Amberfish at the TREC 2004 Terabyte Track

Nassib Nassar
Etymon Systems
Research Triangle Park, NC
nassar@etymon.com

Abstract

The TREC 2004 Terabyte Track evaluated information retrieval in large-scale text collections, using a set of 25 million documents (426 GB). This paper gives an overview of our experiences with this collection and describes Amberfish, the text retrieval software used for the experiments.

1 Preface

This is the first year of the Terabyte Track and the first time we have participated directly in the TREC conference. It is also the first time that the Amberfish software (etymon.com/tr.html) has been used in TREC. Our goals for this ambitious track were simply to complete the task and to gain some rudimentary experience with evaluation and the Terabyte collection.

This paper presents a summary of the Amberfish software followed by a brief discussion of this year's Terabyte Track.

2 Amberfish

Amberfish is open source text retrieval software developed by the author starting in 1998 and distributed by Etymon Systems, Inc. The project was based on lessons learned from previous implementation experiences with Isearch (1994) and freeWAIS (1992) at the Clearinghouse for Networked Information Discovery and Retrieval (CNIDR). Isearch was the search component of Isite, an open source Z39.50 implementation [3, 5], and freeWAIS was an open source version of WAIS [2]. More recently, Amberfish has been coupled with GIR [6] and Z39.50 for experiments in distributed searching.

The core of Amberfish provides a set of general purpose indexing and searching operations, with conventional support for Boolean queries, right truncation, phrase

searching, relevance ranking, multiple documents per file, incremental indexing, and stemming. Some novel features are indexing of semi-structured text (XML), structured queries for selecting field path subtrees, hierarchical results showing field relationships, and automatic searching across multiple indexes.

The software consists of a C/C++ library and a Unix-like command line interface on the front end. The central data structures are a dictionary and linked postings lists [4], with a simple prefix B-tree used to store the inverted file. The postings can be converted to sequential lists with an optional second indexing pass. Several speed optimizations are used in indexing, such as merge update of the B-tree [1]. Additional files optionally store word positions (for phrase/proximity) and field structures, both associated with individual postings.

3 Terabyte Track

The first few weeks of handling the Terabyte collection primarily entailed discovering all of the approaches that were *not* going to work. The collection consisted of 25 million web pages from the .GOV domain, or about 426 GB. The initial plan was to process the collection using HTML-to-XML conversion software such as HTML Tidy and html2text, so that the structure could be indexed by Amberfish. However, the software was unable to convert successfully more than 99% of the documents. With limited time remaining and in order to avoid confusion, it was determined to index the documents as plain text, although this unfortunately meant that tag element and attribute names would be indexed as words. In addition, the relevance scoring function for "bag of words" queries was not implemented in time for submitting query runs. As a result, the plan changed to finishing the task and leaving evaluation until next year.

Three runs were submitted. The first was produced by the simple union of documents matching title terms from the topic. The second was a simple intersection. The third was a combination of union and intersection, with results from the intersection weighted more heavily. The results were poor enough that they need not be included here. In particular the lack of attention to word proximity seems to have hurt precision. However, this could also be a result of our incomplete scoring function, which in effect was a very simplified variation of NTC for document weights with no term weighting.

Although a Porter stemmer was used, no stopwords were removed. Even so the indexing process was very fast, under 45 hours with fairly modest hardware: 2 GB RAM on a Xeon 3.06 GHz system, storing 279 GB of index files. (The system had quite a bit more memory which was not used in these experiments.) The index was partitioned into 27 subindexes and the searches distributed over them.

All indexing and searching was done using the available "stock" version of Amberfish, which includes options for outputting results in TREC-run format. It is hoped that prospective participants, especially students, will find the software to be a useful tool while learning about TREC.

4 Acknowledgements

I owe special thanks to Gregory Newby, of the Arctic Region Supercomputing Center, who generously offered his time, suggestions, and access to computer systems over several months. Many others have contributed directly or indirectly to the Amberfish project, and a few of them must be gratefully mentioned here: Kevin Gamiel, James Fullton, Erik Scott, Edward Zimmermann, and David Green.

References

[1] CUTTING, D., AND PEDERSEN, J. Optimizations for dynamic inverted index maintenance. In *13th International Conference on Research and Development in Information Retrieval* (Brussels, Belgium, 1990).

[2] FULLTON, J. WAIS. In *Intelligent Information Retrieval: The Case of Astronomy and Related Space Sciences*, A. Heck and F. Murtagh, Eds. Kluwer, Dordrecht, 1993, pp. 113–118.

[3] GAMIEL, K., AND NASSAR, N. Structural components of the Isite information system. In *Z39.50 Implementation Experiences*, P. Over, R. Denenberg, W. E. Moen, and L. Stovel, Eds., National Institute of Standards and Technology Special Publication 500-229. U.S. Department of Commerce, 1995, pp. 71–74.

[4] HARMAN, D., AND CANDELA, G. Retrieving records from a gigabyte of text on a minicomputer using statistical ranking. *Journal of the American Society for Information Science 41*, 8 (1990), 581–89.

[5] KUNZE, J. J., AND RODGERS, R. P. C. Z39.50 in a nutshell. Lister Hill National Center for Biomedical Communications, National Library of Medicine, July 1995.

[6] NEWBY, G. B., GAMIEL, K., AND NASSAR, N. Secure information sharing and information retrieval infrastructure with GridIR. In *Intelligence and Security Informatics: Proceedings of the First NSF/NIJ Symposium* (2003), H. Chen et al., Eds., Springer-Verlag, p. 389.