Evaluation Problems in Interactive Information Retrieval

G. Salton

Technical Report
No. 69-39
August 1969

Department of Computer Science
Cornell University, Ithaca, N. Y.

Evaluation Problems in Interactive Information Retrieval

G. Salton [+]

Abstract

Interactive retrieval procedures are normally based on rapidly accessible files. Special storage organizations and file search techniques are used, and the system user is made to fulfill an important role during the retrieval process.

In the present study, the interactive retrieval environment is briefly examined. The special problems which arise in the evaluation of interactive retrieval are then discussed, and methods are described for evaluating partial file searches and user feedback techniques. Evaluation results obtained with the SMART system are presented.

1. Introduction

Over the last few years, increasing attention has been given to the design of information retrieval systems which could provide customized services to individual users, and furnish rapid responses to user requests. While the majority of the operational computer-based retrieval systems are still functioning in a batch-processing mode — that is by accumulating incoming requests for a period of time, and then processing the whole set of queries serially against the complete file — there now exist a number of experimental systems, based on high-capacity random-access storage mechanisms and conversational input-output con-

---

[+]Department of Computer Science, Cornell University, Ithaca, N. Y. 14850

This study was supported in part by the National Science Foundation under grant GN-750.

soles, which make it possible to process queries individually and permit a rapid interchange of information between users and system. [1,2,3,4,5,6]

In such an environment, it is no longer sensible to process the complete file against a given incoming query, since the response time would then become excessive. Instead, certain portions of the file must be chosen, and only these portions are then actually compared with the query statement. Furthermore, the users may influence the outcome of the search by providing feedback information to the system in the course of the operations.

In the present study, the organization of interactive retrieval systems is first briefly examined. File organizations based on the use of inverted files and on document clusters are considered in detail. Thereafter, the problems of search evaluation in an interactive retrieval situation are treated, and it is shown that special parameters — in addition to recall and precision — are useful to measure both the machine effort required to locate relevant information, as well as the user effort expended during the search. Finally, procedures are outlined for evaluating cluster searches in which only a portion of the file is examined, and searches in which user feedback information plays an important role. Evaluation results obtained with the experimental SMART document retrieval system are presented.

2. File Organization for Real-time Retrieval

A) Inverted Files

Two principal file organizations are presently used for operational retrieval situation, known respectively as the direct and inverted systems. In the direct system, the file is stored in order by document references numbers, and items to be retrieved are identified by a sequential scan of the complete file. Obviously, the direct file system cannot be used if rapid responses are to be made available to waiting customers.

The inverted system on the other hand, is based on an arrangement of the information file in order by the main information identifiers, normally a set of keywords or index terms, or alternatively a set of author names, or possibly journal titles. Thus, all documents indexed by keyword A are listed together, followed by those identified by B, C, D, and so on. Each information item is then listed as many times as there are assigned keywords. To retrieve information stored in an inverted file, it is then necessary to extract from the file the sections that correspond to the index terms used to formulate the search request. If the number of applicable index terms is small, this operation requires only a small number of file accesses. Thereafter, the document references listed under the various index terms must be examined to see whether the search logic is satisfied.

The operational systems which currently provide fast responses to incoming queries are based, without major exception, on an inverted file organization. Usually, a small, controlled vocabulary is available for indexing purposes, and document as well as query indexing are performed manually. The search operations are then often handled automatically, using up to four distinct files, as shown in Table 1. A term index, or directory, makes available the starting addresses of the inverted file blocks listing the document references for the individual keywords. The directory may be stored in fast storage, whereas the inverted file of document references is stored on disk, or other similar medium. Once the individual document numbers responding to the query statement are known, a document index may be consulted to obtain pointers to the addresses of the corresponding information in the document file. The total number of disk accesses to be performed is then equal to the number of terms included in the search request (to obtain the corresponding document references from the inverted file), plus the number of documents actually displayed for the user (to obtain the document information).

In some of the practical applications, the terms used in formulating the search requests may be submitted to a normalization process before being used in the search process. This may be done, for example, by reducing each full term to a term stem, in which case the inverted file becomes a file of term stems. [2,3,4] Alternatively individual terms may be combined into phrases, by including in the inverted file the phrase component number of the given term, as well as the names of the remaining components of each applicable phrase.[2] Finally, term expansions are possible, by including in the inverted file or possibly in the term directory, the names, or references to names, of terms related to those initially available. [1]

The principal advantages of the inverted file system are provided by the fast query-document correlation process. It is generally only necessary to examine the few lists of document references corresponding to the query formulation, and documents which do not match the search logic are never retrieved from the file. The disadvantages of the inverted file system are in part those afflicting any organized file, in that the file updating problems are severe; in addition, only a small number of usable terms is normally provided — the more terms apply to a given document, the more often must the corresponding document references be listed in the inverted file. Thus, the search time depends on the length of the query statement, and on the type of query alteration which takes place during the search process. Since the query updating process, based on the display of term dictionaries, or alternatively on information extracted from previously retrieved documents is a principal advantage of real-time systems, an inverted file system is not necessarily optimal in an interactive environment.

B) Clustered File Organization

It is well known that groups of related terms, or documents can be generated automatically from an indexed document collection. [7,8] The following steps are normally used for this purpose:

a) a term document description matrix is formed exhibiting the list of terms or concepts assigned to the various documents;

b) a term-term, or document-document similarity matrix is formed, listing for each pair of terms, or documents, a similarity coefficient based on similar term assignments to the documents;

c) the similarity matrix is purged by deletion of similarity coefficients which do not reach a given threshold value;

d) classes of terms, or documents are formed by starting with single terms, or documents, as seeds and adding similar items according to a given grouping method;

e) duplicated or highly overlapping classes are eliminated;

f) a class description, or centroid vector, is constructed to represent each class, consisting of an aggregate of the descriptions of the included terms or documents.

A typical class description vector is shown in Table 2 for a collection of four documents identified by ten possible concepts. It is seen in the example of Table 2 that the weight of each concept in the class vector is defined as the mean of the weights of the same concept in the individual documents.

In a clustered file organization, the documents are then grouped into, possibly overlapping, classes of items, and incoming queries are compared initially with the class description vectors only. These class description vectors are stored internally, and replace the term index and inverted term files shown in Table 1. For those classes whose description vectors are sufficiently similar to the query description, the individual term-document descriptions are next examined, and documents similar to the queries are retrieved in decreasing similarity order. The examination of the document vectors included in certain classes is simplified if the document file is stored in class order, or if the document descriptions within a single class are chained together by a set of pointers. The files used with a clustered organization are listed in Table 3.

The main advantage of the clustered file organization resides in the fact that the number of file accesses, proportional to the number of document classes examined, does not depend on the number of terms present in the query or in the documents. Thus, no disadvantage results if large numbers of terms or concepts are assigned to queries and documents, as is the case with many automatic indexing methods. Furthermore, the interactive procedures, which normally consist in updating the query formulations by addition (or subtraction) of concepts derived from stored dictionaries or from previously retrieved document vectors, do not require special access to the document file. It is also possible in a clustered organization to vary the amount of information actually considered by examining a smaller or larger number of document classes.

The disadvantages of the clustered organization have to do with the document grouping operation itself. The standard grouping methods require of the order of N^2 operations, where N is the number of documents (or terms) being grouped, since the construction of a document, or term, similarity matrix normally depends on a comparison of each item with each other. There is also the problem of class updating and reorganization which must be undertaken as the composition of the file changes by item additions or deletions. In practice, it is likely that fast grouping methods must be used which dispense with the similarity matrix, and operate instead with a set of initially available classes, or with a random grouping of items to be refined by subsequent interchanges of items from one group to another. One such clustering method, investigated in connection with the SMART system requires of the order of N · m operations where N is the number of items, and m the number of item classes to be constructed. [9]

3. Evaluation for Real-time Retrieval

A) Standard Cost Evaluation

Much of the retrieval evaluation work presently described in the literature is based on two user parameters, known as recall and precision, which measure respectively the amount of relevant material actually retrieved, and the amount of retrieved material actually relevant. In a real-time environment, the amount of stored information examined during a search does not remain constant from one search to another, and neither does the amount of user effort expended during the search. Thus, the cost related parameters of <u>machine effort</u> and <u>user effort</u> play a more important role than in standard serial searches (where the machine effort is fairly constant, and the user effort small or nonexistent). Before examining this problem in more detail, it is appropriate to consider the type of cost evaluation applicable to standard systems.

In the literature, two types of cost evaluation are extensively discussed. The first one, relating to operational situations, computes the costs of the individual process steps, and multiplies this cost by the existing volume figures to obtain total cost estimates (alternatively, the cost of the individual process steps may be obtained by dividing total allocated budget figures by the actual volume of operations).

Thus, one can ascertain the cost in time and personnel required by the manual term assignment, keypunching, file entry, searching, generation of title lists, output editing, typesetting, and so on, and multiply the respective amounts by the number of citations stored, or the average number supplied in answer to the queries. This produces representative amounts for the costs of the various retrieval operations. [10,11,12]

An alternative method for conducting a cost evaluation would construct a mathematical model of the system to be evaluated, and use this to compute costs under various assumed operating conditions. Such a simulated evaluation is advantageous in that the results may be applicable to many diverse situations; the evaluation procedure can then be used to forecast the cost of future systems to be implemented, or of changes in the design of existing systems. [13,14,15,16] In general, the model must be based on three types of information:

a) a statement of time and cost figures for equipment, personnel, materials, and procedures used in the system;
b) an account of actual operational data such as file size, volume figures, user population served, and so on;
c) a set of equations relating the various parameters including materials, equipment and search queries.

The difficulties of the simulation approach are those always encountered when modeling large systems. Ideally hundreds, or possibly thousands of parameters need to be included in the model, and the interrelationships between them are often unknown, and may in any case not be representable by a solvable set of equations. The model must then be simplified either by eliminating many variables, or by linearizing the relationship between parameters. Unhappily, the simplified model may no longer represent an actual situation, and the resulting cost analysis may not be trustworthy.

One additional shortcoming of most existing cost models is the absence of user parameters. This is due in part to the difficulty of measuring individual user actions performed in an interactive environment, [16] and in part because the cost of an imperfectly specified series of man-machine interactions is difficult to relate to any specific user benefit [17] — the value of stimulating a user's thoughts as a result of a successful interaction is not easily measurable. There is also the additional complication which arises from the fact that in an interactive environment a given user does not normally operate alone. In fact, several users are likely to be active simultaneously, and the cost of an interaction then depends on several additional parameters as follows: [18]

a) the maximum number of simultaneously active users;

b) the response time (elapsed time to completion) when less than one "time slice" is needed to complete the task (a time slice is a time period made available by the system to a given program; tasks which are not completed in a given time slice are disconnected and must reenter an input queue before a new time slice is allocated);

c) the average task input rate;

d) the average processor time required to complete tasks needing less than one time slice for completion;

e) the ratio of tasks requiring less than one time slice to completion to total number of simultaneously active users;

f) the elapsed time multiplication factor, that is the excess of time needed to complete a given task requiring more than one time slice in a given task mix, over the time needed by that task assuming it were the only one active in the system;

g) the processor elapsed time to completion, that is the time elapsed between entering the last character of a task at a terminal and receiving the first character of the final output at the terminal.

At the present time, no cost models appear to have been constructed with a sufficient sophistication to include the foregoing types of parameters. More modest efforts have, however, taken place to evaluate user behavior, or user effort, in an interactive environment. These are briefly described in the next few paragraphs.

B) Evaluation of User Effort

A recent study was made of a set of users in a problem solving situation, using either a conventional batch processing system, or a time-sharing system permitting user interaction. [19] The tasks were open-ended in the sense that only the final goal was specified but not the processing sequence needed to reach the goal. Thus, a change in parameters or a better strategy might lead to improved output results. The users were then to continue operating until either a satisfactory result had been achieved, or the allocated time had run out. In that sense, the work was in fact similar to an information retrieval task.

Four types of evaluation parameters were used to ascertain the effectiveness of the performance:

a) the cost of using the system, including amount of user resources as well as the computer resources and their value (computer resources were estimated at \$350.— per hour for batch processing, and \$485.— per hour for time sharing use);

b) the effectiveness of the performance produced by the system, measured by recall and precision for information retrieval tasks;

c) the speed with which the results were produced, that is, total time expended and time per interaction;

d) the attitudes of system users, that is, the preceived ease of use of the system, the ease of access, the perceived value of the output, and so on.

The main results of the study showed that the overall <u>cost</u> of the problem solution did not differ substantially between the time-sharing and batch processing environments. In fact, the higher computer cost in the time sharing situation was compensated by lower user costs, because responses were available so much earlier. On the other hand, in the batch processing mode, the user costs were up and the computer costs down. The following specific data were obtained:

a) the time sharing users reached their most effective decision rule after 6.5 hours; the batch users after 12 hours;

b) a higher perception and understanding of the problem was found by the time-sharing users;

c) the maximum performance was also in favor of the time-sharers;

d) the computer time showed a large advantage for batch processing, using only about one fifth of the machine time required by the time sharing group;

e) for the time sharing users a relation was found to exist between performance and computer time used in reaching a solution.

These results appear to be borne out by a recent study of various interactive procedures used for document retrieval tasks. [20] In that study, a variety of interactive search and retrieval procedures were considered in which information supplied by the user population was taken into account in an attempt to achieve improved system responses. Two main types of user-system interaction were studied in detail:

a) the pre-search methods in which refined query formulations were constructed, using dictionary displays and similar methods before any file search was actually attempted;

b) the post-search methods in which an original query is first processed, and a query reformulation is attempted after the results of an initial search are actually available.

The various query updating methods used in that study with a collection of 200 documents and 42 queries in aerodynamics are summarized in Table 4, and the main evaluation results are contained in Table 5.

The performance summary of Table 5 shows again that computer costs, and to some extent user effort correlate with system performance.

In fact the best methods from the performance viewpoint (measured by recall and precision), are the post-search methods. However, these methods require two separate file search operations — one prior to the interactive process and one following it. The computer demands are therefore comparatively higher for post-search than for the other methods. From the user's viewpoint, the less information is displayed, the easier will normally be the interactive process. The post-search methods displaying information relating to previously retrieved documents are therefore relatively onerous in terms of user effort. The exception is the relevance feedback process where the user must merely identify some of the previously retrieved documents as either relevant or nonrelevant. The partial cluster search methods are of course least expensive in terms of computer time, and the user effort is not larger than for full searches. However, the performance of partial search methods in an interactive environment remains to be evaluated.

4. Feedback and Cluster Search Evaluation

It was stated in the previous sections that cluster search techniques and interactive query adjustment methods appear to be advantageous in real-time information retrieval. At the same time, a performance evaluation seems even more difficult in an interactive environment comprising many users, than in a standard batch processing situation, because computer costs for individual tasks are harder to assess, and user effort is difficult to measure. The result is that many different parameters appear to be needed to measure system performance, depending on the particular viewpoint, and these parameters are hard to combine into an overall performance coefficient.

This suggests that it would be useful to alter the standard system evaluation by incorporating into the normal performance measures parameters reflecting computer and user costs. Some efforts have already been made in this direction by Lancaster and Keen who defined the "novelty ratio" and "relative recall", respectively, as parameters reflecting to some extent the user satisfaction. [21] The application of these measures to an interactive user environment is obvious from the definitions:

novelty ratio: the proportion of items retrieved and judged relevant by a user of which he was not previously aware;

relative recall: the proportion of relevant items retrieved and seen by the user compared to the total relevant which the user would like to see.

In the remainder of this section, additional modifications to the standard recall and precision measures are suggested to incorporate the computer and user parameters which appear most important in a real-time retrieval environment.

A) Cluster Search Evaluation using Correlation Ratio

It is customary to measure the efficiency of a search operation by computing the number of required query-document matches. In the case of a standard file search, the operation is then of order N, where N is the number of documents in the collection. Consider now a clustered collection consisting of x group of approximately N/x documents each. If the documents in k of the x groups ($k \leq x$) are individually examined, then the total number of query-document matches required for the clustered collection is equal to

$$x + k\frac{N}{x} .$$

The advantage inherent in the cluster search, over and above a standard full search may be then measured by the correlation percentage (C.P.) as follows:

$$\text{C.P.} = \frac{1}{N}\left\{x + k\frac{N}{x}\right\} .$$

It is seen that this percentage can theoretically exceed 1 when k approaches x. In practice, however, k may be expected to be small, so that the correlation percentage will normally lie below 0.25.

The standard evaluation output, consisting for the SMART system of recall-precision tables and graphs, do not reflect the correlation percentages. Indeed, in the SMART evaluation process, it is customary to keep the precision constant when a relative recall of 1 is reached, or alternatively to assign ranks of N, N-1,...N-r+1 to the r relevant documents which are not retrieved (in case the relative recall never reaches a value of 1).

This situation is illustrated in Table 6 for a total of 10 documents of which two are assumed relevant, and for a cut-off after 5 retrieved documents. It is seen in Table 6(a) that if the relevant are all retrieved before the cut-off is reached, the precision stays constant after the last relevant is reached (after rank 4 in the illustration). On the other hand, if not all relevant are retrieved, the example of Table 6 shows that the relevant which are not retrieved are assigned the lowest possible ranks for computational purposes (10 in the illustration of Table 6(b)). Obviously, these convertions do not take into account the actual number of query-document correlations performed (5 for the examples of Table 6).

Dattola [9] has suggested that the correlation percentage be incorporated into the normal recall-precision evaluation to reflect the machine effort devoted to the search as follows:

a) if all relevant are retrieved before cut-off, the precision is allowed to drop up to a rank corresponding to the correlation percentage; thereafter, the precision stays constant (for full searches, this implies that the precision is permitted to drop all the way to the last document);

b) if all relevant are not retrieved before cut-off, the unretrieved relevant are assigned ranks which are uniformly distributed throughout the range of ranks greater than the total number of correlations performed.

The example of Table 7 for a correlation percentage of one-half shows the new evaluation for the cases previously exhibited in Table 6. It is seen that in case (a) the precision now drops down to 0.4, after which it remains constant. In case (b), the unretrieved relevant is assigned the mean rank of 8 (between ranks 6 and 10). The precision stays constant after rank 8, since the correlation percentage has already been used in computing the ranks of the nonretrieved relevant items. It is clear from the examples, that the higher the correlation percentage, that is, the larger the number of query-document correlations actually performed, the lower will the precision be permitted to drop. Thus, if everything remains equal except for the correlation percentage, the search with the highest correlation percentage, representing the largest machine effort will produce the lowest recall-precision graph.

Fig. 1 shows a typical comparison between a standard full-search and two clustered searches using the new evaluation process. [9] Document-level averages are shown in each case, averaged over 42 queries for a collection of 200 documents in aerodynamics. For the clustered searches, the collections were partitioned into 13 clusters with an overlap of 1.5 percent, and 15 clusters with an overlap of 0.5 percent, respectively. The correlation percentages for these two cases were 0.23 and 0.20, respectively. The results of Fig. 1 show that the clustered search with correlation percentage equal to 0.23 is more effective than the full search up to a recall of about one-half. For higher recall values the full searches remain generally preferable.

B) Feedback Evaluation with Constant User Effort

The problems which arise in feedback evaluation are somewhat akin to those affecting cluster search evaluation except that it is the <u>user effort</u> rather than the machine effort which proves troublesome. Consider the example of Table 8 in which the initial run and first feedback iteration are shown for a case with five documents initially retrieved and shown to the user. The user identifies two of the documents as relevant (nos. 229 and 68 with initial ranks of 1 and 4), and the query is updated using a standard relevance feedback procedure. [20] The document ranks, as well as recall and precision figures after the first iteration are shown on the right-hand side of Table 8.

While the performance is much improved after the first feedbackrun, it is clear from the Table, that the advantage is due to two entirely different circumstances:

a) the improvement in rank of documents already seen by the user and previously identified as relevant (document 68 in the example of Table 8);

b) the improvement in rank of relevant documents not previously seen by the user (document 67 in the example of Table 8).

Since the user is not necessarily interested in reviewing several times a document which had already been presented to him during previous search iterations, it seems that these two cases ought to be distinguished by removing from consideration during subsequent iterations documents previously seen by the user. In other words, the evaluation of each feedback iteration ought to be based on a constant amount of user effort (assuming that the same number of documents are presented each time), and recall and precision should not be made to depend on increases (or decreases) in the ranks of previously retrieved items.

Several procedures for a feedback evaluation independent of user effort have been outlined by Ide.[22] The first consists in freezing the ranks of previously retrieved items and in permitting items to move up only to the highest frozen rank plus one. For the example of Table 8, this means that documents 229 to 205 would remain frozen in ranks 1 through 5, and other documents not previously retrieved would be allowed to move up only to rank 6. The result of a frozen rank process is shown in Table 9(a) for the example of Table 8. It is seen that relevant document 67 which originally moved from rank 8 to rank 4 between initial run and first feedback iteration is now constrained to move only up to rank 6 since the higher ranks are preempted. This produces markedly worse performance indicators with the result that an evaluation based on frozen rank is biased against the search system. The rank freezing process should therefore not be used in practice.

The next feedback evaluation process is known as residual collection evaluation. Here the initial run is performed as usual. The documents seen by the user and utilized for query modification are then removed from the document collection to form a new reduced collection. The reduced collection which includes no documents previously shown to the user is then processed both against the original queries and against the modified queries, and the results of the reduced searches are compared to estimate the usefulness of the feedback process.

Table 9(b) shows the results of the first feedback run using the reduced collection. Document 67 now receives rank 1, instead of rank 4 as in Table 8(a), since the three documents of higher rank (numbers 229, 68, and 79) were removed from the collection. Similarly document 188 is assigned a rank of 2 instead of 5, 29 is assigned a rank of 3 instead of 6, and so on. The output of Table 9(b) must now be compared against the standard initial run (Table 8(a)) with the top five documents removed, that is, a run for which documents 16 to 30 in Table 8(a) receive ranks 1 to 5.

A typical residual collection evaluation is shown in graph form in Fig. 2(a) averaged over 200 documents and 42 queries. [23] The differences between the initial run and the first feedback iteration using the residual collection are shown by cross-hatching in Fig. 2(a). It is seen that the performance improvement due to the feedback process is of the order of fifteen percent for low recall values, and of the order of ten percent at high recall. For comparison purposes, an initial run and first feedback output using frozen collection evaluation is superimposed on the graph of Fig. 2(a). The performance differences for the frozen process are seen to be much lower because of freezing effect previously described.

The residual collection process provides an unbiased evaluation of the feedback performance. Its disadvantage in terms of work performed, both for the reranking of documents and the reprocessing with the reduced collections, is however not inconsiderable. Furthermore, while the <u>differences</u> in performance between the various runs are accurate, the reduced collection runs exhibit lower overall performance than the standard runs, since the relevant items previously retrieved are no longer included. The reduced collections will thus possess lower generality, reflected in the lower recall-precision graphs of Fig. 2(a).

An alternative feedback evaluation method in which the overall performance level can be maintained is the <u>test and control</u> methodology. [22,23]

The basic notion is to separate a document collection into two equal parts, while trying to maintain identical collection properties (number of documents, generality, etc.). The feedback process is then performed with one-half of the collection, the test collection, and the evaluation process is performed with the second collection, the control collection, which had not previously been used to modify the queries. The process is described in the flowchart of Fig. 3.

The results of a comparison between initial query processing using the control collection, and "test collection modified" query processing using the control collection are shown in the graph of Fig. 2(b). The results of Fig. 2(b) are averaged over 153 queries and 424 documents in aerodynamics, the test and control collections consisting of 212 documents each. The initial run output of Fig. 2(b) shows that the test and control collections used in the evaluation process were not unfortunately entirely comparable — the performance curve for the control collection is superior to that of the test collection. As a result, the differences due to feedback measured by the cross-hatched area of Fig. 2(b) may be somewhat unreliable (they should be of the same order of magnitude as those of Fig. 2(a)). Still, the test and control evaluation appears entirely appropriate if care is taken in the construction of the subcollections. The first iteration control curve, superimposed on the graph of Fig. 2(b) corresponds to the standard evaluation process, previously exhibited in Table 8, which does not distinguish between feedback and ranking effects for previously retrieved documents.

The evaluation procedures incorporating correlation percentages for partial searches, and constant user effort parameters for feedback runs can of course be combined when feedback methods are applied to cluster searches. The proposed methodology does not provide a complete cost evaluation but makes a start by considering user effort and search efficiency in addition to the standard recall and precision parameters.

References

[1] R. M. Curtice and P. E. Jones, An Operational Interactive Retrieval System, Arthur D. Little Inc., Cambridge, Massachusetts, June 1969.

[2] R. S. Marcus, P. Kugel and R. L. Kusik, An Experimental Computer-Stored, Augmented Catalog of Professional Literature, Proc. AFIPS Spring Joint Computer Conference, Thompson Book Co., May 1969, p. 461-473.

[3] W. D. Mathews, TIP Reference Manual, Report TIP-TM-104, MIT, Cambridge, Massachusetts, August 1968.

[4] E. B. Parker, SPIRES — Stanford Public Information System, 1968 Annual Report, Institute for Communication Research, Stanford, January 1969.

[5] I. H. Pizer, A Regional Medical Library Network, Bulletin of the Medical Library Associaiton, Vol. 57, No. 2, April 1969, p. 101-115.

[6] R. K. Summit, Dialog II — User's Manual, Report 6-77-67-14, Lockheed Missiles and Space Co., Palo Alto, September 1967.

[7] K. Sparck Jones and D. M. Jackson, The Use of Automatically Obtained Keyword Classifications for Information Retrieval, Final Report ML 211, Cambridge Language Research Unit, Cambridge, England, February 1969.

[8] 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.

[9] R. T. Dattola, Experiments with a Fast Algorithm for Automatic Classification, Information Storage and Retrieval, Report No. ISR-16 to the National Science Foundation, Department of Computer Science, Cornell University, 1969.

[10] F. Alouche, N. Bely, R. C. Cros, J. C. Gardin, F. Bély and J. Perriault, Economie Génerale d'une Chaine Documentaire Mécaniseé, Gauthier Villars, Paris 1967.

[11] M. M. Cummings, Needs of the Health Sciences, in Electronic Handling of Information: Testing and Evaluation, Thompson Book Co., Washington, 1967, p. 13-24.

[12] L. V. Overmyer, An Analysis of Output Cost and Procedures for an Operational Searching System, American Documentation, Vol. 14, No. 2, April 1963, p. 123-142.

References (Continued)

[13] E. C. Bryant, Modeling in Document Handling, in Electronic Handling of Information: Testing and Evaluation, Thompson Book Co., Washington, 1967, p. 163-173.

[14] N. R. Baker and R. E. Nance, The Use of Simulation in Studying Information Storage and Retrieval Systems, American Documentation, Vol. 19, No. 4, October 1968.

[15] C. P. Bourne and D. F. Ford, Cost Analysis and Simulation Procedures for the Evaluation of Large Information Systems, American Documentation, Vol. 15, No. 2, April 1964.

[16] A. Andersen and Co., Research Study of Criteria and Procedures for Evaluating Scientific Information Retrieval Systems, Final Report to the National Science Foundation, New York, March 1962.

[17] R. V. Wiederkehr, A Net Benefit Model for Evaluating Elementary Document Retrieval Systems, in Procedural Guide for the Evaluation of Document Retrieval Systems, Westat Research Inc., Report No. PB 182 710, Bethesda, Maryland, December 1968.

[18] S. Stimler, Some Criteria for Time Sharing Performance, Communications of the ACM, Vol. 12, No. 1, January 1969.

[19] M. M. Gold, Time Sharing and Batch Processing: An Experimental Comparison of their Values in a Problem Solving Situation, Communications of the ACM, Vol. 12, No. 5, May, 1969.

[20] M. E. Lesk and G. Salton, Interactive Search and Retrieval Methods using Automatic Information Displays, Proc. AFIPS Spring Joint Computer Conference, Thompson Book Co., Washington, May 1969.

[21] C. W. Cleverdon, The Methodology of Evaluation of Operational Information Retreival Systems based on a Test of Medlars, Cranfield College of Aeronautics, Cranfield, June 1968.

[22] E. Ide, Relevance Feedback in an Automatic Document Retrieval System, Cornell University Master's Thesis, Report No. ISR-15 to the National Science Foundation, Cornell University, Department of Computer Science, January 1969.

[23] C. Cirillo, Y. K. Chang, and J. Razon, Evaluation of Feedback Retrieval using Modified Freezing, Residual Collection, and Test and Control Groups, Report No. ISR-16 to the National Science Foundation, Department of Computer Science, Cornell University, 1969.

Type of File	Function	Storage Medium
Term Index or Directory	Stores the list of terms together with a pointer for each term giving the appropriate starting address in the inverted term file	internal core
Inverted Term File	Stores list of terms and for each term the list of applicable document references with weights	disk store or tape strips
Document Index or Directory	Stores list of document references together with a pointer for each term giving the appropriate address for document information in the document file	internal core
Document File	Stores document references together with appropriate document information including citations, abstracts, etc.	disk store, or tape strips, or photographic medium

Files used with Inverted Organization

Table 1

Docs \ Terms	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}
D_1	12	.	12	.	6	4	.	.	12	.
D_2	.	6	6	.	12	.	.	.	4	4
D_3	.	.	6	4	12	12	.	6	6	.
D_4	.	2	12	.	6	.	.	2	2	.
Class Vector	3	2	9	1	9	4	.	2	8	1

Formation of Document Class Representation

Table 2

Type of File	Function	Storage Medium
Class Vectors	Stores list of concepts and weights for each class vector and document references pertaining to each class	internal core
Class Index or Directory	stores for each class the starting address of the block of corresponding document vectors	internal core
Document File	stores document concept vectors and appropriate document information in class order	disk store

Files Used with Clustered Organization

Table 3

Query Alteration Process	Explanation
<u>Pre-Search</u>	
1. Repeated Concepts	User chooses query terms to be repeated for emphasis
2. Thesaurus Display	User chooses terms obtained from thesaurus display to update query (with or without time restrictions)
3. Word Frequency	User looks at display of word frequency information before updating query
4. Source Document	User looks at display of source document before updating
<u>Post-Search</u>	
5. Title Display	User looks at titles of first five retrieved documents before updating
6. Abstract Display	User looks at abstracts of first five retrieved documents
7. Relevance Feedback	Query is updated automatically using relevance judgments supplied by user following an initial search
<u>Combined Methods</u>	
8. Abstract plus Thesaurus	User looks at pre- and post-search information

Typical Query Updating Methods
(from Lesk and Salton [20])

Table 4

Processing Method	Computer Costs	User Effort	Precision Rise over Word Stem	
			Low R	High R
A) Fully Automatic				
word stem match	normal	none	-	-
automatic thesaurus	normal	none	+4%	+6%
B) Pre-Search Inter-action				
thesaurus display	normal +	medium	+6%	+4%
source document display	normal +	medium	+8%	+5%
C) Post-Search Inter-action				
title display	high	medium	+13%	+2%
abstract display	high	very high	+17%	+5%
relevance feed-back	high	low	+10%	+7%
D) Partial Search				
cluster search	low	none	+5%	-10%
cluster search with relevance feed-back	low +	low	?	?
cluster search with abstract display	medium	very high	?	?

Performance Summary for Interactive Methods
(from Lesk and Salton [20])

Table 5

Rank	Relevant	R	P
1	✓	0.5	1.0
2		0.5	0.5
3		0.5	0.5
4	✓	1.0	0.5
5		1.0	0.5
cut-off			
6		1.0	0.5
7		1.0	0.5
8		1.0	0.5
9		1.0	0.5
10		1.0	0.5

a) Relevant Ranks (1,4)
(all relevant retrieved before cut-off)

Rank	Relevant	R	P
1	✓	0.5	1.0
2		0.5	0.5
3		0.5	0.33
4		0.5	0.25
5		0.5	0.20
cut-off			
6		0.5	0.17
7		0.5	0.14
8		0.5	0.13
9		0.5	0.11
10	✓	1.0	0.20

b) Relevant Ranks (1,6)
(not all relevant retrieved before cut-off)

Standard Retrieval Evaluation

Table 6

Rank	Relevant	R	P
1	✓	0.5	1.0
2		0.5	0.5
3		0.5	0.33
4	✓	1.0	0.5
5		1.0	0.4
cut-off			
6		1.0	0.4
7		1.0	0.4
8		1.0	0.4
9		1.0	0.4
10		1.0	0.4

a) all relevant retrieved before cut-off

Rank	Relevant	R	P
1	✓	0.5	1.0
2		0.5	0.5
3		0.5	0.33
4		0.5	0.25
5		0.5	0.20
cut-off			
6		0.5	0.17
7		0.5	0.14
8	✓	1.0	0.2
9		1.0	0.2
10		1.0	0.2

b) all relevant not retrieved before cut-off

Evaluation Incorporating Correlation Percentage

Table 7

Initial Run			
Rank	Relevant	R	P
1	229 ✓	0.25	1.0
2	183	0.25	0.50
3	79	0.25	0.50
4	68 ✓	0.50	0.50
5	205	0.50	0.40
6	16	0.50	0.33
7	78	0.50	0.28
8	67 ✓	0.75	0.37
9	29	0.75	0.33
10	30	0.75	0.30

a) initial run

cut-off

First Feedback Run			
Rank	Relevant	R	P.
1	229 ✓	0.25	1.0
2	68 ✓	0.50	1.0
3	79	0.50	0.66
4	67 ✓	0.75	0.75
5	188	0.75	0.60
6	29	0.75	0.50
7	205	0.75	0.45
8	30	0.75	0.37
9	80 ✓	1.00	0.44
10	78	1.00	0.40

b) first feedback run

Standard Feedback Evaluation

Table 8

First Feedback Run			
Rank	Relevant	R	P
1	229 ✓	0.25	1.0
2	183	0.25	0.50
3	79	0.25	0.33
4	68 ✓	0.50	0.50
5	205	0.50	0.40
6	67 ✓	0.75	0.50
7	188	0.75	0.43
8	29	0.75	0.37
9	30	0.75	0.33
10	80 ✓	1.00	0.40

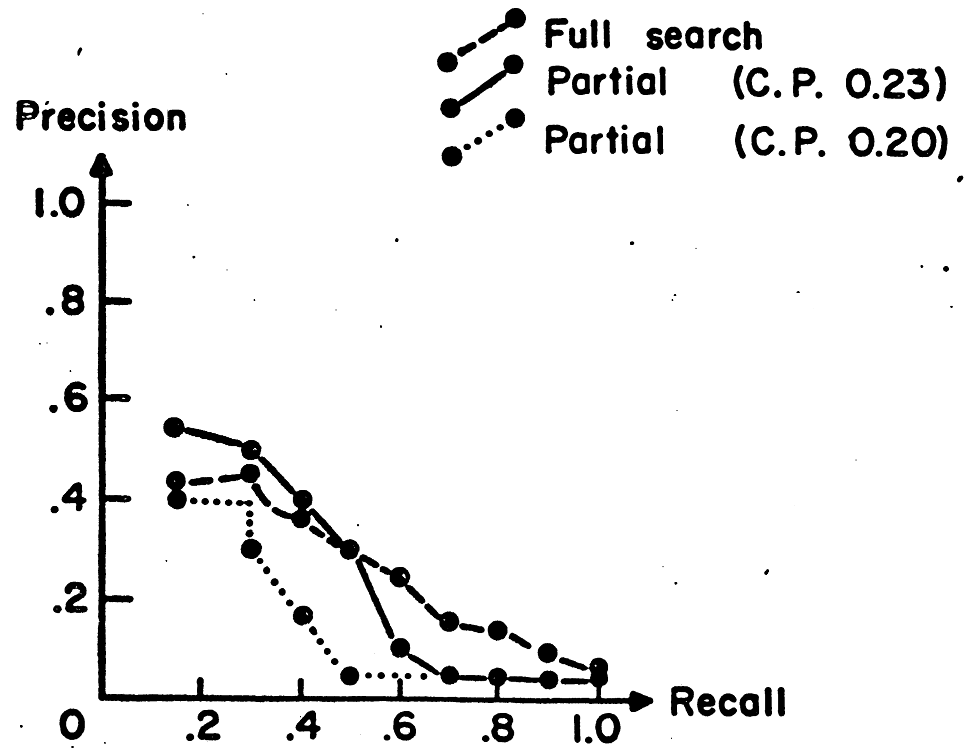
a) frozen rank evaluation

First Feedback Run			
Rank	Relevant	R	P
1	67 ✓	0.25	1.0
2	188	0.25	0.50
3	29	0.25	0.33
4	30	0.25	0.25
5	80 ✓	0.50	0.40
6	78	0.50	0.33
7	"	"	"
8	"	"	"
9	"	"	"
10	"	"	"

b) residual collection evaluation

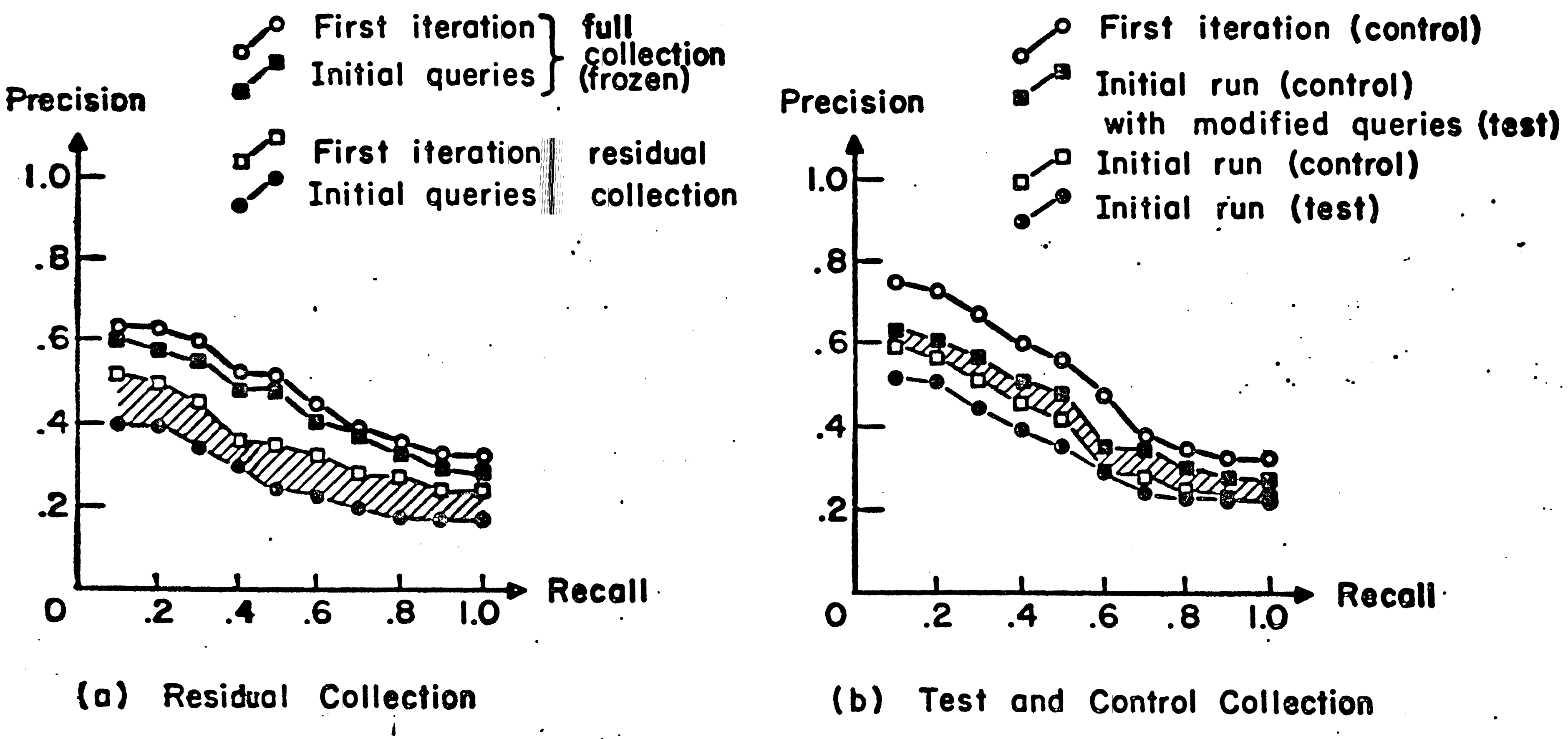
Alternative Feedback Evaluation

Table 9



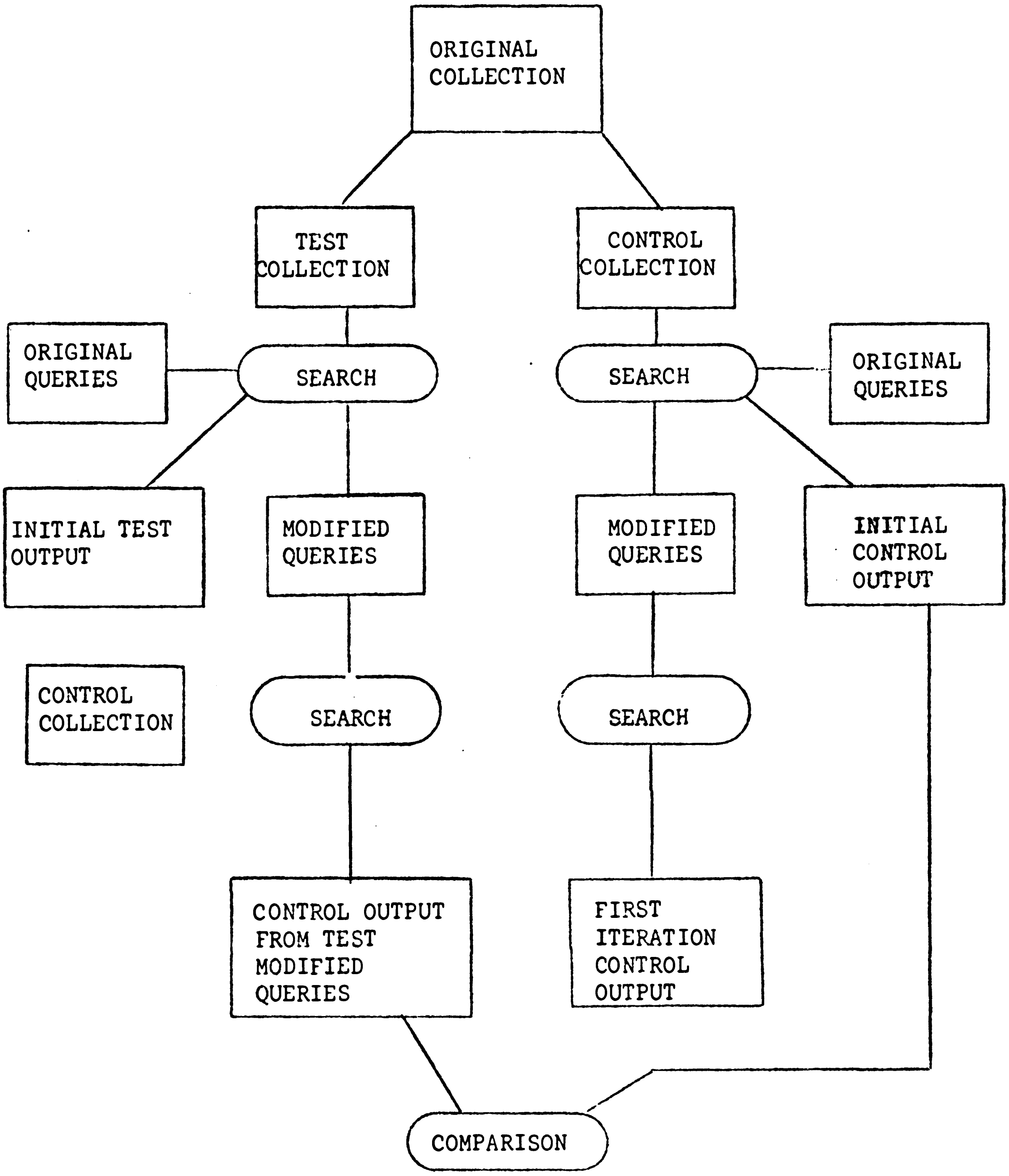
Comparison of Full with Clustered Search
(200 documents, 42 queries)
(from Dattola [9])

Fig. 1



Feedback Evaluation Graphs

Fig. 2



Test and Control Evaluation

Fig. 3