-performing run from each group. The curve is relatively flat, with all eight groups achieving reasonable performance.

5 The Future

For TREC 2005, the Terabyte Track will continue to use the GOV2 collection, giving us a total of 100 topics over the collection. We plan to collect more and better information regarding system performance, with the hope that system performance comparisons can be made more realistically. Finally, a known-item retrieval task may be added to the track.

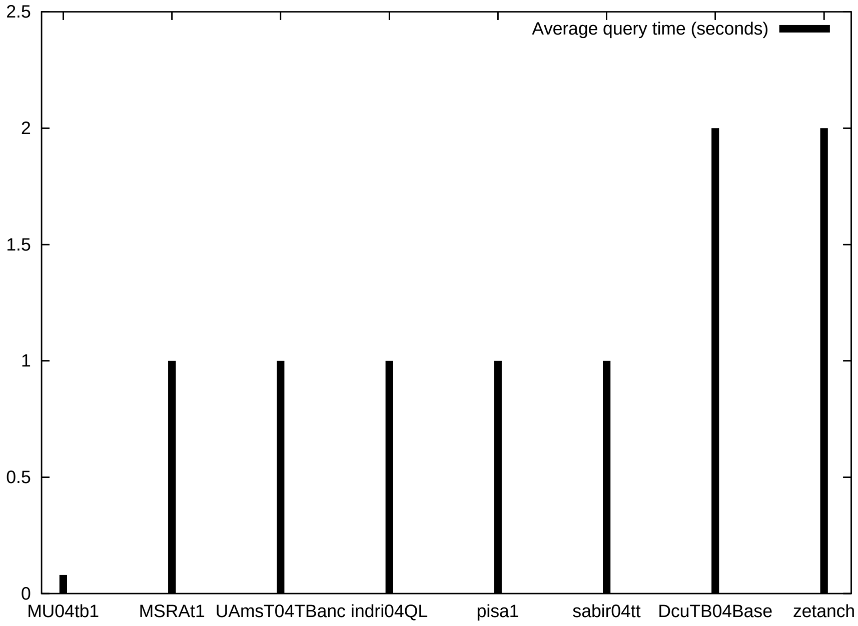

Figure 5: Average Query Time (seconds) — Top 8 Groups

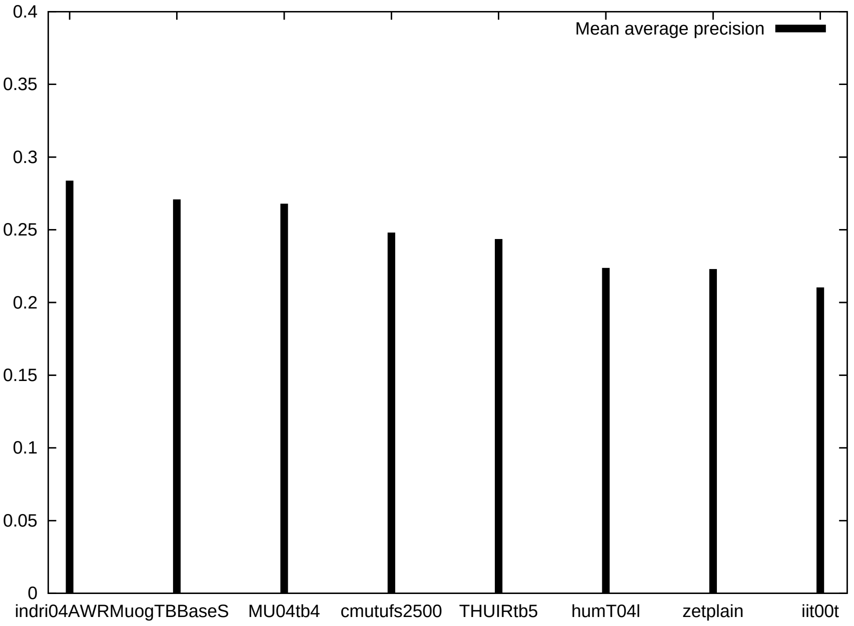

Figure 6: Mean Average Precision (MAP) — Top 8 Groups

6 Acknowledgments

We thank Doug Oard for hosting our Web crawler at the University of Maryland, and CSIRO for processing and distributing the GOV2 collection.