High-Performance, Distributed Information Retrieval

Kenneth Baclawski* and J. Elliott Smith
Northeastern University
College of Computer Science
Boston, Massachusetts 02115
{kenb,esmith}@ccs.neu.edu

We propose an information retrieval (IR) system called KEYNET which is a high-performance, distributed search engine for locating information objects in a large subject-specific corpus. The system could handle up to a few million information objects with performance at a level of hundreds of queries per second (with current workstation technology).[1] We have developed a prototype keynet system for testing the validity of the basic ideas and also to get preliminary performance results.

A KEYNET system requires the development of a subject-specific ontology that is understandable to a literate practitioner of the field. A keynet ontology represents knowledge using a directed graph of conceptual categories and relationships between them. The Unified Medical Language System (UMLS) developed by the National Library of Medicine is an example of such an ontology[HL93, LHM93].

Each information object must be annotated with a small directed graph (called a *keynet*) that indicates what portion of the ontology relates to the content of the object. Keynets are semantically intermediate between keywords and semantic networks[Lev92]. Keynets provide an overall framework that generalizes many commonly used mechanisms for information retrieval, such as: subject classification schemes, keywords, document abstracts, reviews, content labels for non-textual information objects, properties such as author or date of publication, ranges of text strings such as "wild card" match strings, and ranges of quantities. The KEYNET system allows a uniform treatment of these disparate techniques in a system that permits a great deal of flexibility compared to traditional database and information retrieval systems. For example, one could combine all of the above mechanisms in a single system, and one can easily add new features to the ontology, such as new attributes and keywords.

In addition, the keynet framework allows for sequences of concepts linked by relationships and expressed in natural language using phrases, clauses, sentences and paragraphs. No current systems use such a capability, so there is no evidence that it would have a significant

*This material is based upon work supported by the National Science Foundation Grant No. IRI-9117030.

[1]By comparison, the LEXIS legal information retrieval system of Mead Data Central, answers 250,000 queries per day, an average of 3 queries per second, although the peak load is much higher.

impact on retrieval effectiveness. Nevertheless, the KEYNET system establishes that there is no technological or user-interface barrier to such semantically rich IR methods.

Some examples of information objects that would be well-suited to retrieval using the KEYNET technique include:

- Scientific research papers. This was the original motivation for the development of the technique. We are developing natural language processing tools for extracting keynets directly from the document[BFH+93].
- Scientific data files. Such files must be accompanied by (or include) a description of the contents. If such a description is structured, then it forms a keynet.
- Satellite images. The geographic coordinates, time taken, etc. can be incorporated into a keynet.
- Videotapes. The director, producer, stars, etc., along with a structured description of the content, forms a keynet.

The KEYNET technique is designed to be used in a highly distributed environment. It is assumed that the information objects themselves are widely distributed. At the KEYNET site itself, the keynets are kept on disks and distributed among the nodes of a local-area network. The index to the keynets is kept in the main memories of the nodes of the LAN. Processing of queries as well as insertion of new keynets is done using distributed algorithms.

Each keynet contains information about locating and acquiring the actual information object. The KEYNET system is only concerned with finding information objects. Acquiring (and paying for) objects is an independent issue.

The user's computer is responsible for presentation (user-interface) services. To accomplish this the user must have a copy of the ontology. Using current technology, this would be kept on a CD-ROM. We have developed a prototype interface of this type[BF93].

Queries and responses are sent over the network. The prototype achieves response times that are so fast that access to the local CD-ROM drive would generally be slower than access to the search engine. The prototype system uses the datagram protocol (UDP) of the Internet TCP/IP protocol family.

At the KEYNET site, an interface computer is responsible for relaying query requests to one of the search engine computers. The search engine itself is a collection of processors (or more precisely server processes) joined by a high-speed LAN. A query is itself a keynet. Queries are answered by fragmenting them into terms that are matched against similar fragments obtained from the information objects. A scatter/gather algorithm is employed to distribute the query terms and collect the matching objects. Relevance is measured using standard vector methods, specifically the cosine measure[Sal89].

Responses are sent directly to the requester from the search engine processor that collects the search result. The prototype differs from the proposed architecture only in that it randomly generates the keynet repository as well as queries sent to it.

The prototype runs on a network of up to 8 sparcstations connected by a twisted-pair network. The network is not dedicated to our research project. Among other results, we found that it is feasible to implement a high-performance search engine that "borrows" underutilized resources on a network of workstations.

The prototype is fully distributed, using a pure message-passing communication mechanism. All messages are one-way: no process ever waits for a reply to a message. The memory model is local, i.e., a "shared nothing" system. The individual nodes are implemented as servers. Specifically, they are implemented as connectionless, multi-threaded, interrupt-driven, stateless servers. Threads explicitly yield control and are never preempted, but they can be interrupted. Each server is responsible for a fixed amount of memory, chosen to be small enough for page faulting to be unusual. The indexing uses a hash algorithm using direct addressing with secondary hashing in the event of a collision. Collisions did not have an impact on performance even when the hash tables were 90% full. Each document keynet was generated randomly and had 200 index terms. This corresponds roughly to an document abstract that is around 100 to 150 words long, or equivalently, to a content label with 200 attributes. Queries had 10 terms each. In one run, the throughput for an 8-node network indexing 80,000 documents was 900 queries/sec, with a median response time of 2.3 seconds. At a load of 400 queries per second, the median response time was 0.3 seconds, and more than 95% of the queries were answered in less than 0.6 seconds.

References

[BF93] K. Baclawski and N. Fridman. M&M-Query: Database support for the annotation and retrieval of biological research articles. Technical Report NU-CCS-94-07, Northeastern University, College of Computer Science, 1993.

[BFH+93] K. Baclawski, R. Futrelle, C. Hafner, M. Pescitelli, N. Fridman, B. Li, and C. Zou. Data/knowledge bases for biological papers and techniques. In *Proc. Sympos. Adv. Data Management for the Scientist and Engineer*, pages 23–28, 1993.

[HL93] Betsy L. Humphreys and Donald A.B. Lindberg. The UMLS project: making the conceptual connection between users and the information they need. *Bulletin of the Medical Library Association*, 81(2):170, Apr 1 1993.

[Lev92] Robert Levinson. Pattern associativity and the retrieval of semantic networks. *Computers and Mathematics with Applications*, 23(6-9):573–600, 1992.

[LHM93] D.A.B. Lindberg, B.L. Humphreys, and A.T. McCray. The Unified Medical Language System. *Methods of information in medicine*, 32(4):281, Aug 1 1993.

[Sal89] Gerard Salton. *Automatic Text Processing.* Addison-Wesley, Reading, MA, 1989.