Dynamic Document Processing

Gerard Salton

TR 72 - 121

Department of Computer Science
Cornell University
Ithaca, New York

Dynamic Document Processing+

Gerard Salton+

Abstract

The current role of computers in automatic document processing is briefly outlined, and some reasons are given why the early promise of library automation and of the mechanization of documentation processes has not been fulfilled.

A new dynamic document environment is then outlined in which clustered files are searched, and information is retrieved following an interactive user-controlled search process. Methods are described for an automatic query modification based on user needs, and for a continuous reorganization of the stored information as a function of earlier file processing and of normal collection growth. The proposed procedures provide powerful tools for information retrieval and for the control of dynamic library collections in which new items are continually added and old ones are retired.

1. The Library Problem

It is convenient to consider four different areas of computer usage in the library: library housekeeping operations, including ordering and receiving of monographs and serials, the circulation control of library items, and the preparation of catalogs and listings of many kinds; cataloging and allied content analysis operations designed to assign to each item subject identifications, as well as call number or

---

+ Dept. of Computer Science, Cornell University, Ithaca, N.Y. 14850.

+ This study was supported in part by the National Science Foundation under grant GJ-314, and in part by the National Library of Medicine, NIH, and under grant LM-00704.

related classification information; storage and retrieval operations which make it possible for the user population to obtain access to the desired information items in response to appropriate requests; and collection control operations designed to insure an orderly collection development through additions and deletions of materials and file changes when necessary.

The large majority of the existing computer applications in the library are of the housekeeping variety. Unfortunately, an examination of these operations from a data processing point of view reveals that their implementation in an automatic environment is likely to produce a considerable number of problems. [1] The files which require processing are quite large, including sometimes many millions of items; the file maintenance and updating operations are extensive, and a large variety of different output products are issued; finally, real-time control of the collection is considered desirable, in the sense that the whereabouts of each item at any given time should be ascertainable.

Obviously, such a set of requirements is certain to impose great strains on the computing facilities, because no application involving large files subject to a great deal of updating is easy, or cheap, to mechanize; and the real-time control by itself requires for implementation a large complement of expensive, fast-access files. In these circumstances, it is not surprising that the application of computers in the library has not proven to be cost-effective for the most part. Nor is it likely, that the plea of many librarians for cheaper and more rational computational processes is likely to be heeded, because costs are not coming down by the two orders of magnitude that are apparently needed to produce a viable mechanized process.*

What is needed first is a change in administrative practices within the library. In particular, the widespread duplication in acquisitions, ordering, circulation control, and catalog production must be abandoned in favor of standardized procedures implemented cooperatively. When each item will be cataloged once only, instead of hundreds of times, and a given mechanized card catalog will prove acceptable for use by many organizations, instead of only by a single one, the existing cost projections will no longer apply, and mechanized library housekeeping operations will have a much better chance of proving cost effective than under present operational conditions.

2. Cataloging and Content Analysis

Unlike the library housekeeping problem, the questions of content description and analysis of written materials have preoccupied scholars in many fields for centuries. At the present time, the prospects for the generation, within the foreseeable future, of a viable system for the automatic content analysis of written texts do not appear to be very bright.

On the one hand, a better understanding of the syntax and semantics of natural languages is said to be necessary to solve the open problems in language processing. On the other hand, the developments over the past few years indicate plainly that the likelihood of achieving substantial and timely progress in computational linguistics is remote: the linguistic

---

* It has been pointed out that the maintenance of a mechanized library catalog would cost about 200 times more than a conventional manual card catalog. [2]

models being designed are becoming increasingly complex, in an effort to mirror precisely the complications inherent in many natural language structures; unhappily, the more involved the models under consideration turn out to be, the less likely it is that they will prove beneficial in a practical system for automatic analysis. The conclusion which often emerges is that "the analysis and synthesis of information, though it may be aided by machines, can only be carried out by skilled human labor". [3]

While the formal situation relating to the content analysis of texts appears unpromising, a great many efforts have been made over the last fifteen years in automatic indexing, and results are available which indicate that practically useful output is obtainable with amazingly little effort. [4] In particular, it has been noted that for a large proportion of the existing technical documents, a correlation exists between the document title and the subject classification manually assigned by human subject experts. Furthermore, about sixty percent agreement is found to exist between the terms manually assigned to documents for content identification, and the content indicators automatically extracted from document texts. Swanson has noted in this connection that the retrieval performance obtainable with all systems now in existence — whether manual or automatic — leaves much to be desired, and he concludes that "though machines may never enjoy more than a partial success in library indexing, ... people are even less promising." [5]

The effectiveness of an automatic term extraction process, using document <u>abstracts</u> as the principal machine input, is compared with a sophisticated, intellectual indexing procedure, performed under controlled conditions by trained subject experts, in the output of Table 1. The recall and precision* values shown in the Table are used to compare the indexing performances of the well-known Medlars system, operating at the National Library of Medicine with the help of carefully trained indexers, and of the automatic SMART retrieval system in which all content analysis operations are carried out automatically. [6] Average performance figures are shown for 29 search requests originally submitted to Medlars, processed against a collection of 450 medical documents; a standard keyword search system — further described in the next section — is assumed to be used for retrieval purposes.

It may be seen from the Table that a deficiency in performance of 30 to 40 percent is produced by the automatic term extraction process incorporated into the SMART system. Furthermore, when an automatic word discrimination list is used by the SMART system to eliminate the so-called "common words" from the indexing process, or a thesaurus which recognizes certain synonyms, the deficiency of the automatic indexing process is still of the order of 20 to 30 percent. [7]

Obviously, refinements going beyond a simple automatic word extraction system must be incorporated into the automatic retrieval environment if performance results competitive with those obtainable conventionally are expected.

---

* Recall is the proportion of relevant material actually retrieved, while precision is the proportion of retrieved material actually relevant. Ideally, everything relevant is retrieved while at the same time everything extraneous is rejected, producing recall and precision values equal to 1.

3. Standard Document Retrieval

The great majority of the existing, operating retrieval systems are based on a manual assignment of keywords, or index terms, to documents and search request, and on the use of an "inverted file" organization. Vocabulary control during indexing and searching is normally provided by means of printed authority lists and thesauruses.

The operations of an inverted file system are best explained by reference to the diagram of Fig. 1. An inverted file directory is used which supplies for each index term acceptable to the system a list of references, or accession numbers, for all documents identified by the given term. The accession lists corresponding to the various query terms are then merged to reflect the correct query formulation, and the document citations for all items exhibiting the appropriate combination of terms are retrieved from a citation file. Whereas only one access per query term is normally required into the inverted directory, a separate access is needed in the citation file for each retrieved document citation. The number of file accesses, and hence the retrieval time, thus depends on the length of the query, and on the number of retrieved items.

As is true of all keyword retrieval systems, the retrieval process separates the collection into two parts, containing respectively the retrieved items — those identified by the appropriate term combination — and the rest which are not so identified. No ranking, or order of preference is provided within either set, and each retrieved item is inherently considered to be as important as any other. Thus to obtain a desired level of performance in terms of recall and precision, it is necessary to find just the right kind of query formulation. For if the formulation is too broad, the retrieved set is likely to be very large, thereby imposing a great burden on the user and producing low precision even when the recall is fairly high; on the other hand, a narrow formulation is likely to restrict severely the level of recall which may be attained.

Lancaster has made a number of exhaustive studies of keyword retrieval system failures. A typical list is reproduced in Table 2. [8,9]. Fortunately, some of the most damaging features inherent in a keyword retrieval system operating with an inverted file directory are automatically absent from a computer-based automatic indexing system. In particular, the indexing and search policy need not be specified so precisely in a natural language system based on abstract processing, because the natural language contains some redundancy, and many different query formulations are usable to achieve a given performance level. Furthermore, when a computer is available, the similarity between a user query and a given stored document can be ascertained more precisely than by a simple count of matching index terms.

In the SMART system, for example, a vector matching function is computed for each query-document pair being compared, and a numeric coefficient of similarity is obtained ranging from zero for no similarity to one for perfect agreement. This simple refinement makes it possible to obtain a ranking of the output documents in decreasing order of the similarity with the query, thereby enabling the user to look only at the top item (the one exhibiting the greatest similarity with the query), or at the top five, or the top fifty, or the top five hundred.

The ranking feature produces considerable improvements in retrieval effectiveness, even if one remains in a conventional retrieval framework. Consider in this connection the performance figures of Table 3 which are again produced by a comparison of Medlars and SMART. Whereas in each search used for the output of Table 1, the number of documents retrieved by SMART was identical with that previously obtained by the corresponding Medlars keyword search, the SMART ranking ability is used in Table 3 in such a way that all documents with a sufficiently high query-document similarity are retrieved. To permit a comparison with the Medlars searches, the cut-off in the similarity coefficient is so chosen that over all 29 test queries the total number of retrieved items is the same for SMART as for Medlars; however, the idiosynchracies of the conventional keyword retrieval systems are avoided by allowing the SMART system to retrieve many of the most important items for each query, independently of the size of the Medlars output for that query.

A comparison of the figures of Table 3 with those of Table 1 indicates that a deficiency of the automatic indexing system of from 20 to 40 percent is reduced to 0 to 20 percent when the ranked output is used. The SMART thesaurus, in fact, produces a slight advantage over the conventional, controlled Medlars indexing.

While the ranking feature is clearly useful, the real virtue of an automatic retrieval environment consists, however, in the flexibility which may be introduced into the search and retrieval operations. Three main aspects must be considered in this connection:

a) each user involved in a retrieval situation potentially provides feedback information which may be helpful to future users; this information could be saved and utilized in future system operations;

b) changes in the composition of the user population could be reflected by corresponding changes in the document organization;

c) alterations in the stored information files must be accommodated by creating simple procedures for the addition of new items and the retirement of old ones.

Such a flexible retrieval environment is sketched in the remainder of this study, and evaluation results are given for the main file processing operations.

4. Interactive Search in Clustered Collections

A) Inverted versus Clustered File Organization

One of the obvious shortcomings of the conventional library operations is the static nature of the existing environment. Most of the procedures are apparently based on the premise that user characteristics and collection maintenance procedures are entirely unrelated. Furthermore, no attractive provisions are made for collection growth or retirement. A flexible retrieval environment, on the other hand, would specifically utilize collection control methods tailored to the user population, and would permit alterations in the query formulations, as well as in the document indexing data, based on feedback information gained in the course of normal system operations.

Unfortunately, the indexing information attached to queries and documents cannot easily be changed if an inverted file organization is used, because the inverted file directory and the file of document citations mentioned in Fig. 1 do not also include the full document term vectors which are needed for updating purposes.

A clustered file organization is therefore used with the SMART system in which documents carrying somewhat similar content descriptions are automatically grouped into clusters. [10,11] Each cluster is identified by a representative cluster profile, somewhat akin to the center of gravity of a set of mass points.* A search in such a clustered file is carried out by first comparing each query with the file of profile vectors. For those profiles which exhibit a sufficiently high similarity with the query, the individual document vectors in the corresponding clusters are examined next, and the document citations are ranked for output purposes in decreasing query-document similarity order, as previously explained. Both profile and document vectors are explicitly stored as shown in the search diagram of Fig. 2.

The essential difference between inverted and direct clustered file searches is that at some point the inverted search makes random accesses into the data base for individual document citations. The cluster search also makes random accesses, but only for groups of documents (namely those whose profile vectors are sufficiently similar to the user queries).

---

* A cluster profile is simply a set of weighted terms, representative of the documents included in the corresponding cluster. In theory, it is possible to use an inverted file organization to access the document clusters — for example, by inverting on the terms in the cluster profiles. In practice, a direct organization is much preferable since the complete document indexing information can then be stored and accessed together for all documents in a given cluster, and the file structure can be altered as required.

A detailed comparison of inverted and clustered file organizations indicates that the clustered file is more economical of storage and permits more flexible (feedback-type) searches. [12] For a specific number of file accesses, a fixed number of documents is retrieved with an inverted file, and generally high precision is obtained at a specific recall level. For the same number of file accesses, a cluster search can provide many or few items, and while the precision is generally smaller, the recall may be higher or lower.

However, even if the recall-precision performance of an inverted file search is satisfactory, that is, if the user is interested mostly in high precision, the inverted file organization <u>cannot</u> be used for flexible feedback searches because

a) the required file storage space would be doubled if full document term vectors were stored in addition to the inverted directory;

b) the search time which in an inverted system depends directly on query length would become excessive, because each feedback operation increases the number of search terms.

For a typical experimental collection of 1400 documents, Murray [12] stipulates a storage size requirement of 69 disk tracks for the clustered file, 71 tracks for the corresponding standard inverted file, but 119 tracks if complete document term vectors are stored in addition to citations. Similarly the number of file accesses per query is about 15 for the clustered file no matter how many items are retrieved; but it grows from 19 to 25 for the inverted file as the number of retrieved documents increases from 10 to 50.

Since the feedback searches and file reorganization methods are considered essential in future document processing systems, a clustered file organization is assumed in the remainder of this study.

B) Standard Query Alteration

Many different strategies can be used for taking an original user query and improving its formulation during the course of the retrieval operations. Thus, various kinds of dictionary displays can help the user in finding appropriate ways of expressing his information needs; typically, for each term originally present in a search formulation, a set of synonymous or related terms might be presented for the user's attention. Alternatively, the original query formulation can first be used to perform a trial search, and document information pertaining to some of the retrieved items — for example, document titles or abstracts — can be submitted to the user to help him rephrase the query in the most appropriate way. [13]

The preferred method of query alteration used with the SMART system is known as relevance feedback because the queries are automatically updated, based on relevance information furnished by the user about previously retrieved documents. Specifically, the relevance feedback process assumes that an initial search is first performed for each query entering the system. A small amount of output, consisting of some of the highest scoring documents is then presented to the user, and the user is asked to identify some of these documents as being either relevant to his information need (R), or nonrelevant (S). These relevance judgments are then returned to the system, and used automatically to adjust the search request in such a way that the query terms present in the relevant documents are promoted (by increasing their weight), whereas terms occurring in the nonrelevant documents are simultaneously demoted.

The R previously identified relevant documents and the S nonrelevant ones then serve to construct a new query formulation q' which may be expected to be more similar to the relevant, and less similar to the nonrelevant documents, than the original query q. If the terms from the relevant items are added to the search request, whereas terms from the nonrelevant are subtracted, the query updating which implements the relevance feedback operation can be represented by equation (1)

$$q' = q + \alpha \sum_{i \varepsilon R} r_i - \beta \sum_{i \varepsilon S} s_j \qquad (1)$$

where r_i is the i^{th} document included in the relevant set R, s_j is the j^{th} document included in the nonrelevant set S, and α and β are constants.

An evaluation of the relevance feedback process indicates that of the various interactive retrieval methods, relevance feedback produces the best results, while at the same time placing the least burden on the user. [14] When the relevance feedback process is applied to the medical collection previously used for the SMART-Medlars comparison of Tables 1 and 3, the output of Table 4 is produced. It may be seen that after two feedback searches the SMART output is at least ten percent better than Medlars for the word stem analysis process, and twenty to thirty percent better when the thesaurus is used for analysis purposes.

Improvements in retrieval effectiveness up to 45 percent in recall and precision are possible with the relevance feedback method. The process does, however, require that file organizations be used which simplify the needed vector modification.

C) Standard Document Alteration

The query alteration process described in the previous subsection was based on information obtained from the user population in the course of the normal retrieval process. No good reason exists, however, for not also utilizing customer intelligence to help improve the document vectors themselves by promoting, so to speak, documents about which the user has reported favorably, while similarly demoting the others.

Specifically, when a number of documents retrieved in response to a given query are labelled by the user as "relevant", it is possible to render them more easily retrievable in the future by making each of them somewhat more similar to the query used to retrieve them. Similarly, retrieved documents labelled as nonrelevant are rendered less easily retrievable by being shifted away from the query. Hopefully, following a large number of such interactions, documents which are wanted by the users can be moved slowly into the active portion of the document space — that part in which a large number of user queries are concentrated, while items which are rejected are moved to the periphery from where, eventually, they may be discarded.

Brauen [15] has implemented and tested a document space modification process using the following strategy:

a) the document vector for an item identified as <u>relevant</u> during the feedback process is altered by adding query terms, or incrementing the weights of terms jointly present in the document and query vectors; on the other hand, document terms absent from the query are decreased in importance by being assigned a lower weight;

b) similarly, for documents identified as <u>nonrelevant</u>, the document terms jointly included in the document and query vectors are reduced in weight, while document terms absent from the query are increased in weight.

The procedure was tested by first using a set of 125 user queries to modify a given document space. A new set of 30 queries was then processed against the original document collection (prior to vector modification), and also against the final, modified space, following the processing of the earlier 125 queries. The output of Table 5 shows that the thirty new users profited from the earlier user interaction, since the retrieval results with the new, modified space are improved by 3 percent in normalized recall, and 8 percent in normalized precision* over the results obtained with the original space.

In a practical environment, it is of course necessary to consider the document space modification process as a permanent feature of the system, and over the years many thousands of vector alterations may be needed before an equilibrium condition is attained for the stored collection

---

* Normalized recall and precision are global measures derived from the standard recall and precision parameters. [10]

D) Document Profile Alteration

If the previously outlined document space modification is implemented for a clustered collection, a decision must be made about how each document profile should be represented, and whether the profiles should be modified when the document vectors in the corresponding clusters are changed.

Consider first the profile representation. If each profile were defined simply as the sum of all document vectors included in the corresponding cluster, each profile vector would include many hundreds of different terms, and the variations in term weights would be severe. Long vectors are, however, undesirable in view of the resulting high storage costs, and uneven weight distributions cause difficulties when the profiles are compared with the document and query vectors.

Experiments performed with a variety of profile definitions indicate that the best performance is obtained with short profile vectors which exhibit only small variations in the assigned term weights. The following conditions appear to be of main importance [12]:

a) profile weights should be derived from term <u>frequency ranks</u>, rather than from total term frequencies; that is, the term of lowest frequency is assigned weight 1, the next lowest frequency term receives weight 2, and so on;

b) up to eighty percent of the profile terms of lowest weight can be deleted, because the performance with the reduced profiles is equivalent to that with full profiles to within a few percent in both recall and precision;

c) the weights assigned to profile terms may be standardized for added storage economy, by replacing the full term weight distribution by only four different weight classes (that is, each profile term is assigned one of four distinct weights).

The profiles which result from these transformations are short and exhibit uniform weight characteristics. This reduces profile storage costs and simplifies profile manipulations.

When the document vectors are altered as a function of incoming user queries, as explained in the previous subsection, the corresponding profiles will in time become ineffective as a representation for the altered document cluster. A procedure implemented by Kerchner uses exactly the same strategy for profile alteration as for the earlier document space modification. [16]

Specifically, each time a relevant document vector is changed by addition (or upweighting) of one or more terms from a user query, these same query terms are also used to update the corresponding document profile. The weight of all profile terms also present in the user query are incremente by one; query terms not already present in the profile are added to the profile vector.

Table 6 shows the results of an experiment in which 175 search requests were used to update both the documents as well as the profiles of a clustered collection of 1400 items in aerodynamics. A new set of 50 queries not previously utilized for the document space alteration was then processed against both the original and the altered spaces. The improvement in performance obtained with the modified collection is seen to reach almost 10 percent in normalized recall and precision for the original queries included in Table 6 and to exceed ten percent for the first iteration feedback queries. [16]

Eventually, as the number and magnitude of the document space alterations increases, it will become necessary to allow for a shifting of documents from cluster to cluster. Such a reclustering operation is envisaged in the next section.

5. Dynamic Collection Environment

A) Collection Growth

Up to this point, the vector modification operations discussed in this study were all performed in response to the normal query processing, the intention being to keep the collection updated with changes in user interests. There are, however, even more important reasons for insisting on a dynamic document environment, and these have to do with collection growth and retirement. For the most part, library and information center personnel are fully aware of the severity of the problems created by normal collection growth, and by the lack of viable retirement policies for journal articles and monographs. Unfortunately, the state-of-the-art is such that the inevitable response of library administrations faced by accelerating document growth rates is an insistence for additional buildings, funds and personnel.

In the document environment stipulated in the present study, it is possible to institute document retirement policies based on actual experiences with the local user populations. Furthermore, new documents introduced into existing collections can be accommodated within the framework of the existing operations.

Consider first the question of collection growth. In the clustered environment, the principal question to be resolved concerns the manner of updating the clusters when new documents are introduced. The illustration of Fig. 3 shows two typical clusters on the left side of the figure. If new documents are simply added to the cluster identified by the closest profile vectors, a situation similar to that illustrated in the center of Fig. 3 results after some time. Eventually, it becomes necessary to generate completely new profile vectors if an accurate representation of the collection is wanted. The results of the required reclustering operation are shown for the sample document space on the right-hand side of Fig. 3.

In principle, it is desirable to reorganize the file as often as possible. In practice, the work involved in reclustering a sizeable file is so large that alternative methods must be used whenever possible. This is true even though one pass clustering methods can be utilized which operate in a time of order n to cluster n items, rather than of order n^2 as is the case for most classical clustering processes. [17] Murray [12] finds that the normalized recall and precision figures decrease by about 4 percent for a fifty percent updating range (that is, when about fifty percent of the document vectors in the collection are modified once or are newly introduced), and that the loss increases to 8 percent for a seventy-five percent updating rate. Murray concludes that "enough decay in the file organization occurs with 25 to 50 percent updating to warrant a complete reclustering operation; the break-even point is probably

Between reclustering operations, the collection can be updated by suitable changes in the profile vectors as new documents are introduced into the clusters. Three modes of profile alteration appear possible:

a) new documents are associated with the best existing cluster (the cluster for which the profile-document similarity is highest), but all profile vectors remain unchanged;

b) new documents are associated with the existing clusters, and the profiles are changed by updating only existing profile terms, that is, the weights of existing profile terms may be altered, but no new terms added, thereby insuring a constant profile dimension;

c) new documents are associated with the existing clusters, and the profiles are changed by updating the existing terms and introducing new terms taken from the documents added to the clusters.

To determine which of the profile updating methods operates most satisfactorily, an experiment was performed by Kerchner [16] in which a collection of 700 documents in aerodynamics was augmented by a new set of 700 additional documents in the same subject area, thus accounting for a total updating rate of fifty percent for the complete collection. For each added document all three profile updating methods were utilized, that is, no updating, updating of existing profile terms only, and updating of existing as well as addition of new terms. Furthermore, the organization of the collection was also regenerated by a complete reclustering procedure.

The output of Fig. 4 represents the average performance of 50 queries using each of the four document organizations. Fig. 4 (a) plots the recall against the precision values, the curve closest to the upper right-hand corner (where both recall and precision are optimal) representing the best performance. Fig. 4 (b) contains selected precision values at certain fixed recall points; the percentage improvements over the values obtained with the original space involving no profile updating are also listed.

As expected for a fifty percent updating rate, the completely reclustered collection provides the best performance, and the unaltered profiles are least effective. Of the two profile alteration methods, the one keeping a constant profile length is somewhat preferable, particularly for high recall values; it is of course also the preferred method from the viewpoint of storage space utilization.

One other collection growth policy deserves investigation, and that involves the maintenance of user query clusters, in addition to the normal document clusters. Under this policy, incoming queries are clustered in a manner similar to the documents, each query cluster thus representing a composite user profile. New documents entering into the collection can then be matched against the existing query profiles (rather than the document profiles) before being assigned to the appropriate document clusters. Such a strategy remains to be fully investigated. [18,19]

B) Document Retirement

Possibly the most important current problem in library management — one which has been attacked with remarkably little success — is the question of document retirement. By retirement is meant not a complete loss of an item, but merely its removal from the central file system — the one searched every time — to an auxiliary storage area which may be accessed only in special circumstances.

The classical approach to collection retirement consists in introducing concepts such as the half-life of a collection, that is, the time required for obsolescence of one-half of the currently published literature [20,21,22] , or the utility of a document in terms of the number of references that it can be expected to attract in its context during the remainder of its existence [23,24] . In each case, a short half-life, or a low utility factor implies rapid obsolescence, and therefore potential retirement of an item. In order to translate concepts such as these into an active retirement policy, measurements must be made to be translated into an obsolescence threshold, and these measurements usually take one of three forms:

a) the number of citations to a given item in the literature which post-dates the item in question (e.g. the number of bibliographic references in later journal papers);

b) the usage of a given item as measured by the number of times it is removed from the library shelf, or the length of the borrowing period when the item circulates;

c) the age of the item, that is, the number of years since it was first published.

In each case, a low citation count, a small rate of usage, and high age lead to a high obsolescence figure.

Unfortunately, the approach based on measurements is impossible to implement, for practical purposes, because accurate values of these parameters (except for the age factor) are basically unobtainable. First of all, each measure depends on a specific library or user environment, and the calculated values cannot be translated into other environments. Second, it is not clear what periods of observation, and sample sizes are required in order to obtain reliable measurements. Finally, for the technical literature, at least, it is important to distinguish the general scientific usage of a document from its historical usage (for survey and other retrospective purposes). Obviously, the rates of obsolescence are different for the two cases. In summary, a retirement policy based on measures which cannot accurately be determined is not likely to be successful.

A new policy is therefore proposed which directly uses the dynamic document environment previously introduced. Specifically, a generaliz document vector modification policy is suggested based on the following three factors:

a) the closeness of a given document to the set of query profiles, measured by the size of the similarity coefficients between query profiles and the documents;

b) the rank of a given document in the list of retrieved items, whenever it is retrieved in response to a given user query;

c) the user response whenever a given document is retrieved, that is, the judgment which the user renders concerning its potential utility to his information needs.

The idea is that documents located close to the centers of user interest (close to the query profiles), or retrieved with a low rank in response to a query (say in the top 50 documents), or known to be relevant to the users' needs, should be promoted by being shifted closer to the respective queries where user interest is concentrated; at the same time, documents which are far removed from the query profiles, or which are retrieved with very high rank (in the bottom 50 for example), or which are known to be nonrelevant to the users' needs, are demoted by being shifted away from the current query positions. If such a policy is implemented correctly, it is clear that items which are never wanted, or which are always near the bottom of the retrieved list, will be shifted toward the periphery, away from the active part of the file, and will eventually become irretrievable. Simultaneously, documents which are promoted will become more easily retrievable in the future if new queries, similar to the currently active ones should be received.

A number of problems will be met in such a dynamic retirement environment. First, it is important to pick adjustment parameters for the term weights that are large enough to make themselves felt eventually, but not so high as to cause violent perturbations in the document space. In particular, it is likely that different adjustment rates must be used for items about which something specific is known, for example in the form of a user relevance judgment, and for those which merely happen to be retrieved with high or low document ranks. This leads to the fast and slow increment functions which have been used successfully to update user profiles in certain information dissemination systems. [25]

Second, when documents are demoted, that is, when the shift is away from a given item (rather than toward a specified area), it is necessary to take special precautions to prevent the "disappearance" of a vector, that is, the reduction of all term weights to zero. [26] In particular, certain terms must be reinforced while others are demoted. The new procedures which automatically recognize good discriminating terms from nondiscriminators (common terms) can insure that the best discriminators are reinforced while the importance of the other terms is decreased. [27]

The automatic retirement policies remain to be tested experimentally. In view of the promising results obtained with dynamic query and document vector modifications following user interactions and document additions, one may expect that similar results will apply to document retirement.

The complete dynamic environment must also be subjected to appropriate cost studies. Obviously, any dynamic file process will require some apparatus not needed for static file management. On the other hand, the costs presently inherent in the maintenance of ever-growing files are staggering, and the trained personnel, storage space, and new buildings are becoming scarcer. A self-monitoring, dynamic environment such as the one proposed may then be easier to justify than would appear both economically and technically.

One may hope that, in time, the dynamic file environment described in the present study will be developed into a viable system for library and collection management.

References

[1] G. Salton, Computers and Libraries — A Reply, The Library Journal, Vol. 96, No. 18, Oct. 15, 1971, p. 3277-3282.

[2] W.N. Locke, Computer Costs for Large Libraries, Datamation, Vol. 16, No. 2, February 1970, p. 69-74.

[3] B.C. Vickery, Techniques of Information Retrieval, Archon Books, Hamden, Conn., 1970.

[4] G. Salton, Automatic Text Analysis, Science, Vol. 168, 17 April 1970, p. 335-343.

[5] D.R. Swanson, Searching Natural Language Text by Computer, Science, Vol. 132, 21 October 1960, p. 1099-1104.

[6] G. Salton, The SMART Retrieval System — Experiments in Automatic Document Processing, Prentice Hall Inc., Englewood Cliffs, N.J., 1971.

[7] G. Salton, A New Comparison between Conventional Indexing (Medlars) and Automatic Text Processing (SMART), Technical Report, Dept. of Computer Science, Cornell University, October 1971.

[8] F.W. Lancaster, Evaluation of the Operating Efficiency of Medlars, National Library of Medicine, Final Report, 1968.

[9] F.W. Lancaster, An Evaluation of EARS (Epilepsy Abstracts Retrieval System) and Factors Governing its Effectiveness, Report to NINDS, University of Illinois, October 1971.

[10] G. Salton, Automatic Information Organization and Retrieval, McGraw Hill Book Co., New York, 1968.

[11] G. Salton, Search Strategy and the Optimization of Retrieval Effectiveness, in Mechanized Information Storage, Retrieval, and Dissemination, K. Samuelson, editor, North Holland Publishing Co., Amsterdam, 1968, p. 73-107.

[12] D.M. Murray, Document Retrieval based on Clustered Files, Doctoral Thesis, Cornell University, Dept. of Computer Science, 1972.

[13] M.E. Lesk and G. Salton, Interactive Search and Retrieval Methods using Automatic Information Displays, Proceedings Afips Spring Joint Computer Conference, Afips Press, Montvale, N.J., 1969, p. 435-446.

[14] G. Salton, The Performance of Interactive Information Retrieval, Information Processing Letters, Vol. 1, 1971, p. 35-41.

[15] T.L. Brauen, Document Vector Modification, in The SMART Retrieval System — Experiments in Automatic Document Processing, G. Salton, editor, Chapter 24, Prentice Hall Inc., Englewood Cliffs, N.J., 1971, p. 456-484.

[16] M.D. Kerchner, Dynamic Document Processing in Clustered Collections, Doctoral Thesis, Scientific Report No. ISR-19, Dept. of Computer Science, Cornell University, October 1971.

[17] D.B. Johnson and J.M. Lafuente, A Controlled Single-Pass Classification Algorithm with Application to Multilevel Clustering, Scientific Report No. ISR-18, Section XII, Department of Computer Science, Cornell University, October 1970.

[18] V.R. Lesser, A Modified Two-Level Search Algorithm Using Request Clustering, Scientific Report No. ISR-11, Section VII, Dept. of Computer Science, Cornell University, June 1966.

[19] S. Worona, Query Clustering in a Large Document Space, in The SMART Retrieval System — Experiments in Automatic Document Processing, G. Salton, editor, Chapter 13, Prentice Hall Inc., Englewood Cliffs, N.J., p. 298-310.

[20] R.E. Burton and R.W. Kebler, The Half-Life of Some Scientific and Technical Literatures, American Documentation, Vol. 11, 1960, p. 18-22.

[21] A. Sandison, The Use of Older Literature and its Obsolescence, Journal of Documentation, Vol. 27, No. 3, Sept. 1971, p. 167-183.

[22] R.S. Grant, Predicting the Need for Multiple Copies of Books, Journal of Library Automation, Vol. 4, No. 2, June 1971.

[23] B.C. Brookes, Obsolescence of Special Library Periodicals: Sampling Errors and Utility Contours, Journal of the ASIS, Sept.-Oct. 1970, p. 320-329

[24] B.C. Brookes, The Growth, Utility and Obsolescence of Scientific Periodical Literature, Journal of Documentation, Vol. 26, No. 4, December 1970, p. 283-294.

[25] C.R. Sage, R.R. Anderson, and D.R. Fitzwater, Adaptive Information Dissemination, American Documentation, Vol. 16, No. 3, July 1965, p. 185-200.

[26] J. Kelly, Negative Response Relevance Feedback, in The SMART Retrieval System — Experiments in Automatic Document Processing, G. Salton, editor, Chapter 20, Prentice Hall Inc., Englewood Cliffs, N.J. 1971, p. 403-411.

[27] G. Salton, Experiments in Automatic Thesaurus Construction, Proceedings IFIP Congress 71, to be published by North Holland Publishing Company, Amsterdam, 1972.

Analysis Method	Recall	Precision
Medlars (controlled terms)	0.3117	0.6110
SMART word stem	0.1814 (-42%)	0.4141 (-32%)
SMART discriminator list	0.2462 (-21%)	0.4518 (-26%)
SMART thesaurus	0.2181 (-30%)	0.4512 (-26%)

Natural Language Word Extraction versus
Medlars Controlled Terms

(450 documents, 29 queries)

Table 1

Analysis Method	Recall	Precision
Medlars (controlled terms)	0.3117	0.6110
SMART word stem	0.2622 (-16%)	0.4901 (-19%)
SMART discrimination list	0.2872 (-8%)	0.5879 (-4%)
SMART thesaurus	0.3232 (+4%)	0.6106 (0%)

SMART - Medlars Comparison using
Document Ranking Feature

(450 documents, 29 queries)

Table 3

Type of Failure	Explanation
I. Indexing	
a) failure of indexing policy	too few terms produce low recall and too many terms produce low precision;
b) failure of indexers	too many, too few or incorrect terms;
II. Index Language	
a) lack of specific terms in the vocabulary	affects mostly the controlled indexing languages;
b) ambiguous or spurious relations between terms	terms causing retrieval may be unrelated in the document being retrieved, or they are related but not in the way intended by the user;
III. Searching	
a) use of incorrect terms in the query	
b) use of wrong strategy level	too specific a search strategy in relation to need leads to high precision, but possibly low recall;
c) failure to cover all possible approaches to retrieval	lack of perseverance and perspicacity on the part of the searcher is likely to lead to recall failures;
IV. User-System Interaction	
a) user effort and time needed to find appropriate search formulation	user may not be able to obtain easy access to the controlled vocabulary listings, etc.;
b) requirement for perfection of entry	spelling and punctuation;
c) failure to allow for common human errors	each simple error may cause a complete search to be cancelled;
d) confusing or inexplicit system responses	user may not infer the cause of failure from the system response.

Common Retrieval System Failures

(adapted from Lancaster [9])

Table 2

Analysis Method	Recall	Precision
Medlars (controlled terms)	0.3117	0.6110
SMART word stem		
0 - initial search	0.2622 (-16%)	0.4901 (-19%)
1 - iteration feedback	0.3235 (+ 4%)	0.6385 (+ 5%)
2 - iteration feedback	0.3433 (+10%)	0.6892 (+13%)
SMART thesaurus		
0 - initial search	0.3232 (+ 4%)	0.6106 (0%)
1 - iteration feedback	0.3915 (+25%)	0.7427 (+18%)
2 - iteration feedback	0.4029 (+29%)	0.7438 (+22%)

SMART - Medlars Comparison Using Feedback Feature

(450 documents, 29 queries)

Table 4

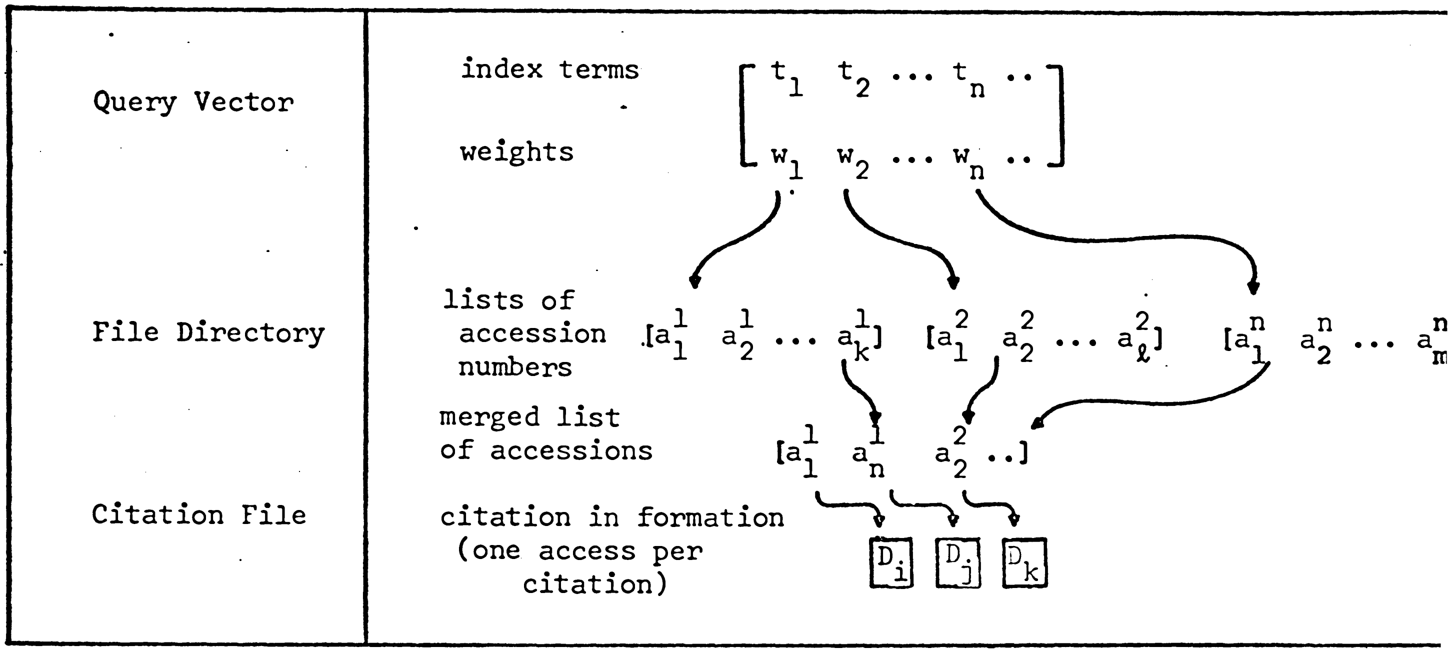
Type of Document Space	Normalized Recall	Normalized Precision
Original Space (standard run)	0.8975	0.6500
Modified Space 1 (β=24, γ=0.1, δ=12)	0.9079 (+2.8%)	0.7220 (+7.2%)
Modified Space 2 (β=30, γ=0.25, δ=8)	0.9081 (+2.9%)	0.7334 (+8.3%)
Modified Space 3 (β=30, γ=0.33, δ=8)	0.9083 (+2.9%)	0.7342 (+8.4%)

β, γ, δ are modification parameters introduced by Brauen [15]

Standard Document Space Modification

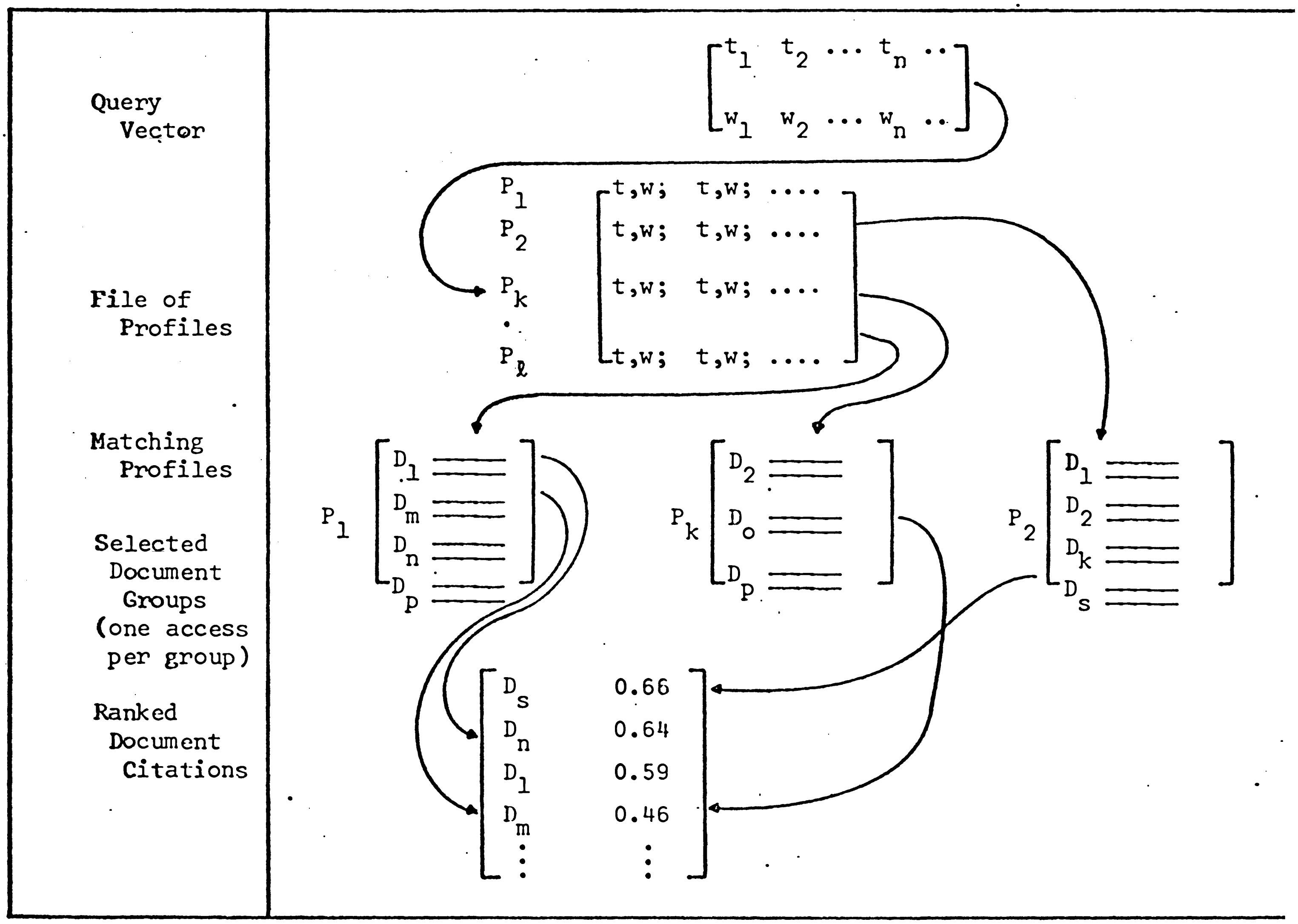
through Feedback Action

(424 documents, 30 queries)



Retrieval Strategy for Inverted File Structure

Figure 1



Retrieval Strategy for Clustered File Structure

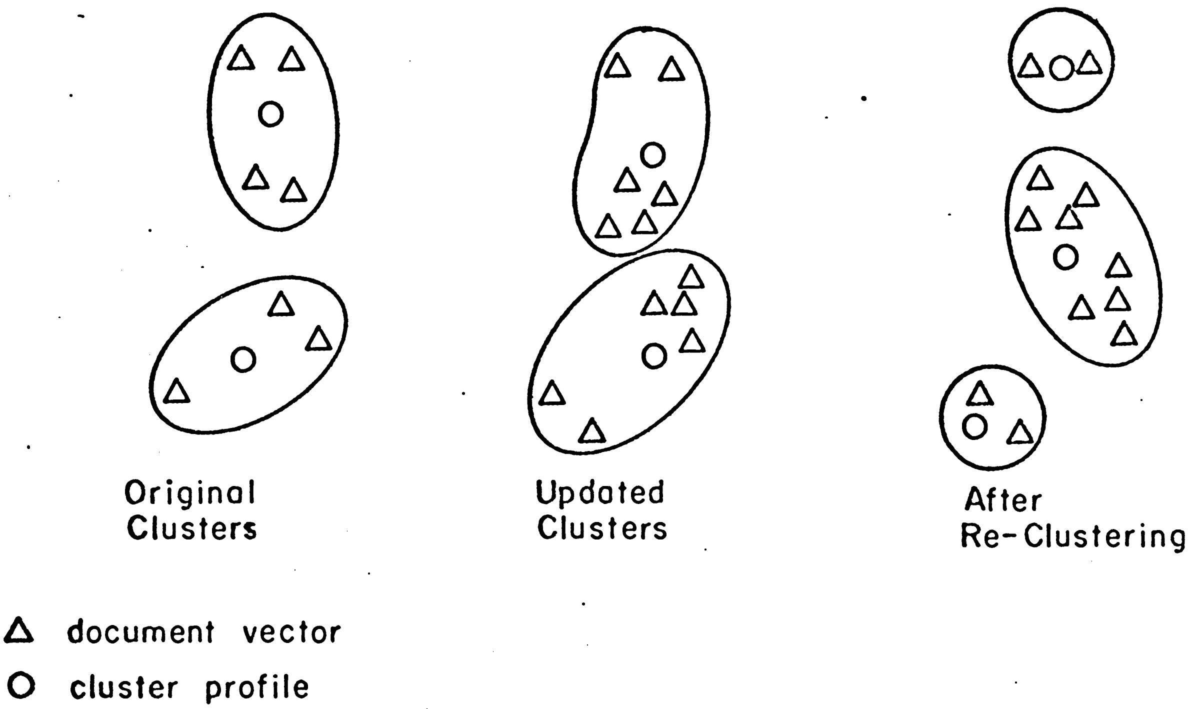
Figure 2

Type of Document Space	Normalized Recall	Normalized Precision
Original Space (initial run)	0.3874	0.3317
Updated Profiles due to Feedback Action (initial run)	0.4187 (+8%)	0.3616 (+9%)
Original Space (first feedback iteration)	0.3879	0.3748
Updated profiles due to Feedback Action (first feedback iteration)	0.4346 (+12%)	0.4094 (+9%)

Profile Updating due to Feedback Action

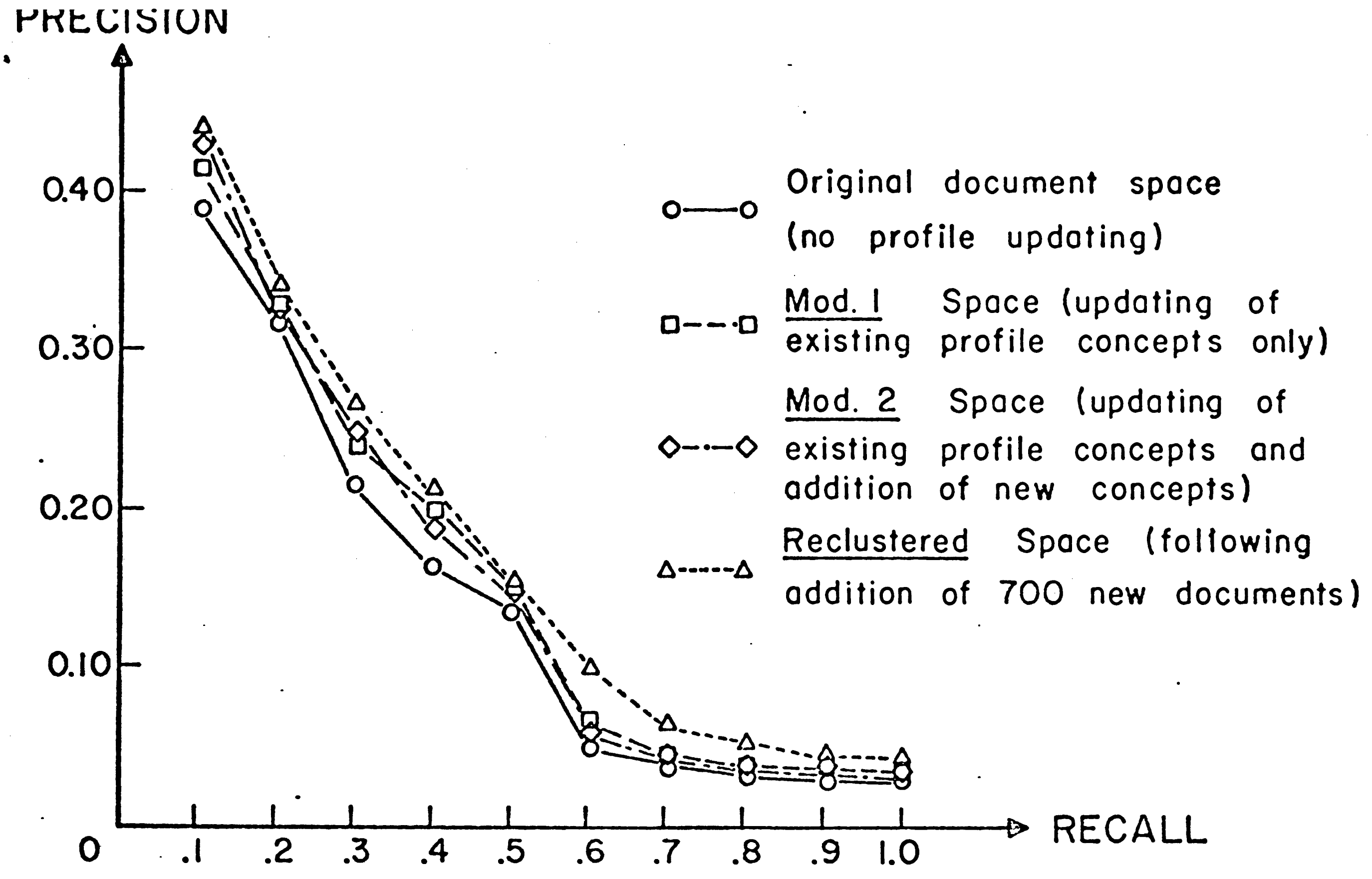
(1400 documents, 50 queries; adapted from [16])

Table 6



Cluster Updating with Profile Alteration

(new document addition)



a) Recall-Precision Graph

Recall	Precision			
	Original Space	Mod.1 Space	Mod. 2 Space	Reclustered
0.1	0.3981	0.4234 (+ 6%)	0.4296 (+ 8%)	0.4313 (+ 8%
0.3	0.2190	0.2432 (+11%)	0.2460 (+12%)	0.2631 (+20%
0.5	0.1390	0.1547 (+11%)	0.1526 (+10%)	0.1408 (+ 1%
0.7	0.0422	0.0458 (+ 9%)	0.0444 (+ 5%)	0.0717 (+70%
0.9	0.0282	0.0287 (+ 2%)	0.0285 (+ 1%)	0.0388 (+38%

b) Recall - Precision Table

Figure 4

Document Profile Updating following Addition of New Items

(1400 documents, 50 queries; adapted from [16])