Text Retrieval - A trendy cocktail to address the Dataworld

Dr. Mohan Chellappa
Shankar Kambhampaty
Kanishka Systems, Singapore

Abstract

Speed, Accuracy, Portability, Multilingual Support are issues which play a key role in the assessment of the value of Text retrieval engine. Multimedia search capabilities is fast becoming a necessity as the data has become composite in recent times. Indexing speed and storage overhead, once the big issues in text retrieval systems, are becoming less and less important as hardware is getting better and cheaper day by day. Globalization of data has brought into emphasis how necessary multilingual support would be in text retrieval systems. End user availability of large amount of data has also made it necessary for navigation tools to be endowed with certain amount of intuitive intelligence to minimize the time spent in reaching data which is relevant to that person. The availability of data to assess the user pattern of search material would benefit the data providers in planning their presentation of data and methodology and grouping of data collection.

This paper analyzes these issues in the light of a free text search engine developed with the Institute of Systems Science, Singapore.

1.0 Introduction

The real challenge of information retrieval is the indeterminacy, complexity and variety of users' needs, and the correct approach to developing indexing and searching techniques is to relate these firmly to the properties of users[Belkin and Vickery, 1985; Sparck Jones, 1992].

User applications for document imaging and multimedia incorporating the text retrieval engines have to be more versatile than traditional systems for libraries.

Some imaging applications require that sections of text in image files be recognized by optical character recognition(OCR) and retrieved based on user supplied criteria. The amount of text that requires to be indexed and retrieved is usually not as voluminous as in traditional library applications. However, the speed of retrieval is crucial to the success of these applications.

The relevance of a retrieval also holds the key to several performance issues in multimedia systems. For instance, if the documents contain plenty of graphics and the documents are not ranked in a suitable manner, the user will have to spend considerable time browsing through the documents before finding the document of his or her interest.

The global flow of data across the information networks will eventually result in the smooth flow of text in multiple languages with applications performing transliteration and translation for information exchange with users in different parts of the world. An engine adaptable to the requirements of different languages will greatly enhance the flexibility of the applications incorporating it.

A Free Text Search Engine was designed taking into account the above mentioned considerations for speed and accuracy of retrieval and the flexibility to provide multilingual support wherever necessary. This paper details the engine developed with the Institute of Systems Science, Singapore and discusses several issues and results of the implementation.

2.0 Architecture of Free Text Search Engine (FTSE)

The design goal for the FTSE was to develop an engine that is optimized on retrieval time and storage overhead. The system was required to

(i) Retrieve a ranked set of documents from a document collection of 5000 documents when queried with 50 query words in less than 5 seconds.

(ii) have a storage overhead of less than 40% of the data indexed.

0730-3157/94 $04.00 © 1994 IEEE

With the hardware becoming more powerful and less expensive, indexing speed becomes less important when compared to retrieval time and storage overhead.

The many studies done in the past showed, in particular, both that performance for quite different techniques used in indexing and retrieval, when seriously applied, was much the same, and thus that simple techniques were very competitive with more sophisticated ones and that absolute performance is not high[Sparck Jones, 1981].

It became quite clear[Chan et al, 1985] that there was a need to look much more carefully at conventional indexing in all its variety and in all its aspects - philosophy, implementation, index language design and indexing description principles.

The engine was therefore designed to cater to a provide various degrees of precision based on applications need. The nature of index information stored is dependent on the precision required.

2.1 For systems that incorporate relevance feedback or for systems that do not require to rank documents based on the "proximity" of query words to each other, documents are ranked based on document weights calculated using inverse document frequency(idf) and the frequency of occurrence of each of the query words in the documents retrieved.

The indexing module of the engine stores the frequency of occurrence of each of the words. The inverse document frequency, idf is calculated on the fly.

2.2 For systems that require a high degree of precision, the engine stores information related to position of each word in the indexed document. The text in the document is divided into "logical" paragraphs. Paragraphs are made up of sentences. The occurrence of each of the word in the sentences of the paragraph is represented by a bitmap structure in which a bit corresponding to the sentence is set in which the word occurrs. During retrieval, the engine computes the weights based on the "proximity" of the query words in the retrieved documents.

2.3 Certain well-proven techniques to reduce the index size and improve the quality of retrieval are employed some of which are given below:

2.3.1 The individual text words are recognized for languages like English in which letters for words and word boundaries are clearly defined by white space. For other languages the unicode equivalent of the text is used to determine the "keywords" that are indexed.

2.3.2 Certain common words(such as "and", "or" in English) are eliminated by consulting a list of "stop words"[Salton & Buckley 1992].

2.3.3 For languages such as English, the words are reduced to word-stem form before indexing. As studies have shown that "weak" stemming improves the performance, a modified version of Porter's algorithm has been used for stemming words.

3.0 Implementation

The Free Text Search Engine(FTSE) contains modules that implement four processes:

a) Database Access Process
b) Indexing Process
c) Retrieval Process
d) Document Database Maintenance Process

These processes are executed when the User Application makes the corresponding API calls.

The process diagram of FTSE is shown in figure 1.

3.1 Database Access Process:

This process(P1.1) controls access to other processes and database files. When the user application requests a database open, the process opens the necessary files and creates a task which remains active until a database close is requested by the application.

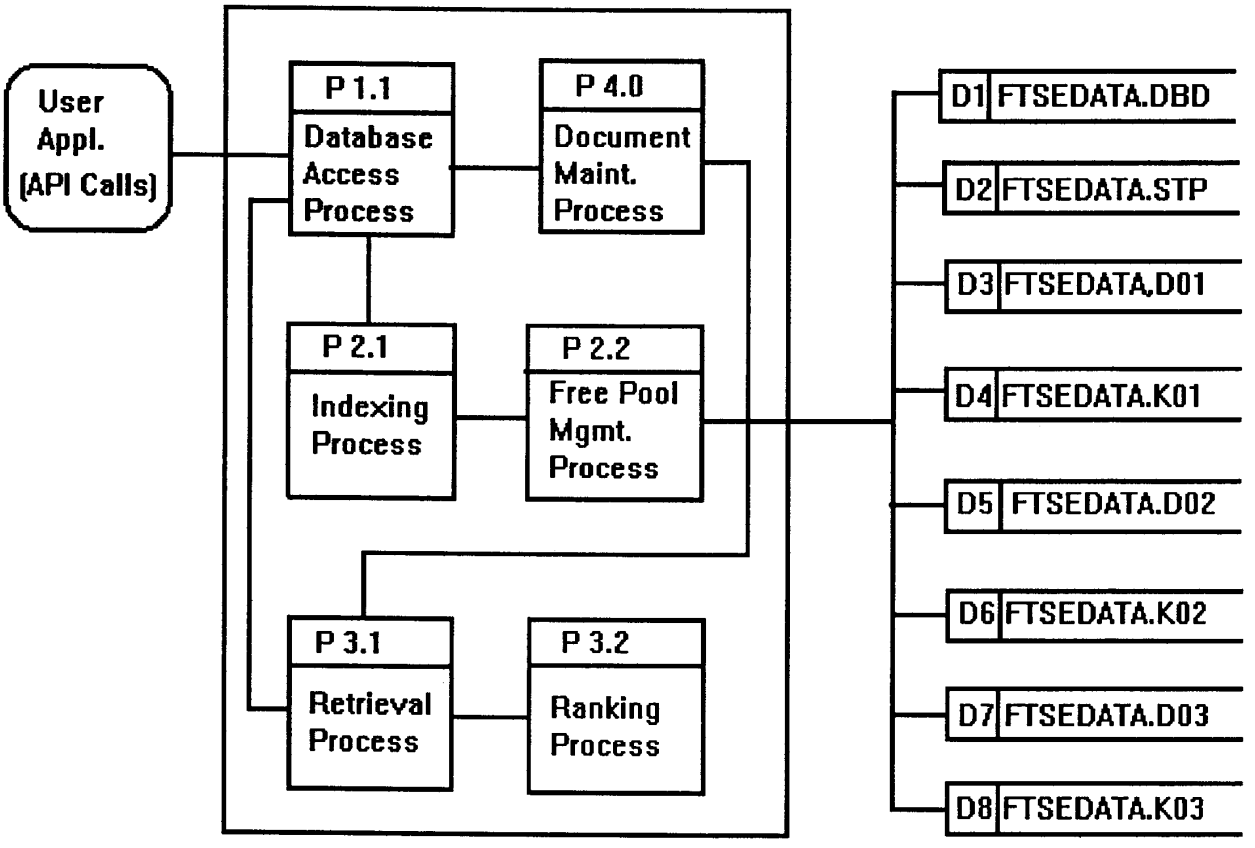

Figure 1

For every call executed by the user application, this process performs checks to ensure that the user application is permitted to do so and generates error codes, if necessary.

3.2 Indexing Process

When the user application requires a document to be indexed, the Indexing Process extracts the text from the document and indexes the words. For each word in the document, the process checks to ensure that it is not in the stoplist. An inverted list of all the words in the document is prepared with the index information of each of these words and their relative importance values. The information is then compressed and stored in the document database.

3.3 Free Pool Management Process

In order to achieve improved indexing speed and minimal storage space, the indexing process activates a process that implements a strategy called Free Pool Management. Index information for each word in a document is prepared as a "chunk" and stored in the database. When these words occur again in other documents that get indexed subsequently, a fresh "chunk" is prepared and the original chunk released to a free pool so that it may be reused to store index information of other words. This process ensures minimal disk accesses and optimal utilization of disk space.

3.4 Retrieval Process

When the user application provides a set a query words for retrieving documents, the retrieval process orders the words based on their relative importance. The inverted list of words stored is consulted to extract the index information for each of the query words.

3.5 Ranking Process

The retrieval process activates the Ranking process to rank documents. The ranking process uses the index information to compute document weights that form the basis for ranking.

The relative "usefulness" of documents that contain the supplied query terms is dependent on a number of a factors. A great deal of effort was made in determining the combination of factors that yields the most desirable ranked list.

3.6 Document Database Maintenance Process:

When the user application wishes a document index to be cleared, the index words for each of the words in the document is identified. The storage space is released to Free Pool for reuse. If a word occurs only in the document that is being deleted, it is removed from the inverted list.

D1-D8 Database Files:

These Database Files store the stopword list, the inverted index list, and the index information in a compressed form.

4.0 Results:

It has been found that use of a general purpose database tool for storing the index information results in poor performance. A fast network model database was initially used to store the index information. TIPSTER data was used to benchmark the engine. On account of several overheads involved in general purpose databases, the results obtained were far below expectations.

On profiling the application it was found that about 68% of the indexing and retrieval time was spent in several housekeeping activities of the database tool(For example, updating the date and time stamp on the records).

The results obtained on performing tests on a Sun Sparc 1 are in Figure 2 below:

No. of Documents in document collection:	5000
No. of Query terms:	50
Retrieval time(in seconds):	**12**
Storage Overhead(% of data indexed):	**80**
Indexing speed (in MB/hr):	**2**

Figure 2

When the general purpose database was replaced by low-level data management routines considerable improvement was seen in all parameters as can be seen in Figure 3 below:

No. of Documents in document collection:	5000
No. of Query terms:	50
Retrieval time(in seconds):	**3**
Storage Overhead(% of data indexed):	**36**
Indexing speed (in MB/hr):	**20**

Figure 3

5.0 Conclusions:

The text retrieval engines developed for document imaging and multimedia applications have crucial requirements in speed, accuracy and multilingual support. With huge amounts of data being accessible by home users via data networks it becomes necessary that text retrieval engines be versatile and flexible to cater to the requirements of fast and intelligent retrieval.

Bibliography:

[Belkin and Vickery, 1985] N. J. Belkin and A. Vickery. Interaction in information systems. Library and information Research Report, 35, 1985. The British Library, London.

[Chan et al, 1985] L. M. Chan, P. A. Richmond, and E. Svenonius, editors.Theory of Subject Analysis: A Sourcebook. Libraries Unlimited, Littleton, CO 1985.

[Salton & Buckley, 1992] Salton. G. & Buckley C, Automatic text structuring experiments, 1992

[Sparck Jones, 1992] Sparck Jones K. Assumptions and issues in text-based retrieval, 1992

[Sparck Jones, 1981] Sparck Jones K. Information Retrieval Experiment. Butterworths, London, 1981.

Fagan, J.L.(1989) The effectiveness of a non-syntactic approach to automatic phrase indexing for document retrieval, Journal of the American Society for Information Science, Vol. 40.

Jacobs, Paul S., (1992) Text-based intelligent systems: current research and practice in information retrieval and extraction. Hillsdale, NJ, Lawrence Erlbaum Associates.

Maron, M.E. & Kuhns, J.L.(1960) On relevance, probabilistic indexing and information retrieval. Journal of ACM, vol. 7: 216-244.

Smeaton, A.F., (1990) Natural language processing and information retrieval, Information Processing and Management, vol. 26.