N94-32441

HIGH-SPEED DATA SEARCH

James Driscoll
Department of Computer Science
University of Central Florida
Orlando, Florida 32816, USA

ABSTRACT

The high-speed data search system developed for KSC incorporates existing and emerging information retrieval technology to help a user intelligently and rapidly locate information found in large textual databases. This technology includes: natural language input; statistical ranking of retrieved information; an artificial intelligence concept called semantics, where "surface level" knowledge found in text is used to improve the ranking of retrieved information; and relevance feedback, where user judgments about viewed information are used to automatically modify the search for further information. Semantics and relevance feedback are features of the system which are not available commercially. The system further demonstrates a focus on paragraphs of information to decide relevance; and it can be used (without modification) to intelligently search all kinds of document collections, such as collections of legal documents, medical documents, news stories, patents, and so forth. The purpose of this paper is to demonstrate the usefulness of statistical ranking, our semantic improvement, and relevance feedback.

INTRODUCTION

Locating information using large amounts of natural language documents (text) is an important problem. Examples at KSC are searching press releases and numerous other documents to quickly answer media questions, accessing bulky manuals and schematics compactly stored on a CD via a laptop computer, and retrieving digital images by means of their catalog descriptions.

The primary intent of our work has been to provide convenient access to information contained in the numerous and large public information documents maintained by Public Affairs at NASA Kennedy Space Center (KSC). The documents maintained by Public Affairs at NASA KSC consist of press releases, and other printed information created at KSC, and other NASA offices using various wordprocessors. There are also documents from outside contractors, such as Rockwell, which produces the "NASA National Space Transportation System Reference" more often called the "shuttle manual." During a launch at KSC, about a dozen NASA employees access these printed documents to answer media questions. The planned document storage for NASA KSC Public Affairs is around 300,000 pages (approximately 900 megabytes of disk storage).

Current commercial text retrieval systems focus on the use of keywords to search for information. These systems typically use a Boolean combination of keywords supplied by the user to retrieve documents. In general, the retrieved documents are not ranked in any order of importance, so every retrieved document must be examined by the user. This is a serious shortcoming when large collections of documents are searched.

The QA system is a high-speed data search system developed jointly by NASA KSC, the University of Central Florida, and Florida High Technology and Industry Council. It is a statistically based text retrieval system which ranks retrieved documents according to their statistical similarity to a user's request. Statistically based systems provide many advantages over traditional Boolean retrieval methods, especially for users of such systems, mainly because they allow natural language input. These systems have been a research success for over twenty years [9]. However, the transfer of this retrieval technique into large operational systems has been very slow because, until recently, there was no evidence that statistical ranking could be done in real-time on large document collections [4]. There are only three commercial systems in the United States which allow natural language input and perform statistical ranking of retrieved information [2].

The QA System incorporates two other features which are not available in any commercial text retrieval system, but have been shown to dramatically improve the statistical ranking of retrieved information. The first is an artificial intelligence concept called semantics, where "surface level" knowledge found in text is used to improve the ranking of retrieved information. The second is relevance feedback, where user judgments concerning viewed information are used to automatically modify the search for more information.

The QA System is very close to being a commercial product. It has been used to participate in a (first) Text Retrieval Conference (TREC-1) managed by the National Institute of Standards and Technology (NIST). Our participation in TREC-1 was funded by the Defense Advanced Research Projects Agency (DARPA). Participation in TREC-1 has enabled the QA System to be tested in an environment other than answering questions, and applied to databases other than aerospace text collections [3].

Conventional information retrieval using statistical ranking is demonstrated first in this paper. Demonstrations of improved statistical ranking due to the use of semantics within the QA System are then presented for comparison. This is followed by a demonstration of relevancy feedback within the QA System. In all demonstrations, the focus on paragraphs of information for retrieval will be evident. Finally, the issues of platforms and high-speed for the QA System are discussed in the Conclusion.

CONVENTIONAL INFORMATION RETRIEVAL

Finding relevant text and ranking the retrieved documents is not new and there are commercial systems which already perform this activity; we mention here an example of ranked, relevant text retrieval. For a demonstration to NASA KSC, the 1000 page shuttle manual was used by considering each paragraph of the manual as a document. This resulted in a collection of 5143 documents. A commercial hypertext IR system called SPIRIT [11] was used to automatically index the collection and provide natural language access. SPIRIT is a mainframe system. Running on an IBM 4381, SPIRIT required three and one-half hours of clock time to index the collection of 5143 documents.

Figure 1 is a screen generated by SPIRIT for asking the natural language query

What are the dimensions of the cargo area in the shuttle?

Figure 2 is a screen generated by SPIRIT revealing a ranked list of 245 relevant documents with CLASS 1 being the most relevant. Figure 3 is a screen generated by SPIRIT revealing the first document in CLASS 6, which contains the answer to the query. This paragraph was found by reading the single paragraph in CLASS 1 first, then the single paragraph in CLASS 2, and so on until the answer was read in the tenth paragraph.

NATURAL LANGUAGE QUERY ON THE SHUTTLE BASE

<1>: What are the dimensions of the cargo area in the shuttle?

EMPTY WORDS: What, are, the, of, the, in, the.

KEYWORDS: dimensions, cargo, area, shuttle.

***

Figure 1. Natural Language Query to the SPIRIT System.

CLASSES	NB DOCS	KEYWORDS
1	1	dimensions, cargo, shuttle.
2	1	cargo, area, shuttle.
3	1	dimensions, area.
4	2	dimensions, shuttle.
5	4	cargo, area.
6	30	cargo, shuttle.
7	12	area, shuttle.
8	7	dimensions.
9	40	cargo.
10	147	area.
BOTTOM OF LIST		

Figure 2. Document Classes Generated by the SPIRIT System.

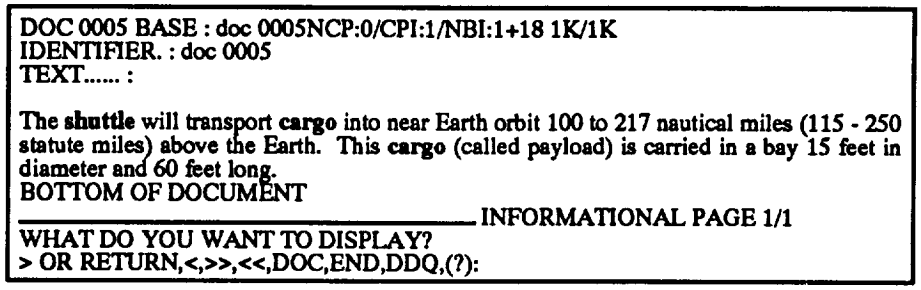

Figure 3. Document Display by the SPIRIT System.

Note that performance in this Question/Answer environment is measured by counting how many documents were examined to find the document containing the answer. This is not the usual way of measuring the performance of IR systems, but it is very appropriate for a Question/Answer environment.

The underlying principles and algorithms of automated IR systems like SPIRIT are well-known. Terms used as document identifiers are keywords modified by various techniques such as stop lists (removal of useless or empty words), stemming, synonyms, and query reformulation. Here, we present basic concepts associated with the calculation of weighting factors.

The calculation of the weighting factor (w) for a term in a document is a combination of term frequency (tf), document frequency (df), and inverse document frequency (idf). The basic term definitions are as follows:

tf_{ij} = number of occurrences of term T_j in document D_i

df_j = number of documents in a collection which contain T_j

$idf_j = \log\left(\frac{N}{df_j}\right)$, where N = total number of documents

$w_{ij} = tf_{ij} \cdot idf_j$.

When an IR system is used to query a collection of documents with t terms, the system computes a vector Q equal to $(w_{q1}, w_{q2}, \ldots, w_{qt})$ as the weights for each term in the query. The retrieval of a document with vector D_i equal to $(d_{i1}, d_{i2}, \ldots, d_{it})$ representing the weights of each term in the document is based on the value of a similarity measure between the query vector and the document vector. A common similarity function which normalizes the the similarity coefficient in case of different document sizes is the following:

$$sim(Q, D_i) = \frac{\Sigma_{j=1}^{t} w_{qj} \cdot d_{ij}}{\sqrt{\Sigma_{j=1}^{t} d_{ij}^2}} \quad (1)$$

It is important to note that the calculation of a similarity coefficient for each document and the ranking of the documents relevant to a query is rather time consuming. This is due to the summations that occur in the above formula and the fact that every document that has a term in common with a given query must be considered. The main problem with text retrieval using statistical ranking has been the time required to produce the document ranking given a query. Consequently, query response time has been typically slow.

SEMANTIC APPROACH

Although the basic statistical ranking approach (as demonstrated by SPIRIT) has shown some success in regard to natural language queries, it ignores some valuable information. We now know that these systems can be further improved by imposing a semantic data model upon the "surface level" knowledge found in text.

Semantic Modeling

Semantic modeling was an object of considerable database research in the late 1970's and early 1980's [1]. Essentially, the semantic modeling approach identified concepts useful in talking informally about the real world. These concepts included the two notions of entities (objects in the real world) and relationships among entities (actions in the real world). Both entities and relationships have properties.

The properties of entities are often called attributes. There are basic or surface level attributes for entities in the real world. Examples of surface level entity attributes are Size, Color, and Position. These properties are prevalent in natural language. For example, consider the phrase "large, black book on the table," which indicates the Size, Color, and Position of a book.

In linguistic research, the basic properties of relationships are discussed and called thematic roles. Thematic roles are also referred to in the literature as participant roles, semantic roles, and case roles. Examples of thematic roles are Beneficiary and Time. Thematic roles are prevalent in natural language, they reveal how sentence phrases and clauses are semantically related to the verbs in a sentence. For example, consider the phrase "purchased for Mary on Wednesday" which indicates who benefited from a purchase (Beneficiary) and when a purchase occurred (Time).

Consider the following query:

How long does the payload crew go through training before a launch?

The basic statistical approach dismisses the following words in the query as empty: "how", "does", "the", "through", "before", and "a". Some of these words contain valuable semantic information. The following list indicates some of the thematic roles triggered by a few of the words in the above query:

long ⇒ Duration, Time
through ⇒ Location/Space, Motion With Reference To Direction, Time
before ⇒ Location/Space, Time

As another example, consider the query in Figure 1:

What are the dimensions of the cargo area in the shuttle?

The keyword "dimensions" indicates the attribute General Dimensions and the keyword "area" indicates both the thematic role Location/Space and the attribute General Dimensions. It would be reasonable to expect that the document that answers this query would have words in it that fall in the category of General Dimensions.

The primary goal of the QA System has been to detect thematic and attribute information contained in natural language queries and documents. When the information is present, the system uses it to help find the most relevant paragraph to a query. In order to use this additional information, the basic underlying concept of text relevance was modified. The major modifications include the addition of a lexicon with thematic and attribute information, and a modified computation of the similarity measure given in (1).

The Semantic Lexicon

The QA System uses a thesaurus as a source of semantic categories (thematic and attribute information). For example, Roget's Thesaurus contains a hierarchy of word classes to relate word senses [5]. For our research, we have selected several classes from this hierarchy to be used for semantic categories. We have defined thirty-six semantic categories as shown in Figure 4.

In order to explain the assignment of semantic categories to a given term using Roget's Thesaurus, consider the brief index quotation for the term "vapor":

vapor		
n.	fog	404.2
	fume	401
	illusion	519.1
	spirit	4.3
	steam	328.10
	thing imagined	535.3
v.	be bombastic	601.6
	bluster	911.3
	boast	910.6
	exhale	310.23
	talk nonsense	547.5

Thematic Role Categories
Accompaniment
Amount
Beneficiary
Cause
Condition
Comparison
Conveyance
Degree
Destination
Duration
Goal
Instrument
Location/Space
Manner
Means
Purpose
Range
Result
Source
Time

Attribute Categories
Color
External and Internal Dimensions
Form
Gender
General Dimensions
Linear Dimensions
Motion Conjoined with Force
Motion in General
Motion with Reference to Direction
Order
Physical Properties
Position
State
Temperature
Use
Variation

Figure 4. Thirty-Six Semantic Categories.

The eleven different meanings of the term "vapor" are given in terms of a numerical category. We have developed a mapping of the numerical categories in Roget's Thesaurus to the thematic role and attribute categories given in Figure 4. In this example, "fog" and "fume" correspond to the attribute State; "steam" maps to the attribute Temperature; and "exhale" is a trigger for the attribute Motion with Reference to Direction. The remaining seven meanings associated with "vapor" do not trigger any thematic roles or attributes. Since there are eleven meanings associated with "vapor," we indicate in the lexicon a probability of 1/11 each time a category is triggered. Hence, a probability of 2/11 is assigned to State, 1/11 to Temperature, and 1/11 to Motion with Reference to Direction. This technique of calculating probabilities is being used as a simple alternative to a corpus analysis. It should be pointed out that we are still experimenting with other ways of calculating probabilities.

Extended Computation of the Similarity Measure

The probabilistic details of a semantic lexicon and the computation of semantic weights can be found in [13]. A detailed explanation of the manner in which the QA System combines semantic weights and keyword weights can be found in [12].

Essentially we treat semantic categories like indexing terms, and the probabilities introduced by a semantic lexicon mean that the frequency of a category in a document becomes an expected frequency and the presence of a category in a document becomes a probability for the category being present. This means that the document frequency for a category becomes an expected document frequency, and this enables an inverse document frequency to be calculated for a category.

So the computation of a similarity coefficient as shown in (1) can be used, but now the summations in the formulas include semantic categories in the documents as well as terms in the documents. In other words,

$$sim(Q,D_i) = \frac{\sum_{j=1}^{t} w_{qj} \cdot d_{ij} + T \sum_{j=t+1}^{t+s} w_{qj} \cdot d_{ij}}{\sqrt{\sum_{j=1}^{t} d_{ij}^2 + B \sum_{j=t+1}^{t+s} d_{ij}^2}} \qquad (2)$$

where $s = 36$ is the number of semantic categories, and T and B are scaling factors for adjusting the blend.

SEMANTIC IMPROVEMENT

The QA System has demonstrated a noticeable semantic improvement using the similarity function in (2). Consider the same document collection and natural language query shown in the commercial system example of Figures 1, 2, and 3. Using the commercial system SPIRIT, ten paragraphs were read in order to find the answer to the following query:

What are the dimensions of the cargo area in the shuttle?

Considering the QA System, Figure 5 is a screen generated for asking this same natural language query. Figure 6 is a screen generated by the QA System graphically showing to the user the importance of the keywords found in the query. Figure 7 is a screen generated by the QA System graphically showing to the user the importance of semantic information found in the query. Notice the "importance" of the semantic category General Dimensions in the screen shown in Figure 7. This long bar means that the semantic category General Dimensions is present in the query and there are very few documents retrieved (using keywords) having this type of semantic content. Hence, the importance of the category.

Finally, Figure 8 is a screen generated by the QA System revealing the second paragraph found by proceeding through the ranked list of documents retrieved by the QA System for this query. The semantic information found in the query and displayed in Figure 7 is the reason the QA System ranked the answering paragraph second instead of tenth as did the SPIRIT system. Notice that the answering document in Figure 8 has several words in it which trigger the semantic category General Dimensions. We have lots of data like this and several technical papers which reveal a significant performance improvement due to semantic modeling in the NASA KSC Question/Answer environment.

For another example of semantic improvement, consider the shuttle manual and the query:

How fast does the orbiter travel on orbit?

This query is interesting for two reasons. One is that the words "orbiter" and "orbit" are rather frequent words in the shuttle manual so lots of paragraphs are retrieved. The other reason is that the word "fast" is used for reference to velocity or speed.

Figure 9 shows the number of paragraphs one must read to find a particular answering paragraph to this query for both a small and large collection of documents. In the small collection, the word "fast" does not occur at all and for the large collection, the word "fast" never occurs in an answering paragraph. Consequently, keyword only statistical ranking is never very good. But by using semantics, the word fast causes a similarity to paragraphs using the words velocity or speed. Consequently, semantics improves the statistical ranking of an answering paragraph. Different blends of keywords and semantics are shown using the similarity function in (2).

RELEVANCE FEEDBACK

It has been pointed out that conventional IR systems have a limited recall [6]; only a few relevant documents are retrieved in response to user queries if the search process is based solely on the initial query. This indicates a need to modify (or reformulate) the initial query in order to improve performance. It is customary to search the relevant documents iteratively as a sequence of partial search operations. The results of earlier searches can be used as feedback information to improve the results of later searches. One possible way to do this is to ask the user to make a relevance decision on a certain number of retrieved documents. Then this relevance information can be used to construct an improved query formulation and recalculate the similarities between documents and query in order to re-rank them. This process is known as relevance feedback [7,8,9,10] and it has been shown experimentally to improve the performance of the retrieval system.

The basic assumption behind relevance feedback is that, for a given query, documents relevant to it should resemble each other in a sense that they have reasonably similar keyword vectors. This implies that if a retrieved document is identified as relevant, then the initial query can be modified to increase its similarity to such a relevant document. As a result of this reformulation, it is expected that more of the relevant documents and fewer of the nonrelevant documents will be extracted.

The automatic construction of an improved query is actually straightforward, but it does increase the complexity of the user interface and the use of the retrieval system, and it can slow down query response time. Essentially, the terms and semantic categories for documents viewed as relevant to a query can be used to modify the weights of terms and semantic categories in the original query. A modification can also be made using documents viewed as not relevant to a query. Experimental results show a very promising improvement for relevance feedback within the QA System.

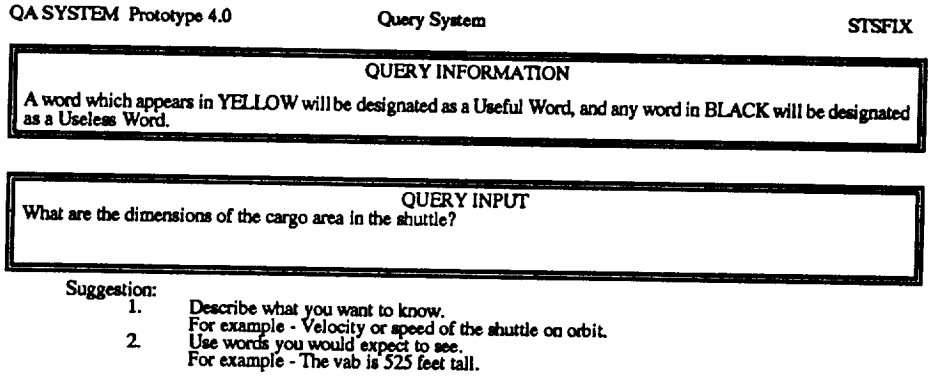

Figure 5. Natural Language Query to the QA System.

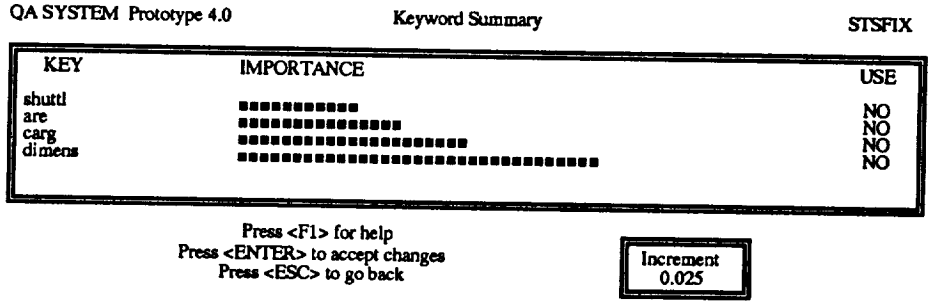

Figure 6. Keyword Summary by the QA System.

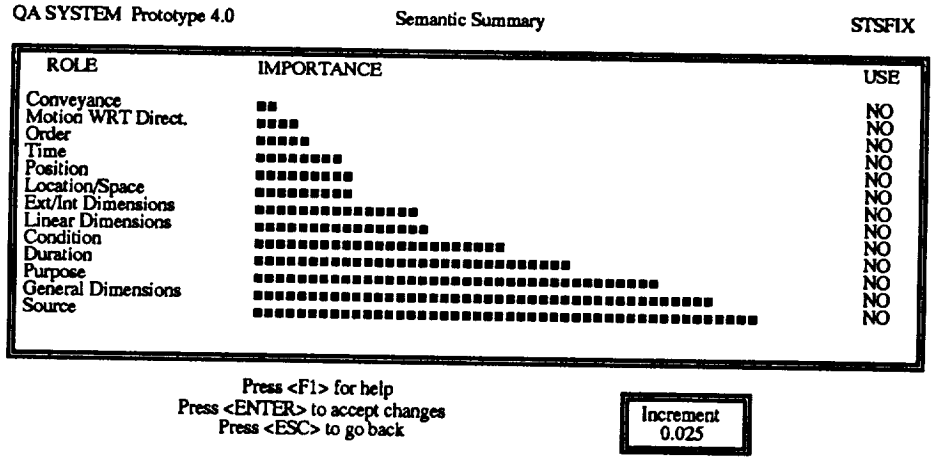

Figure 7. Semantic Summary by the QA System.

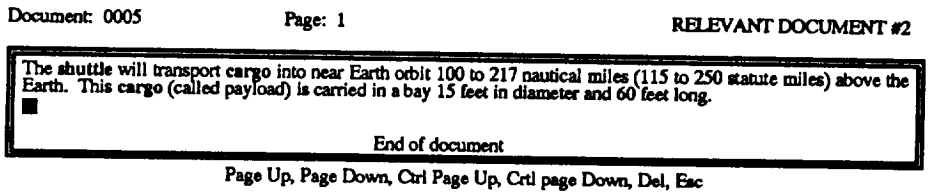

Figure 8. Document Display by the QA System.

	Keywords Only	$T-B=1.10206$ Blend of Keywords and Semantics	$T-B=8.0$ Blend of Keywords and Semantics
First 26 pages of the shuttle manual (160 documents)	19	4	2
The entire shuttle manual (5143 documents)	145	126	14

Figure 9. Number of paragraphs read to find a particular answering paragraph for: How fast does the orbiter travel on orbit?

Figure 10 provides an example using the first 26 pages of the shuttle manual and the query:

How fast does the orbiter travel on orbit?

Recall from Figure 9 that 19 paragraphs were read to find an answering paragraph. The document identifiers for these 19 paragraphs are shown in the left column of Figure 11 along with the notes that Document #13 and Document #16 were considered relevant to the original query, and Document #14 answered the query. All the other viewed documents were not relevant to the query.

If relevance feedback is selected within the QA System and the system is told to display two documents and then reformulate the query, then the documents shown in the right column are viewed. Each document viewed must be tagged as relevant or not-relevant. Document #14 shows up earlier in the statistical ranking primarily because Document #13 was tagged as relevant to the original query.

It is interesting to note that if one tags Document #14 (which answers the query) as relevant, then Document #87 is retrieved and it almost exactly answers the query. Document #87 would never be retrieved using just keywords without feedback because it has no keywords in common with the original query. Documents 13, 14, 16, 69 and 87 are shown in Figure 11. The keywords that these documents have in common with the original query are underlined. Clearly, Document 69 is not relevant to the original query.

CONCLUSION: PLATFORMS AND THE ISSUE OF HIGH SPEED

Originally, the QA System was restricted to an IBM compatible PC platform running under the DOS operating system and without the use of any other licensed commercial software such as a DOS extender. The QA System is implemented in Borland C and one version uses B+ tree structures for the inverted files. We felt the speed of the system and its storage overhead was not efficient so a hashing scheme was added to eliminate the use of B+ trees and provide codes for keywords. We expected this second version to have improved indexing time, storage, and retrieval speed.

Experiments revealed that indexing time of the QA System did not improve much. We were not surprised because the QA System is restricted under the PC DOS platform. This platform has a serious memory addressing restriction which results in memory page swapping and this seriously affects the speed of processing, especially during creation of the hashing table and index structures. The improvement in storage, however, was very impressive. It is very much matched to our objective which is to make our storage ratio of indexes to text, around 0.5. This is comparable to the ratio of very efficient, retrieval systems using statistical ranking.

Addressing the high speed issue, we now have the Borland C compiler for OS/2 so we expect to have a very high speed QA System running under OS/2 very soon. We are also in the process of converting the QA System to run in the UNIX environment. Figure 12 reveals achieved and projected run-time performances of the QA System on different operating system platforms. The DOS, B+ tree version of the system is shown in the upper left corner. Below (diagonally) are shown the OS/2, UNIX B+ tree and hashing versions of the QA System for different amounts of RAM. Indexing and typical query response times are shown for both a small (2.4 megabyte) and a large (1.2 gigabyte) document collection. Data for this chart was obtained in part from experiments performed for TREC-1 [3].

160 Documents

Answer can be found in Document 14, 87

Keywording			Relevance Feedback (view 2)		
1	69		1	69	
2	13	Relevant	2	13	Yes
3	82		3	82	
4	15		4	107	
5	123		5	85	
6	106		6	124	
7	85		7	16	Yes
8	124		8	14	Yes, Answer
9	21		9	87	Yes, Answer
10	23				
11	24				
12	83				
13	31				
14	26				
15	16	Relevant			
16	84				
17	11				
18	12				
19	14	Answer			

:

never get 87 (no query words in 87)

Figure 10. Relevance Feedback Improvement for the Query: How fast does the orbiter travel on orbit?

Document 13

The two orbital maneuvering system engines are used to place the orbiter on orbit, for major velocity maneuvers on orbit and to slow the orbiter for re-entry, called the deorbit maneuver. Normally, two orbital maneuvering system engine thrusting sequences are used to place the orbiter on orbit, and only one thrusting sequence is used for deorbit.

Document 14

The orbiter's velocity on orbit is approximately 25,405 feet per second. The deorbit maneuver decreases this velocity approximately 300 feet per second for re-entry.

Document 16

For deorbit, the orbiter is rotated tailfirst in the direction of the velocity by the primary reaction control system engines. Then the orbital maneuvering system engines are used to decrease the orbiter's velocity.

Document 69

- Atlantis (OV-104), after a two-masted ketch operated for the Woods Hole Oceanographic Institute from 1930-1966, which traveled more than half a million miles in ocean research.

Document 87

Entry interface is considered to occur at 400,000 feet altitude approximately 4,400 nautical miles (5,063 statute miles) from the landing site and at approximately 25,000 feet per second velocity.

Figure 11. Documents 13, 14, 16, 69, and 87. Keywords in common with the original query are underlined.

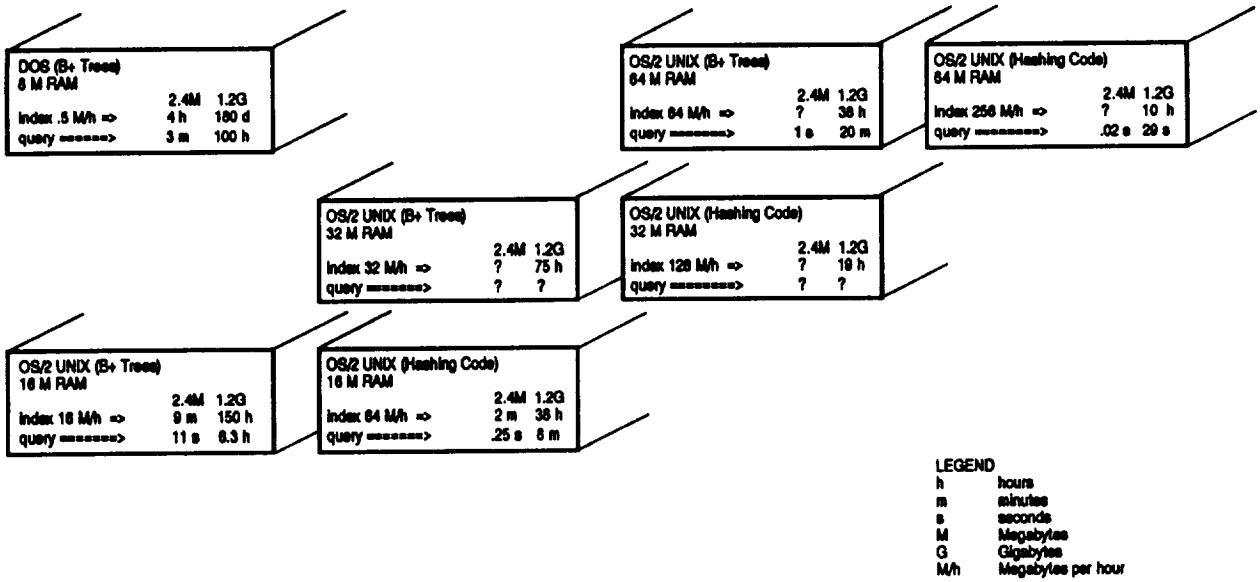

Figure 12. Run-Time Performance of the QA System.

References

[1] C. Date, *An Introduction to Database Systems*, Vol. I, Addison Wesley, 1990.

[2] Delphi Consulting Group, 1991, *Text Retrieval Systems: A Market and Technology Assessment*, 266 Beacon Street, Boston, MA, 1991.

[3] J. Driscoll, J. Lautenschlager and M. Zhao, "The QA System," *Proc. of the First Text Retrieval Conference (TREC-1), NIST Special Publication 500-207* (D. K. Harman, editor), March, 1993.

[4] D. Harman and G. Candela, "Retrieving Records from a Gigabyte of Text on a Minicomputer Using Statistical Ranking," *JASIS*, Vol. 41, pp. 581-589. 1990.

[5] *Roget's International Thesaurus*, Harper & Row, New York, Fourth Edition, 1977.

[6] G. Salton, *Automatic Information Organization and Retrieval*, McGraw-Hill, 1968.

[7] G. Salton, *The Smart Retrieval System—Experiments in Automatic Document Processing*, 1971.

[8] G. Salton, E. A. Fox, and E. Voorhees, "Advanced Feedback Methods in Information Retrieval," *JASIS*, Vol. 36, pp. 200-210, 1985.

[9] G. Salton, *Automatic Text Processing*, Addison-Wesley, Reading, MA, 1989.

[10] G. Salton and C. Buckley, "Improving Retrieval Performance by Relevance Feedback," *JASIS*, Vol. 41, pp. 288-297, 1990.

[11] *SPIRIT Version 2.1 User's Manual*, SYSTEX Company, Ferme Du Moulon, 91190 Gif Sur Yvette, France (French Edition), May 1986.

[12] D. Voss and J. Driscoll, "Text Retrieval Using a Comprehensive Semantic Lexicon," *Proceedings of ISMM First International Conference on Information and Knowledge Management (CIKM-92)*, Baltimore, MD, November 1992.

[13] E. Wendlandt and J. Driscoll, "Incorporating a Semantic Analysis into a Document Retrieval Strategy," *Proceedings of the Fourteenth Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, Chicago, IL, pp. 270-279, October 1991.