X. Design Considerations for Time Shared Automatic Documentation Centers

M. E. Lesk

1. Introduction

Consideration is being given to the design of documentation centers as part of projected general revisions of the scientific information dissemination system. Simultaneously, the changeover in computing equipment at all large data processing installations is stimulating the revision of automatic information retrieval systems. New equipment, such as time-sharing systems and large-scale random-access data-storage devices are now available. The use of fully automatic procedures in documentation centers thus becomes practical.

Experimental investigations into algorithms for automatic content analysis have shown that such methods are competitive with classical (i.e. manual) indexing methods. Although much analysis remains to be done, the total ignorance of automatic methods which prevailed a few years ago is now decreasing.

Despite the progress in fully automatic information retrieval, the designers of large-scale documentation centers still think in terms of systems which operate with manual subject indexers. Because of the current interest in large documentation systems, it is necessary for the proponents of fully automatic systems to formulate practical proposals for computerized analysis projects.

The SMART project, after several years of study with experimental retrieval programs, must therefore consider the extension of its procedures to operational situations and make future plans. The following discussion is intended to be of general interest, although its specific proposals refer to the computing equipment (IBM SYSTEM 360/67) to be available at Cornell and/or Harvard Universities in the near future.

2. Principles

The first question that must be raised about any information retrieval system is that of _purpose_. For what users is the system to be designed, and what service is to be offered to them ? The SMART system has been an exclusively experimental system, whose user population is a small group of researchers at the Harvard Computation Center and whose function is to evaluate retrieval algorithms. It should become a system which provides practical information retrieval services for a user group of working researchers.

This does not imply that no further work is to be done on small-scale experimental systems. There is much to be learned about retrieval procedures through detailed analysis of small examples; new methods and systems will always be proposed that can be tried out economically on small subcollections. However, work on large collections must also begin.

Operating systems are subject to various difficulties that must not be faced in experimental, small-scale systems. Experimental systems are used to investigate the details of retrieval methods in intensively analyzed document collections. This work remains valid, and in fact cannot be done in large-scale systems. However, the problems of _user interaction_, of _time and presentation_ problems, of _economics of use_, of _system maintenance_, of collection selection and updating, occur only in real-life systems. Knowledge of these problems as they affect fully automatic documentation systems are almost nil. They are analogous to problems that must be faced in large nonautomated documentation systems, but there exist also many differences. For example, in many nonautomated systems, the user presents his query to an intermediary who translates it into some index language which is used by the system. In fully automatic systems however, this step is avoided, since the user queries the system directly using the natural language. The relationship of the user to the direct, fully automatic system, is thus clearly different and must be studied before the best user service techniques can be discovered.

Such problems have not been studied before because an operating fully automatic documentation center is needed to study them. Over the last few years, methods have however become available for automatic content analysis; these methods have been carefully studied on small collections, with known performance capabilities, and can now be applied to larger collections in operational environments.

The best reason for studying the design of fully automatic information retrieval centers at the present time is, however, the fact that small systems have finally demonstrated that some mechanized analysis techniques perform more effectively than many manual analysis techniques; it may therefore be expected that fully automatic systems would probably offer better service to real users than present-day manual or semi-manual systems. It may also be hoped that the construction of a real,operating, fully automatic system would dispel the general skepticism about the usefulness of automatic information retrieval.

The new system, then, should be designed with the aim of providing a literature-searching service for scientists. It can also provide a means for the experimental investigation of documentation systems, in the environment of a functioning information retrieval center.

Once the goal of designing an operating documentation center is established, many design principles of small-scale systems must be abandoned. The collection being searched must be large enough to be of interest to someone. This will mean at least 50,000 documents and probably 250,000. The response must be adequately fast to satisfy the users. This will mean the abandonment of the present batch-processing arrangement in use at most computer centers. People who are faced with a one-day delay when using an information retrieval center are likely to avoid it, since it might then be more convenient to go to the library directly in order to perform a rudimentary search. Furthermore, the advantages gained by a batching of requests (the ability to perform many searches at once) are maximized at approximately one core full of requests, or 100-1000 requests. This is a load considerably larger than can be expected to be needed initially in one day in an information system. One might then just as well plan for the processing of each request individually. Accumulating requests for a period of one hour, as a possible compromise, would probably not be sufficient to accumulate enough requests to gain real efficiencies in processing, and probably would antagonize many users. The goal, then, must be a time-sharing mode of operation in which each request is processed individually and in which an effort is made to provide an answer in a matter of seconds to a user who remains at his console during the search.

3. Methods

We can draw on the experience obtained by using the SMART project to select the processing methods which should be used in the planned system. The results of the SMART project on the relative values of various retrieval methods are developed elsewhere,[1] and only a brief summary of some of the relevant points is given here.

For input purposes, the best compromise between economy of space and quantity of information is probably the document abstract. Since most scientific journals require author abstracts, it should not be difficult to obtain a set of abstracts for the document collection being searched.

The search procedure should be based on the use of a thesaurus with phrases. In past experiments with the SMART system, this method has been found to offer the best performance of any method tested on most collections. This method exhibits the additional advantages of simplicity and flexibility. Specialized thesauri can be constructed for individual needs. Isolated errors are easily corrected. Extensions of different languages and adaptations to different subject areas are possible. On the other hand, statistical procedures for automatic synonym detection, are relatively fixed procedures for which adjustments are more difficult to make. It is not clear how such methods can be extended to different languages. Finally, automatic synonym detection is found experimentally to produce results inferior to those obtained by proper thesauri. Hierarchies also produce inferior results.

Based on past SMART experience, we accept as our basic content analysis procedure a thesaurus lookup, and a loose phrase lookup of the type studied there. The entire document collection is passed through this lookup at the beginning of the project; requests and new documents are looked up as they arrive. When new documents are added, they are also studied to see if the dictionaries are still reasonably up-to-date.

Experience shows, and common sense would indicate, that provision for repeated or iterated searches is necessary. Any system is certain to exhibit some peculiarities, derived from the document collection if from nothing else, and requests must be adjusted to them. This has already been noted in the experimental system.

The following may represent a possible outline of the steps involved in processing a request.

1) The user approaches a console, probably located in a library, that is attached to a large time-shared computer. He identifies himself and provides whatever accounting information is required by the system operators, and is then placed in contact with the retrieval programs.

2) The user types his request in natural English, observing only a few rules designed to insure clarity. Basically, any statement of his need which would be clear to a librarian should suffice for the computer.

3) His request is looked up in a thesaurus and its associated phrase dictionary, and the adjusted request image is prepared for searching.

4) The request is now matched against documents from the reference collection, using an efficient but accurate search strategy, and those documents relevant to the request are selected by the computer.

5) Within a time interval measured in seconds, an answer is provided. Depending on the hardware available, this might be a document accession number, a bibliographic citation, a citation plus the document title, a citation plus the document abstract, a microfilm reel and frame number for the library microfilm room, or a picture

on an attached microfilm reader of the document itself. Further answers would be provided in similar forms.

6) The user now evaluates his output. Depending on the form in which the documents were presented, this may require him to leave the console and search for a book or journal in the library stacks. If this is necessary, the computer search should be designed with high precision in mind. If the user can obtain the document directly from an attached or nearby microfilm reader/printer, somewhat less emphasis need be placed on the precision of the methods used.

7) The user may be satisfied with the results of this search, in which case he leaves the console and studies the document(s) produced. He may be extremely dissatisfied, and as a result he may leave the console in disgust. This may be considered to be a failure of the system, and the search strategy should be designed to avoid it. Probably, he will be partially satisfied, and will rephrase his request and try again. This partial satisfaction may arise from one of three causes:

    a) The computer has retrieved documents that are not relevant to the question as stated.

    b) The computer has retrieved documents that are relevant to the question as stated, but because of ambiguities or misphrasings, the retrieved documents are not relevant to what the user actually intended as his request.

    c) The computer has retrieved documents that are relevant to the question as intended and stated, but after consideration of these documents the user has decided that he asked for the wrong topic.

The request may be rephrased by the user, or by the computer, using feedback data from the user. In case a), the problem is basically a flaw in the computer system, which may be compensated by artificially constructing a request which can be answered correctly. The computer is a better judge of its own flaws and requirements than a user who has not seen the system before. It may be expected, then, that in case a) iterative retrieval based upon feedback will be superior to rephrasing by any but the most experienced users. In cases b) and c), however, where parts of the question as stated are actually misleading and where more is required than simply adjusting the style of writing of the user, feedback may be expected to be less useful. In case c) clearly a complete rewriting of the question is called for. The computer might nevertheless assist in this rewriting by displaying, upon request, portions of the thesauri and/or hierarchy in the region being studied.

The above outline should be common to virtually all requests submitted to the retrieval system. Some additional features might be needed frequently. For example, a method of arbitrarily marking certain documents to prevent their retrieval will be necessary. The most obvious examples of this need arise in the case of a multi-lingual document collection where users will wish to specify the languages that they can read, or in the case of a partially classified collection in which users without the appropriate security clearance must be denied access to certain documents. Such an exclusion feature can also be used positively, however. Users might wish to request only current documents, for example. Furthermore, someone who is totally unfamiliar with a field might wish to begin by requesting only review articles and books. There may well be a demand by students for textbooks or other more elementary treatments of a subject.

Another specialized facility that would often be used is a set of special-purpose dictionaries. In fields with rapidly changing vocabulary, for example, there special dictionaries might be constructed for out-of-date material. Dictionaries could also be constructed for different classes of users; for example, circuit designers would probably want the electrical section of the thesaurus arranged in a manner different from that desired by the manufacturers of electrical equipment. Foreign-language processing would also have to be approached by constructing separate dictionaries.

The computer must also be provided with a set of error-detecting mechanisms. Users could then be notified of any words used that are not in the dictionary, or of any questions asked for which no relevant material appears in the collection. Users should also be warned if a request could produce an excessive amount of output, so that they could either rephrase the request by making it more specific, or restrict the output to summary and review papers only.

It may be expected that the documentation system described here would be of great interest to an active group of researchers if it were properly implemented and convenient to use, and its services would be well used.

4. Practicalities

Some more detailed design questions can now be considered with specific reference to the SMART project. The first question that arises concerns the subject area which should be used for the proposed system. Numerous areas could be suggested, but the following requirements might be used to make a decision:

1) It should be a rapidly developing field, to reduce the amount of back-issuing which is necessary to produce a really useful system.

One could of course develop a current awareness system that would slowly grow into a general retrieval system, but such a system would have few users.

2) It should be a field in which the literature is published primarily in English. Although the foreign language problem can unquestionably be overcome, there is no point in tackling the problem immediately if it is not necessary to do so. A purely English collection will also be easier to use, and this may help in attracting users.

3) It should be a field in which very little of the material carries a security classification, since the initial system will be used initially for experiments in documentation systems, and it will thus be desirable to be able to publish the results of these experiments. A system which does not include classified documents will also be easier to use by the customers.

4) It should be a field with a good abstracting service and with strong American scientific societies, which might cooperate with the system designers.

5) It should be a field with at least one hundred active researchers at the institution where the documentation center is established, to insure an adequate user population.

6) It should be a field for which the conventional library has an adequate and easily accessible shelf collection. Nothing would be more frustrating than to have the computer select a document which is only available through a slow inter-library loan procedure.

A field that satisfies most of these criteria is physics, concentrating on the journal articles only, and avoiding much of solid state physics. Physics papers are published primarily in English and Russian, and the major Russian journals are available in English translation. Material becomes out-of-date extremely quickly. As a basic science, it is largely unclassified (avoiding, of course, applied nuclear physics). One predominant publishing group, AIP (the American Institute of Physics) exists, and it is currently interested in documentation problems. Many other groups, such as Euratom, NASA, and the AEC are also interested in any documentation efforts in physics. A file of 25,000 articles a year (about 25-50 journals of 500-1000 articles per year) kept up for ten years should be very attractive to many users. Certain difficulties would arise with physics, of course: there exists a large technical report literature which should be included, but which is largely unabstracted and inaccessible. Also, much strange and inconvenient symbolism is used in writing papers. But these problems are not insuperable, and physics could thus easily serve as the basic collection.

We may then assume that the basic collection would contain about 250,000 100-word abstracts, or a total of $2.5\text{x}10^7$ English words. This represents a total data input of about 10^9 bits and will require about ten to twenty reels of magnetic tape to store. It may be expected to contain on the order of 10^5 different English words, and the most frequently occurring few thousand words will likely include 90% of the total number of word occurrences.

This fact can be used in the construction of an efficient dictionary lookup. When the SMART programs are loaded into memory, as part of the user sign-in procedure, the programs will be accompanied by a short dictionary of 1000 or 2000 words. The user requests will probably be fairly short, about 25 words. They can be looked up in the special high-frequency list in a few milliseconds. Perhaps a few words will remain which were not included in this special list. Based on the first few letters of the word, a computation of its approximate position in the backup dictionary is made, and the appropriate section of the complete dictionary is brought in from disk or data cell. If the disk is used for the dictionary, it will take about 0.075 sec. for access plus a few milliseconds for the search. If the data cell is used, perhaps 0.5 seconds per reference will be necessary. The lookup procedure should be completed in one second or less.

In practice, if the full dictionary is referred to very frequently, it may be desirable to increase the size of the special dictionary to 5000 words or more (this should be possible with a 360/67 computer). If the dictionary is stored on disks this should not be necessary, since ten disk references could be made before the wait becomes excessive. It is unlikely that more than ten references per request would be required, since most requests will contain about ten significant words. Access to the data cell, on the other hand, is slower, and a larger internal dictionary might be needed if a data cell were used for the dictionary storage.

The present SMART system for dictionary storage and search would probably be useable in the new system. Some economies in dictionary storage, however, would probably be made in an effort to save memory space. The total dictionary size should be of the order of 10^7 bits. Note that if the entire dictionary has to be read into memory, the lookup process would take at least ten seconds, even at high read rate of 10^6 bits per second.

If more than one request is submitted at the same time, it might be possible to save time if these requests were to require the same sections of the expanded dictionary to complete the lookup. It is debatable, however, whether the time saved in this way is worth the programming effort involved, considering the improbability of two users happening to submit requests simultaneously.

Once the dictionary lookup is completed, the computer must quickly notify the user of any words used which were not included in the dictionary. Such words should also be saved for investigation by the system programmers. Users should not be permitted to enter words in the dictionaries themselves, since a large dictionary consists of a very complicated structure and changes made in one part are likely to affect other parts in ways unforeseen by the casual user.

The computer must now compare the request against the document collection. The collection has also been looked up previously, and the documents in coded form are presumably stored on (say) the data cell. Each concept detected will require perhaps 15 bits for the concept number and 10 bits for the weight, or a total of 25 bits. A full document representation will consist of perhaps 50 of these, plus identification and other data, so that 2000 bits per document should be adequate for storage purposes. For 250,000 documents this amounts to $5x10^8$ bits or approximately $1\frac{1}{2}$ data cell sections. A full data cell consists of about $3x10^9$ bits; a full disk about $1.5x10^9$ bits. Thus the storage problems are well within the range of practicality.

Clearly, the whole collection cannot be read by the system for each request if real-time answers are to be provided. For $5x10^8$ bits, a reading rate of 10^6 bps would require over eight minutes. The basic search plan would then have to involve a partitioning of the document collection, a comparison with representative "key" vectors to decide which partitions to search, and then a thorough search of the selected sections. These "key" vectors might be the centroid vectors of the document sets created by the partitioning. Documents might be included in more than one cluster. This would however, be slightly wasteful of storage space.

To derive timing estimates, let us assume disjoint document clusters, so that we have approximately n key vectors representing n clusters of 250,000/n documents each. We should expect that about two or three clusters are searched per request, but for safety's sake, let us assume that five clusters are searched per request. If we assume that five correlations can be performed per millisecond, the total time required for internal operations is

$$n/5 + 250{,}000/n \text{ msec or about } 250/n \text{ seconds.}$$

The time required for external operations is the data cell access time ($\frac{1}{2}$ second) and read time for each cluster. Since each cluster contains 250,000/n documents and each document consists of 2000 bits, processed at $7\text{x}10^5$ bits per second, each cluster will require about 750/n seconds to read in. If five clusters are to be read, and one expects an average of two in each data cell, the total read time would be 2(0.5+750.n). Reasonable values for n would thus be n = 500 or n = 1000 which would allow the complete search to be performed in 3 to 5 seconds. Considering the small amount of additional work (sorting the correlations and applying the cutoff and other restrictions), it is clear that 10 seconds should suffice for the complete process, and that five seconds would be a more likely bound.

This assumes that no competition exists from other programs in memory for the data cell sections needed by SMART. Since the data cell sections in question will be solely occupied by SMART data, it is highly unlikely that anyone else will want access to them. But the use of multiple requests, or of the multiplexor channel by some other I-O device, might cause difficulty. Since internal operations are fast compared with the read rate of the documents, time-shared programs which do not require I-O devices should not affect the speed of the SMART programs seriously.

The major problem that arises in connection with this timing estimate is that the time-sharing supervisor may slow down the programs excessively. One well-publicized present-day time-sharing system slows down programs by a factor of 60. It is expected that future systems will be more efficient. Other means of saving time are:

1) increased segmentation of the collection;

2) correlation algorithms that begin with the heavily weighted concepts, and do not go on to process the lightly weighted concepts unless the correlation has a chance of being above the cutoff;

3) use of higher-speed input-output equipment (this need not cost more; the system could make a quick, rough, scan of the request during the first phase of the look-up and transfer the clusters that are likely to be needed from the data cell to the disk or drum);

4) type out correlations above cutoff as they occur, rather than waiting until all have been found;

5) clustering the clusters.

Also, various other printouts could be made during the waiting time, (such as time-of-day, number of documents expected to be retrieved by the search) and options could be presented to the user (e.g. number of output documents wanted) to use the waiting time more efficiently.

The timing problem in general does not seem difficult; method 4) in particular, could offer almost instantaneous response even with present-day time-sharing system supervisors.

Once the answers are known, they must be communicated to the user. Probably the best system would be to present a microfilm reel number and frame number, with reference to a reader (preferably reader/printer) located next to the console. For sufficiently large sums of money one could no doubt buy a microfilm reader which could be spaced to the correct frame by the computer, but this would probably cost too much. Also, the microfilm device alone would be valuable and ought to be usable without the computer. The advantage of microfilm output, which can present an image of the actual document to the user immediately, is that users are likely to judge the system largely by the elegance of the output. Permitting immediate reference to the answers without leaving one's chair does sound very attractive.

However, one must consider the possibility that microfilm equipment will not be available, or that some customer is using a console not in the library, and not close to any collection of journals. In this case the system must depend on its own input-output devices, and it will probably be better to store in the data store a list of the titles, authors, and journals of all articles which can be presented. In fact, microfilm readers may be sufficiently slow to make this desirable under any circumstances. About 1000 bits per document should suffice for this data, or $2.5x10^8$ bits. This takes up about one data cell section. Storing the entire abstract is probably too expensive, and would also in all probability be too slow to type out.

No timing problems arise in connection with the production of output, since whatever methods are used, they will be fast relative to the speed of the output device.

Finally, one must consider cost. The biggest single item arises by the fact that currently there exists no way to obtain machine-readable abstracts except by keypunching. Keypunching 250,000 abstracts would cost perhaps $200,000. To program the system, construct the dictionaries, and do other necessary tasks (e.g. abstract any unabstracted articles) would probably be about eight full-time jobs, for two or three years. A total of $150,000 should be sufficient for this purpose. Renting a data cell, type 2321, or as much of it as needed would cost about $35,000. Counting perhaps $100,000 for machine time rental, and money for microfilm terminals (several thousand dollars) one would hope to accomplish the job for less than $500,000 in two or three years, (this is a rough estimate). Lowering the collection size by a factor of 5 would probably lower the costs by only a factor of 2. The best way to save money might be to do the job quickly, obtain cheap computer time, and make use of an optical print reader. Maintenance cost would consist of keypunching cost for 25,000 documents per year ($20,000) and hardware, plus salaries for one or two people ($50,000). It would probably be wise <u>not</u> to expect cash contributions from the users.

5. Conclusions

The construction of a documentation center operating with fully automatic information retrieval techniques seems well within the possibilities of present-day knowledge and technology. It should be undertaken soon.

References

[1] M. E. Lesk and G. Salton, Design Criteria for Automatic Information Systems, Section V, present report.