Providing Government Information on the Internet: Experiences with THOMAS

W. Bruce Croft and Robert Cook
Computer Science Department
University of Massachusetts
Amherst, MA 01003-4610
Tel: 413-545-0463
E-mail: croft@cs.umass.edu

Dean Wilder
Library of Congress
ITS 9332, 101 Independence Avenue
Washington, D.C. 20540
Tel: 202-707-9629
E-mail: dwil@dwil.loc.gov

ABSTRACT
The THOMAS system is designed to make legislative information available to the general public over the Internet, and can be regarded as a prototype of a government digital library. As a joint project between the Library of Congress and the University of Massachusetts, THOMAS provides significant opportunities for studying the retrieval and interface techniques that are needed to support effective access to complex information in an Internet-based environment. Preliminary experience with THOMAS indicates that there is significant interest in this type of information, and that queries tend to be shorter than those studied in retrieval evaluations such as TREC. Techniques such as query processing, query expansion, and morphological processing all need to be incorporated and improved.

KEYWORDS: Information Retrieval, Internet, Query Processing

INTRODUCTION
In mid-December 1994, the Library of Congress was requested by the newly-elected leadership of the House of Representatives to develop a comprehensive legislative information system for the Internet. This new system was to be the public distribution point for information on the activities of Congress, including all versions of legislation under consideration, the complete text of the Congressional Record, the House and Senate calendars, and e-mail addresses and links to other legislative information resources on the Internet.

The Library of Congress was designated as the primary site for this information, of which all or part had previously been made available through various gopher and telnet sites at the Senate, the House, the Government Printing Office, and a variety of commercial services. Furthermore, the new congressional leadership wanted the new system (THOMAS) to incorporate advanced retrieval techniques that would give the general public simple access to the legislative information through the World Wide Web (WWW).

THOMAS is a joint project between the Library and the Center for Intelligent Information Retrieval (CIIR) at the University of Massachusetts. The CIIR is an NSF-funded consortium involving the university, the State of Massachusetts, and both business and government partners. The CIIR does basic research and technology transfer in the area of text-based information systems. For THOMAS, the CIIR provided the retrieval software (INQUERY), and customized it as needed. The Library of Congress staff developed the interface and the extensions for converting the databases to HTML (the markup language for the WWW).

The Library has previously integrated INQUERY with its Web Server to make a variety of searchable databases available to the public, such as Civil War photographs, early motion pictures, sound recordings, POW/MIA records from the Vietnam War, and Country Studies from the Federal Research Division. This material falls naturally into the category of a digital library, and we believe that THOMAS is also a prototype of a digital library based on government information. The primary characteristics of a digital library from our point of view are providing access to a significant amount of valuable information over a network. The additional feature of "free" access to the general public makes THOMAS even more similar to a public library model.

The high usage of THOMAS from all parts of the United States, as well as many foreign countries, provides the basis for an invaluable testbed for studying how people use a network-based information system. A wide variety of people have accessed THOMAS, including experienced searchers at academic institutions and libraries, high school teachers, lobbyists, congressional staffs, and thousands of individual citizens interested in the process of government.

A full evaluation of the techniques that are being deployed in THOMAS will require recall/precision comparisons based on sample query sets and relevance judgments [8]. Although we intend to carry out this evaluation, at this early stage of the development of the system, new techniques are being introduced primarily on the basis of a small number of test queries and user feedback about specific problems. In this paper, we focus on these initial experiences with THOMAS, including descriptions of usage and query statistics and how query processing has changed to reflect user feedback. We also describe how current research on query expansion and morphological processing could provide techniques that will improve THOMAS and similar systems.

In the next section, we describe the THOMAS database. We then outline the basic features of the INQUERY retrieval system and their importance in the THOMAS system. Following sections give usage statistics for THOMAS, including query length and frequency, describe the query processing that is used to improve the retrieval performance for the type of queries and data in this system, and present some current problems and possible solutions.

THE DATABASE

THOMAS is evolving toward a complete database of legislative information. At this time, it contains the full text of all legislation introduced in the 103rd and 104th Congresses, as well as the text of the Congressional Records for those congresses.

The legislative database consists of all bills introduced in the House and Senate. These bills can exist in as many as 10 versions as they pass through the legislative process. Examples of versions are: “Introduced in the House", “Introduced in the Senate", “Engrossed in the House", “Enrolled Bill Sent to President", etc. Each version is treated as a separate document. In the 103rd Congress, there were about 13,000 separate documents comprising almost 300MB of data. Bills can range in size from 1K to 2MB.

The Congressional Record is published on a daily basis when at least one chamber is in session. Each record consists of a Daily Digest, which is a summary of the day’s events, a House Section, a Senate Section, and an Extension of Remarks section, as appropriate. The Extension of Remarks section contains statements by members which were not actually made on the floor of Congress, but were inserted into the Record. The House and Senate sections are divided into debates or discussions on specific subjects, each with its own title, such as “Balanced Budget Amendment". These subdivisions form the basic documents in the database. As with the bills, there is a large variance in size of the basic documents, from about 1K to 700K. The Congressional Record for the 103rd Congress contains about 30,000 of these documents, for a total size of about 600 MB.

All legislative information is received via FTP from the Government Printing Office on a daily basis. Both the bills and the Record are passed through a preprocessing program which establishes document and title tags for INQUERY, converts GPO publication codes to HTML, and creates tables of contents where appropriate. In order to avoid unnecessary workload for WWW browsers with low bandwidth connections, a maximum segment size of 10K has been established. Tables of contents are created for bills and Record segments which exceed 10K. These tables of contents are returned to the searcher when the full document display is requested.

Additional navigational aids are created for the Congressional Record. A hypertext table of contents is created for each day’s Senate, House, and Extension of Remarks sections. The Daily Digest is converted to HTML and provided with page references linked to the data sections of the day’s Record. An overall calendar is created to reference the four sections of the Record for any given day of the year.

Once the preprocessing is complete, INQUERY is invoked to index the data. Bills are indexed by title, bill number, and the text of the bill. The Congressional Record is indexed by title, document identifier, date, speaker, and page number. Indexing using INQUERY requires approximately 1 hour and 15 minutes to process the 600MB Record for the 103rd Congress on an RS6000 Model 990. Due to the speed of indexing, the entire database is indexed on a daily basis, with no attempt at updating. In order to provide uninterrupted service, two copies of each database are maintained, and the indexing is performed off-line. When the indexing is complete, a current production pointer is switched to the updated database. This also provides immediate backup if the production run should fail. INQUERY also provides an incremental update capability to avoid re-indexing, and this can be done in parallel with queries using concurrency control. In the first version of THOMAS, however, these features were not fully available and were judged to be not necessary.

THE INQUERY SYSTEM

The INQUERY retrieval engine used in THOMAS is based on a probabilistic model of retrieval using a Bayesian net framework [11, 12]. The system has been used in a variety of research projects and applications, and it has been consistently very effective in the government-sponsored TREC evaluations [3]. The following list gives a brief overview of the major features of INQUERY and indicates their use in THOMAS:

- **Ranked output** - INQUERY computes the probability that a document is relevant to a query by combining evidence in the text of the document and the corpus as a whole. The probability value is then used to rank the documents for presentation to the user. The effectiveness of this process has been demonstrated by recall/precision evaluations in TREC and other settings (e.g. [7]), and there are significant usability advantages for ranked output.
- **Passage-based retrieval** - The probability of relevance of a document is computed based both on the entire content of a document and the best matching passage in the document [1]. This improves the retrieval effectiveness and provides a means of viewing large documents. This feature is incorporated in THOMAS as a means of avoiding sequential browsing of very large bills.
- **Ability to handle both simple and complex queries** - The INQUERY query language provides a range of operators which are used to specify how evidence in the document text

should be combined to estimate relevance. This means that INQUERY's probabilistic framework can be used for simple word-based queries, Boolean queries, phrase-based queries or any combination. Examples of operators include averaging and weighted averaging of evidence (#SUM and #WSUM), probabilistic Boolean (#AND, #OR, #NOT), strict Boolean that preserve probabilities (#BAND, #BANDNOT), proximity and phrase (#n, #UWn, #PHRASE), synonym (#SYN), and passages (#PARSUMn). The #n proximity operator allows up to n words between the words in the argument, and insists they occur in the order given. The #UWn (unordered window) operator specifies that the words in the argument must occur in a text window of size n, but they can occur in any order. The #WSUM operator allows relative importance weights to be associated with different components of the query. The query processing in THOMAS uses the passage, proximity, Boolean AND (#BAND), and weighted sum operators to improve the effectiveness of retrieval.

- **Field-based retrieval** - INQUERY also includes fields for indexing and retrieval. The field extensions are part of the probabilistic query evaluation, which means that field-based searches can be combined with complex queries to produce rankings. THOMAS uses fields for some fixed attributes such as bill number and type.
- **Flexible and efficient indexing** - The indexing process is designed to be easily extensible in order to incorporate a variety of document structures (e.g. HTML, MARC, etc.), different morphological processing techniques (e.g. stemming techniques [5]), and domain-specific concept recognizers (e.g. marking occurrences of people, companies, locations, and dates). The recognizer capability is not currently used in THOMAS because we could not predict which particular concepts would be valuable in this application.
- **Tools for query processing and query expansion** - Natural language queries can be transformed into INQUERY queries using tools such as a part-of-speech tagger and a stop structure recognizer. Queries can also be automatically expanded using related phrases from the corpus [4]. These tools are not currently used in THOMAS, but in the section on current problems we describe how they could contribute.
- **Support for relevance feedback and routing** - User feedback on the relevance of retrieved documents can automatically modify either a query in a typical interactive session or a profile in a routing environment where incoming documents are compared to stored profiles [8]. These techniques are not currently used in THOMAS because it is not designed to support routing and relevance feedback can sometimes have unpredictable results when used with databases containing long documents (such as the bills). As these techniques are improved for full-text databases, we would expect to use them in THOMAS.

A common gateway interface to WWW servers has been built using the INQUERY API. This is available either as a client-server or standalone version. THOMAS uses the latter configuration.

USAGE

The immediate response to the announcement of THOMAS was overwhelming. On the first day following the joint press conference given by the Speaker of the House and the Librarian of Congress, the server logged 75,000 transactions. Since then, usage has leveled out to about 40,000 transactions per work day. A transaction on the World Wide Web can range from a database search to the downloading of a single image. A single visit to the THOMAS home page counts as 2 transactions, while viewing a search page counts as 3 transactions (the text, plus 2 GIF images).

A better way to measure serious usage of the system is by examining searches. In the period between January 6 and March 20, 1995, there have been 2,302,589 WWW transactions, which encompassed 294,575 accesses to the THOMAS home page. Of these home page accesses, there were 196,724 accesses to query pages. From the access log, we can determine that there were 94,911 queries where at least one item was examined.

An examination of the text of these queries provides valuable information on user behavior when presented with the opportunity to enter free-form natural language queries. Many of the same queries are repeated many times. Table 1 shows some of the more popular queries with a count of the number of times they were entered. Of the 94,911 separate queries recorded, only 25,321 were unique.

The data recorded from THOMAS indicates that users tend to enter very simple queries. Table 2 shows the number of search terms (including stopwords) recorded in the 25,321 unique searches logged. The fact that 88 percent of all queries contain 3 or fewer words suggests that most queries in this application consist of a single concept expressed as a word or phrase. An examination of the searches most frequently submitted, shown in Table 1, tends to confirm this hypothesis. Although a number of studies have been done on the types of queries submitted to information services [9], there is not a large amount of data on what happens in systems with free-form or "natural language" queries. Examples of natural language searching of legal material at West Publishing [10] suggest that, in that environment, queries tend to be somewhat longer than those seen in THOMAS. Some of the common test collections used in information retrieval research, together with the average number of words in the natural language queries, are Cranfield (9.2), CACM (13), Time (8.9), NPL (7.1), INSPEC (15.6), and West (9.6). An additional factor that tends to constrain query length in real environments is that many computer users have been trained by the text search capabilities in many systems to believe that longer queries will fail to retrieve any documents.

QUERY PROCESSING IN THOMAS

As described before, INQUERY incorporates a powerful query language which supports user-assigned weights, proximity and Boolean operations, passages, and field restrictions. Complex queries can be submitted without substantially affecting response time. This aspect of INQUERY allows the system designer considerable freedom to experiment with query processing techniques that convert a natural language query into a structured INQUERY query.

The first problem noted in THOMAS was that searchers looking for bills with specific titles would not find those bills ranked near the top of the search results. For exam-

Query	Count
balanced budget	2,600
crime	1,057
gun(s)	994
balanced budget amendment	991
s 314	902
telecommunications	888
welfare	846
budget	753
abortion	678
line item veto	610
gun control	539
unfunded mandates	532
welfare reform	513
education	441
tax	415
term limits	401
crime bill	375
contract with America	366
public broadcasting	333
decency	333
immigration	316
balanced	315
health care	305
baseball	303
firearms	300
TOTAL	16,106

Table 1: 25 most common queries in THOMAS

Words	Unique Queries
1	5,767
2	9,646
3	6,905
4	2,240
5	656
6	87
7	19
8	1
Total	25,321

Table 2: Number of words in queries

ple, searchers looking for the "Defense Appropriations Act" would not find the bill in the first 20 items in the retrieved list. Many bills have the words "defense", "appropriations", and "act" in great profusion, while the bill itself often does not include its own name or subject matter within the text. This is a phenomenon common to many documents, where the actual subject matter of the document is assumed by the author and is not mentioned in the text.

After some experimentation, it was found that weighting words (using #WSUM) occurring in the title of the bill by a factor of 20 over words in the text produced good results for the queries that had been pointed out by users to have this problem. As mentioned earlier, query processing techniques should be evaluated using recall/precision experiments, but these have not yet been done.

The next factor which appeared to have an important influence on the success or failure of searches was word proximity. Since it was evident that the vast majority of searches consist of a single phrase, it would seem reasonable to give additional credit to documents that contain the query words in close proximity. For example, a document containing "state department" would be more likely to be relevant than one containing the words "state" and "department" in completely different sections.

Adding increased weight to the occurrence of the search terms in ordered proximity to each other resulted in a considerable improvement in the relevancy of bills returned. Since the sizes of documents in the database varied to such a large degree, very large documents with highly significant passages would be ranked lower than very small documents with widely-scattered search terms. After considerable experimentation, a weight of 90 (using #WSUM with a base weight of 1) was assigned to any occurrence of all of the search terms occurring in an ordered proximity of 3 words from each other.

A further improvement was obtained by adding weight to the unordered occurrence of all search terms within a given window of a specified size within the document. This was accomplished using the INQUERY #UWn. The value of n was arbitrarily set to 10 times the number of search terms. The weight of any occurrence of the terms within the specified unordered window was set to 50% of the weight given to terms found in ordered proximity to each other.

Some users complained that even with the above weighting, documents containing all of their search terms would be ranked below documents containing only a subset of their search terms. This is a problem with the method used to weight terms when documents show a great variance in size and queries are short and simple. The term weighting in INQUERY is a variation of that used in other systems such as SMART [8], and is known as *tf.idf* weighting. The *tf* (term frequency) component of this weight depends on the within-document frequency of the term, and the *idf* (inverse document frequency) component varies inversely with the frequency of the term in the corpus.

When queries such as "pressler public broadcasting" are entered, the proximity rules set forth above are of little use. The

```
#WSUM ( 1.0 balanced 1.0 budget 1.0 amendment
90.0 #3(balanced budget amendment)
45.0 #UW30(balanced budget amendment)
90.0 #BAND(balanced budget amendment)
20.0 #FIELD(TITLE #WSUM(1.0 1.0 balanced
    1.0 budget 1.0 amendment
    20.0 #3(balanced budget amendment)
    10.0 #UW30(balanced budget amendment)
    1.0 #BAND(balanced budget amendment) )
10.0 #PARSUM200(balanced budget amendment) )
```

Figure 1: Transformed query for "balanced budget amendment"

searcher is particularly interested in Senator Pressler and public broadcasting, but the high frequency of the words “public" and “broadcasting" cause the relevant documents to occur far down on the list behind documents that contain many occurrences of “pressler". This problem has been rectified to some extent by adding a high weight to the occurrence of the Boolean AND of the search terms in any document. The INQUERY #BAND operator was used with a weight of 90 to achieve this effect. This problem with the ranking algorithm indicates that further research is needed on methods of combining evidence that emphasize the number of matching terms more strongly.

An example of the current weighted search algorithm is shown in Figure 1. Note that the entire query is repeated inside the #FIELD operator, where it is restricted to the Bill title. The #PARSUM or passage operator is also used as part of the standard INQUERY query processing.

The effectiveness of the above query processing techniques is based upon the assumption that (1) queries are not of an arbitrary length, but rather, rarely exceed 4 terms, and (2) that most queries contain a single concept or phrase. This is a pragmatic approach to query processing which results in a significant improvement in the ranking of relevant documents in the vast majority of cases, but makes no difference in a small number of cases. If an experienced searcher enters a complex search statement, these techniques will have little impact. Of course, these queries are precisely the ones which perform well without further processing.

CURRENT PROBLEMS

In January 1995, the Boston Globe wrote a negative article about THOMAS, claiming that the search techniques employed were faulty [6]. The writer entered the search “elderly black Americans" into the system and received a bill on “black bears" as most relevant, followed by bills relating to “black colleges and universities". This illustrates a common problem in explaining relevance ranking to the user. Since there were no bills in any way related to “elderly black Americans", the system returned the closest results possible - bills with a large number of occurrences of “black" and “American".

In order to correct the misunderstanding of the process by the user, THOMAS now generates several informative messages following a search. The following header is printed:

```
IMPORTANT: Read the following before examining
  the list of bills:
```

followed by one of the following messages:

```
  The following words were not found
    in the database: word1, ..., wordn.

  All of your search terms did not occur
    in any single bill.

  All of your search terms did not occur
    within 50 words of each other in any one bill.

  The phrase "[search terms]" did not occur
    in any bill.
```

In the case of the Boston Globe search, the second message would have been printed.

The search for “elderly black Americans" points out another problem - the need for some kind of thesaurus substitution. The concept of “elderly" could occur as “older" or “aged", while “black Americans" could occur as “African Americans", and be related to “minorities". We are currently investigating the possibility of integrating a pre-existing thesaurus into THOMAS, or making use of an automatic association thesaurus such as INQUERY’s PhraseFinder [4]. When a version of PhraseFinder built using newspaper databases was used with this query, it found the following phrases that were automatically added to the query:

```
retired persons,  poverty line,  poverty rate,
elders, health statistics
```

Making that change to the query improved the retrieval performance so that 9 of the top 10 bills were related to the elderly and poor. The potential confusion of the name “Elders” with “elderly” is related to the issue discussed next.

The use of automatic stemming in THOMAS has both positive and negative aspects. Stemming provides a powerful method for linking different word forms into a single concept cluster. However, it can cause problems when the exact form of a word is required. Since INQUERY indexes only word stems, a search for “Representative Franks" will also return references to two other representatives named “Frank”. A search for “Billy" will return references not only to everyone named “Bill”, but to all references to “bills". This came up, for example, when someone wanted to find references to Senator Byrd’s dog Billy in the Congressional Record.

This problem can be rectified to some extent by substituting the new stemming algorithm KSTEM developed at the University of Massachusetts for the standard Porter Stemmer supplied with INQUERY [5]. KSTEM is a more conservative, dictionary-oriented system that also permits the database designer to set up exception lists of words which are not to be stemmed. This would solve the two problems above, but requires the designer to anticipate all stemming problems which may occur. A better solution might be to store exact word forms in the database, and carry out stemming at query time. For example, the query-based stemmer could decide that a

query word “bill" should be expanded to #SYN(bill bills), but this could be overruled by the user specifying that only the form “bill" is acceptable. Research is now in progress to add such a feature to INQUERY [2].

CONCLUSION

The searching and query processing customizations described above are derived from trial and error experience, rather than being based on the traditional information retrieval model of experiments based on test collections of queries and relevance judgments. Analysts on the THOMAS project are continually testing sample user queries to determine the effectiveness of the techniques employed. The results have been determined to be generally successful based upon user feedback.

Further experiments based on formal relevance judgments are planned to determine the level of effectiveness improvement from the new techniques and to rank performance against similar systems.

Overall, our experience with the THOMAS system shows that it is very important to tune an information system to the user population. The query processing algorithms used in the current system would probably not be appropriate for expert searchers. An advanced retrieval engine such as INQUERY provides the query processing and indexing flexibility to accommodate rapid tuning while retaining efficient performance. The lessons learned from the large variety of users in THOMAS emphasize different issues than can be studied in a formal experimental environment. This has motivated more research in query processing, term weighting, relevance feedback and stemming algorithms.

ACKNOWLEDGMENTS

This work was supported in part by the Center for Intelligent Information Retrieval at the University of Massachusetts.

REFERENCES

1. J. P. Callan. Passage-level evidence in document retrieval. In *Proceedings of ACM SIGIR International Conference on Research and Development in Information Retrieval*, pages 302–310, 1994.

2. W. B. Croft and J. Xu. Corpus-specific stemming using word form co-occurrence. In *Fourth Annual Symposium on Document Analysis and Information Retrieval*, pages 147–159, 1995.

3. D. Harman. Overview of the Third Text REtrieval Conference (TREC-3). In D. Harman, editor, *Proceedings of the Third Text REtrieval Conference (TREC-3)*, pages 1–20. NIST Special Publication 500-225, 1995.

4. Y. Jing and W.B. Croft. An association thesaurus for information retrieval. In *Proceedings of RIAO 94*, pages 146–160, 1994.

5. Robert Krovetz. Viewing morphology as an inference process. In *Proceedings of the 16th International Conference on Research and Development in Information Retrieval*, pages 191–202, 1993.

6. M. Putzel. Room for doubting Thomas. *Boston Globe*, page 92, January 27 1995.

7. T. B. Rajashekar and W.B. Croft. Combining automatic and manual index representations in probabilistic retrieval. *Journal of the American Society for Information Science*, 46(4):272–283, 1995.

8. Gerard Salton and Michael J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.

9. Karen Sparck Jones, editor. *Information Retrieval Experiment*. Butterworth, 1981.

10. Howard Turtle. Natural language vs. Boolean query evaluation: A comparison of retrieval performance. In *Proceedings ACM SIGIR 94*, pages 212–220, 1994.

11. H.R. Turtle and W.B. Croft. Evaluation of an inference network-based retrieval model. *ACM Transactions on Information Systems*, 9(3):187–222, 1991.

12. H.R. Turtle and W.B. Croft. A comparison of text retrieval models. *Computer Journal*, 35(3):279–290, 1992.