VII. Recommendations Based on Present and Prior Experiments

In this section, recommendations for practical interactive retrieval systems and for further research in relevance feedback are made. First, specific recommendations for operational interactive retrieval are drawn from the experimental results presented in the previous section. Then five general areas of concern are discussed and research problems are suggested using present and prior experiments as foundations for conjecture. These general areas are evaluation of relevance feedback performance, feedback of non-relevant documents, partial search strategies, and multiple query feedback strategies.

A. Relevance Feedback Recommendations for Concept Vector Document Classifications Systems

The findings of this study apply to retrieval systems that use positively weighted concept vectors to describe both documents and retrieval requests. Caution is necessary in generalizing these results to systems that differ from the SMART system in such aspects as the vector distance function used for retrieval, or the significance of vector position and weight magnitude. If the cosine correlation is used as the distance function, and if each vector position signifies a subject classification, and if the magnitude of a weight is in some way related to the importance of the corresponding subject in the document being classified, the

means of construction of the concept vectors should not affect the applicability of these results.

Because of the characteristics of the Cranfield 200 collection (Section IV), these results are most relevant to collections of article-length documents from limited technical subject areas, classified by information from the document abstracts. The following considerations are important when generalizing to document collections of realistic size.

a) The generality (ratio of number of relevant documents available to collection size) of a larger document collection would be lower. Lower generality would result in lower precision and less striking precision improvement [18].

b) The relationship of the number of documents provided for feedback to retrieval results would change as the relationship of this number to the collection size changes. Five documents, the number used most often for feedback in this study, constitute 2.5% of the experimental collection, equivalent to fifty documents in a collection of as few as two thousand documents. Therefore the results presented in Section VI-C must be interpreted with regard to the relative as well as the absolute magnitude of N.

c) The proportion of retrieved relevant to retrieved non-relevant documents might become smaller in a larger document collection, although this proportion probably would not maintain a constant relationship to the ratio of retrieved

documents to collection size. Comparison of feedback strategies using only relevant documents to those using non-relevant documents would be particularly affected by this proportion.

d) The number of queries available for the experimental collection is dangerously low from a statistical viewpoint. The subgroups results in Section VI-E divide a barely adequate query sample into even smaller groups. Although care has been taken to choose contrasting subgroups of near equal size, results of these experiments cannot be used for practical recommendations without verification in larger collections.

Because of the importance of these size considerations, experimentation with larger document collections is strongly recommended. The 1000 to 5000 document size is convenient for many reasons. First, the time and money needed for experiments would not be prohibitive. Second, a collection of that size could be found that would be useful to some professional or student group, so that actual users might be made available. Third, subject area clusters of this size would probably constitute a lower search level in a multi-level algorithm for large libraries, so the techniques found useful by experiment could be directly applied as subunits of such an algorithm.

Despite the limiting considerations listed above, some recommendations can be drawn from the data presented. First,

the general usefulness of the relevance feedback technique is supported. Comparison of a larger and more carefully chosen experimental document collection (Cranfield 200) to a smaller and less realistic one (ADI) encourages the generalization of these results to even larger collections by demonstrating that feedback improvement is maintained in spite of a lower ratio of relevant to non-relevant documents available. For a more definite confirmation of this finding performance in the Cranfield 200 collection should be compared to that in the full Cranfield collection, because the ADI and Cranfield 200 collections are not directly comparable in subject area, query construction, or document characteristics.

The demonstrated stability in the performance of algorithms using only relevant documents for various relative weightings of the original query and the retrieved documents also supports the general usefulness of the technique. None of the formulas used in this study for 'relevant only' strategies can be chosen as superior. This conclusion agrees with the results reported by Crawford and Melzer [12], who find no indication that the original query must be retained after the initial search.

Firm conclusions can be reached concerning the number of documents used for feedback with strategies using only relevant documents. The performance improvement caused by feeding back more documents is impressive up to five percent of the collection and still noticeable at seven and one-half percent of the collection. In a document collection of

useful size, input-output time and user effort would limit feedback to far less than five percent of the collection. Therefore the following algorithm for determining the number of documents to be used for feedback is recommended for larger collections on the basis of the results in Section VI-C.

At least n documents are initially retrieved for each user. If none of these n are judged relevant, more documents are retrieved until one relevant document is found or N documents have been retrieved. The numbers n and N are chosen considering cost, input-output time, and user effort in the particular retrieval system. From the results of this study a value of 5 or more is suggested for n, and a value less than or equal to five percent of the collection is recommended for N. This combination feedback algorithm should be tested with strategies that use non-relevant documents for feedback.

For queries retrieving no relevant documents within N documents, the Rocchio strategy (Section VI-D, reference 9) using non-relevant documents is recommended. In fact, the Rocchio strategy is often superior to strategies using relevant documents only even when relevant documents are available for feedback. In the experimental collection the Rocchio strategy is superior on 36% and equal on 32% of the queries that retrieve some but not all relevant documents on the first iteration. Nevertheless, because of the variability in negative feedback performance reported in Section VI-D, feedback of non-relevant documents cannot be

recommended as a general strategy. Possible causes of negative feedback variability are discussed in Section VII-C. The recommendation of the Rocchio strategy

for queries retrieving no relevant documents is supported by Steinbuhler and Aleta [13].

B. Evaluation of Relevance Feedback Experiments

The evaluation problems encountered in this study give rise to several suggestions for future experiments. Some of the recommendations made in this section are applicable only to the evaluation of interactive feedback techniques, but others are generally valid for information retrieval experiments.

The variability of the results reported in Section VI-D casts doubt on all comparisons of average values of retrieval performance measures, and demands tests of statistical significance for meaningful comparison of retrieval parameters. In Figure 17, a difference of 7% in normalized recall is not statistically significant, yet in Figure 19 a difference of 3.1% is found significant at the 0.6% level. Obviously it is dangerous to use the magnitudes of performance differences as the only indicators of significance in the experimental environment of this study. The same evidence supports the recommendation that larger query samples be obtained.

The apparent conflict between the normalized recall measure and the recall-precision curves for negative feedback is resolved by the document curves of Figure 20. This suggests that valuable information is lost by attempts to condense complex retrieval information into overall performance measures. Even the ten-point recall-precision curves do not

preserve the information contained in the document curves. The two-valued measure, normalized recall and precision, loses all indication of the superiority of the Rocchio strategy when less than 40% of the collection has been retrieved.

This situation presents a ~~grave~~ problem in evaluation, because available tests of statistical significance deal with single valued measures of performance. Determining the joint significance of more than one measure requires that the statistical dependence of each measure on the other be known. In information retrieval, all measures of performance are based on a single ranked list and thus cannot be assumed independent, yet the dependence of one measure on another is difficult to determine, and may vary in different experimental situations. For this reason no attempt is made in this study to estimate the joint significance of more than one performance measure. Since no single valued measure preserves the information most meaningful to these experiments, there is no way to determine the overall statistical significance of the differences between positive and negative feedback strategies.

In this complex experimental environment it is imperative that the experimenter have a clear conception of the questions he is asking, and that he choose performance measures that can answer his questions. A convincing example of this necessity occurs in Section VI-C of this study, where the tried-and-true recall-precision curves are found inappropriate as a measure of the effect of amount of

feedback on performance. Both in Section VI-C and Section VI-D the document curves are used to prevent misinterpretation of the more common measures. The evaluation problems mentioned stimulate thought in three areas; summary measures of performance, interpolation methods for recall-precision curves, and evaluation methods for interactive strategies. The suggestions made in these areas arise directly from consideration of the questions being asked by the experimenter and the questions being answered by the performance measure.

Although summary measures of performance such as normalized recall and precision lose information, they are nevertheless valuable for statistical evaluation. Since all information cannot be retained in a summary measure, a measure of the aspect of performance most relevant to the experiment should be chosen. The failure of normalized recall and precision to reflect the early retrieval advantage of the Rocchio strategy suggests that these measures are answering the wrong question. They sum the recall and precision at each possible cut-off point over the entire document collection, and weight each possible recall or precision value equally. From a practical standpoint however, early retrieval performance is more important than performance after most of the collection has been retrieved, especially when interactive iterative search algorithms are being tested. Normalized recall in particular seems intuitively inappropriate for this

study in that a change in rank from 195 to 191 has the same effect on normalized recall as a change from five to one. Yet the idea of summing recall or precision at all cut-off values is a sound basis for a summary measure of performance. Two alternate measures, called weighted recall and weighted precision, are suggested that preserve the summation idea but attach greater importance to earlier retrieval. Rocchio's normalized recall and precision are stated most simply by the following formulas:

$$NR = \frac{1}{N} \sum_{j=1}^{N} R_j$$

$$NP = \frac{1}{N} \sum_{j=1}^{N} P_j$$

where R_j and P_j are recall and precision at a cut-off of rank j. Weighted recall and precision give a weight of N to the recall and precision values at rank 1 and progressively smaller weights to later values, as indicated by the following formulas:

$$WR = \frac{2}{N(N+1)} \sum_{j=1}^{N} (N-j+1)\, R_j$$

$$WP = \frac{2}{N(N+1)} \sum_{j=1}^{N} (N-j+1)\, P_j$$

The multiplier $2/N(N+1)$ gives weighted recall and precision the same range as normalized recall and precision. Similar formulas can be constructed giving more or less relative

weight to earlier retrieval performance. In this way a range of two-valued summary performance measures can be provided from which an experimenter can select the measure that reflects his concerns.

The interpolation methods used for the recall-precision curves in this study have been supported or criticised in the past with regard to the 'meaning' of the average curve obtained. Statistical tests of the significance of precision differences at each level of recall treat each interpolated value as a measure of the performance of a single query, and may compare interpolated precision values to actual values achieved by other queries. Thus the meaning of the single interpolated value is the important factor in a choice of interpolation method, because each interpolation method defines a performance equivalence relation among queries with different numbers of relevant documents.

To make this point clearer, the example query of Figures 1 and 2 is used. This query, now called query A, has four relevant documents and retrieves them with ranks of 4, 6, 12, and 20. Suppose query B has eight relevant documents. What ranks are assigned to these eight documents by query B if it achieves performance equivalent to that of query A?

The rank of every other relevant document retrieved by query B is determined by the precision after each relevant document of query A is retrieved. That is, the second relevant document is retrieved by query B with rank 8, giving

precision of 2/8 (1/4). The fourth relevant document of query B has rank 12, the sixth rank 24, and the eighth rank 40. The ranks of the first, third, fifth, and seventh documents relevant to query B are determined by the interpolation method used, because for statistical comparison the precision after the first relevant document of query B is retrieved must be equivalent to the interpolated value for query A at 12.5%, the precision after the third relevant must be equivalent to the interpolated value for query A at 37.5%, and so forth.

Figure 33 gives the ranks of the eight relevant documents of query B that are defined as 'equivalent' to the ranks of the four relevant documents of query A: 4, 6, 12, and 20, by several interpolation methods including Quasi-Cleverdon and Neo-Cleverdon. Only exact integer ranks are assigned in the SMART system, but integer ranks equivalent to the ranks listed for Quasi-Cleverdon could occur if a query had enough relevant documents. Note the underlined rank of 6 given to the second relevant document by the Neo-Cleverdon interpolation. At this point the Neo-Cleverdon interpolation ignores the actual Query A precision at 25% recall and assigns a new precision value. This discarding of achieved recall levels is done by Neo-Cleverdon whenever the precision at a subsequent recall level is higher. The 'Lower Limit' interpolation represents the worst performance any query could achieve and still maintain the Query A precision values at 25%, 50%, 75%, and 100% recall. The 'Upper Limit' interpolation represents the

Query A: An example query with four relevant documents. For query A precision values at points other than 25%, 50%, 75%, and 100% recall must be interpolated.

Query B: A hypothetical query with eight relevant documents that achieves performance 'equivalent' to query A. In each column of the table, the ranks of the eight relevant documents of query B are set to give the same precision as the interpolated precision defined for query A by a given interpolation method.

Quasi-Cleverdon Neo-Cleverdon: Interpolation methods described by Figures 1 and 2.

Bottom Limit: An interpolation method based on the bottom limit of performance that a query could achieve and still have precision values equivalent to those at the uninterpolated points of the given query.

Top Limit: An interpolation method based on the top limit of performance a query could achieve and still have equivalent precision values at the uninterpolated points.

Equal Proportion: An interpolation method based on assigning an interpolated rank at each recall point such that the assigned rank and the adjacent uninterpolated ranks are related in the same proportion as are the recall points of interpolation and the adjacent achieved recall levels.

Recall Level	Query A Ranks	Equivalent Query B Ranks as Defined by:				
		Quasi-Cleverdon	Neo-Cleverdon	Lower Limit	Equal Proportion	Upper Limit
12.5%		4	3	8	4	1
25	4	8	6	8	8	8
37.5		10.3	9	12	10	8
50	6	12	12	12	12	12
62.5		17.3	20	24	18	12
75	12	24	24	24	24	24
87.5		31.5	35	40	32	24
100	20	40	40	40	40	40

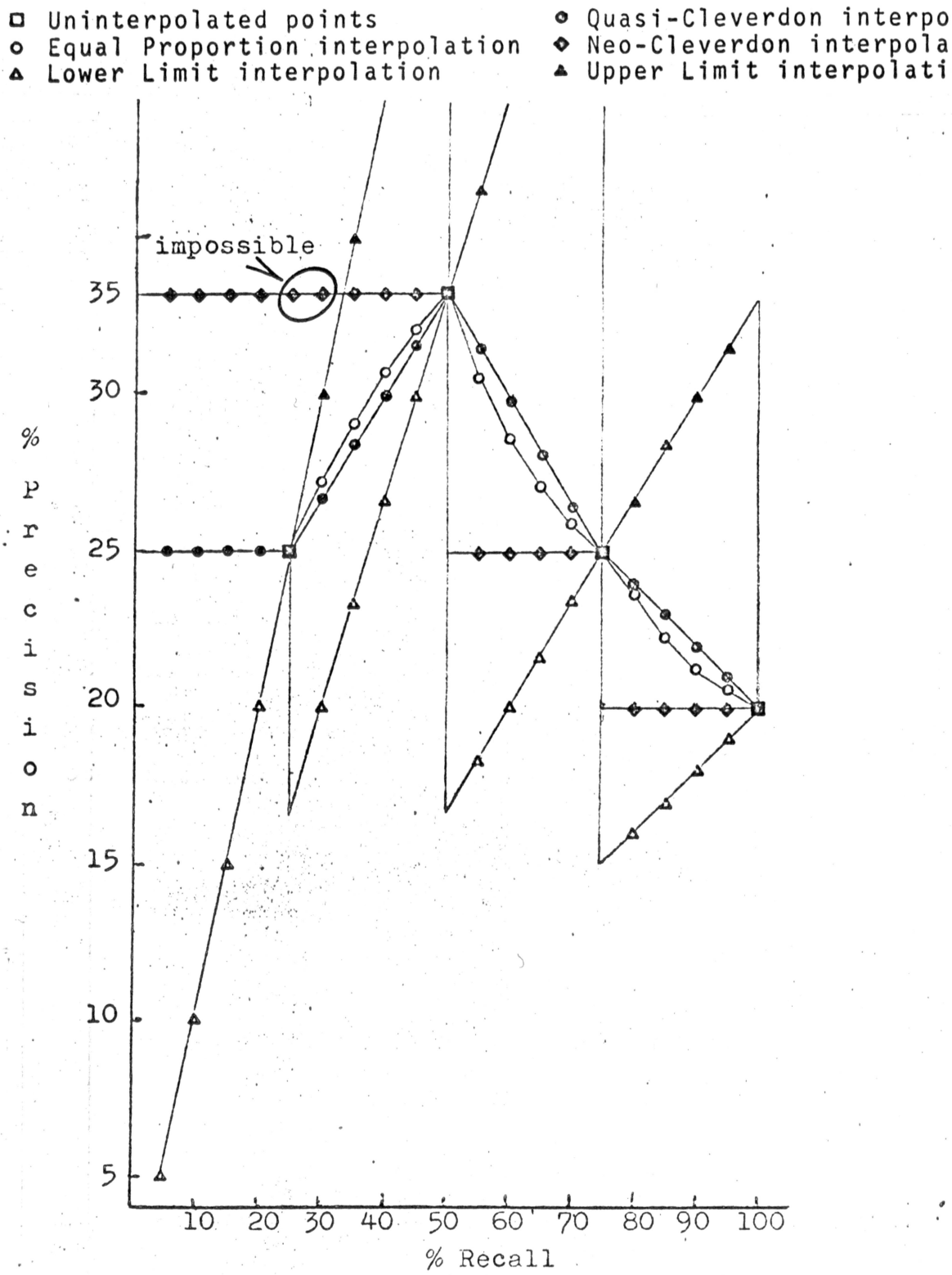
Examples of Performance Equivalence Between Queries As Defined By Different Interpolation Methods

Figure 33

best possible performance. The 'Equal Proportion' interpolation expresses the intuitively appealing idea that the relevant documents retrieved between the recall levels achieved by Query A should be ranked half-way between the adjacent well-defined ranks. Figure 33 shows that the ranks defined by Quasi-Cleverdon Interpolation are only slightly different than the Equal Proportion ranks. However, the Neo-Cleverdon interpolation is closer to the bottom limit after the highest precision point has been achieved and near the top limit before the high point of precision. The underlined rank of 6 is in fact above the top limit.

Figure 34 demonstrates all five interpolation methods in graphical form. The small squares on the graph represent the uninterpolated precision values achieved by Query A. All other figures represent the interpolation points at each five percent of recall defined by the five interpolation strategies described. It is evident that the Quasi-Cleverdon and Equal Proportion interpolations are almost identical. The Upper Limit interpolation is not graphed until 25% recall since it assigns 100% precision to all earlier points. Beyond 25% recall the Upper Limit and Lower Limit interpolations define quadrilaterals within which any precision value is possible to a query with equivalent precision at the uninterpolated points. The two circled points of the Neo-Cleverdon interpolation are outside the defined quadrilateral and thus are impossible.

Note that none of the interpolated curves bear any



Twenty-Point Interpolation From Example Query A

Using Five Different Interpolation Methods

Figure 34

resemblance to the sawtooth curve of Figures 1 and 2. The sawtooth curve represents the behavior of precision values in a single query between achieved recall levels. It is completely irrelevant to interpolation, because the interpolated values are statistically compared to the precision at achieved recall levels of other queries, not to precision between achieved levels. A comparison of Figure 1 and Figure 34 shows that many points on the sawtooth curve fall outside the range of possible interpolation points defined by the Lower Limit and Upper Limit interpolations.

A basic question arises from this discussion: 'What interpolation method provides the most appropriate definition of equivalent performance for queries having different numbers of relevant documents?' Figure 34 shows that the range of possibly equivalent interpolation points is great. One way of defining equivalence would be to pick the performance within the possible range that has the same probability of occurrence as the performance of the query for which interpolated values are sought. However, if it is assumed that integer ranks are assigned randomly without replacement, the Lower Limit interpolation curve in Figure 34 describes a performance level more probable than the performance of example Query A. The assumption of random assignment of ranks is inappropriate for information retrieval, because both the query vectors and the document vectors would have to be random. The controlling probability determinant for this study is the set of document

vectors, because it is unchanged from one experiment to the next. Thus an appropriate definition of equivalent performance would be performance equally probable in the given set of document vectors. This probability could be estimated from experimental results. For experiments that evaluate changes in the document space, the determinants of probable performance would be the constant factors in the experimental environment.

A rough estimate of an appropriate equivalence relation can be derived from the fact that normalized recall, normalized precision, and the document curves each provide a definition of equivalent performance. Equal performance for both recall and precision is defined by the document curves as performance equal at each cut-off rank, and by the normalized measures as performance giving an equal sum over all cut-off ranks. Since precision is defined as relevant retrieved divided by total retrieved, normalized precision equivalent to query A is provided by a query that retrieves a relevant document at ranks 4, 6, 12, and 20 and no other relevant documents, or by any query that has the same sum of precision at each cut-off value. Lower Limit interpolation provides a lower overall sum of precision values because the same precision levels are achieved at lower ranks. Therefore, the normalized precision definition of equivalence would give slightly higher interpolated values than does Lower Limit interpolation.

Recall, however, is defined as relevant retrieved divided by total relevant, so query B would have recall equiva-

lent to query A if it could somehow retrieve the first two relevant documents with rank 4, the second two with rank 6, the third two with rank 12, and the last two with rank 20. The normalized recall definition of equivalent performance demands n times the precision at each recall level for a query with n times the number of relevant documents. The Upper Limit interpolation gives lower values than this at all points.

To provide some estimate of a reasonable equivalence relation for the experimental environment of this study, the relationship of number of relevant documents to initial normalized recall and precision are presented. Spearman's coefficient of rank correlation [21] is positive for precision (.25) and slightly negative for recall (-.017). If either normalized precision or normalized recall provided a valid definition of equivalent performance for this query and document collection, the number of relevant documents would show no correlation with that measure. Therefore a definition of equivalence that coincides with the performance observed in this environment would be somewhere between the definition implied by normalized precision and that implied by normalized recall, but closer to that of normalized recall. That is, an appropriate interpolation method would be closer to Upper Limit interpolation than to Lower Limit interpolation. This conclusion contradicts the opinion expressed by the proponents of Neo-Cleverdon interpolation that the Quasi-Cleverdon method gives artificially high results. The rank correlations of initial

normalized recall and precision to the number of relevant documents indicate that the Quasi-Cleverdon interpolation may be conservative in the environment of these experiments.

Three considerations mentioned in the foregoing discussion support the recommendation that Quasi-Cleverdon interpolation rather than Neo-Cleverdon interpolation be used for investigations in query and document collections similar to the Cranfield data. First, the Neo-Cleverdon interpolation supplies data points that could not occur in a query with precision at uninterpolated recall levels equal to that of the query being represented. Since interpolated data points are statistically compared to achieved data points of other queries, ignoring some of the achieved data points of a query is inappropriate. Second, Quasi-Cleverdon interpolation gives results similar to an intuitively pleasing method (Equal Proportion) that assigns an interpolated rank half-way between the ranks a query with comparable precision at uninterpolated data points could achieve. Third, data supports the conclusion that the Quasi-Cleverdon interpolation does not give interpolated points that are artificially high in this experimental environment. Further investigation of the relationship of retrieval performance to the number of relevant documents should be conducted to support the choice of an interpolation method that provides a meaningful definition of equivalent performance for different queries. Such an interpolation method could lead to more general and more meaningful use of recall-precision curves as measures of retrieval performance.

When Hall and Weiderman [17] propose feedback effect evaluation, they are saying that total performance measures do not answer the question that is most relevant to relevance feedback experiments. Consideration of the questions the experimenter wishes to ask leads to the construction of several evaluation methods appropriate for relevance feedback, one of which is also useful in evaluating other strategies that require partial searches of the document collection.

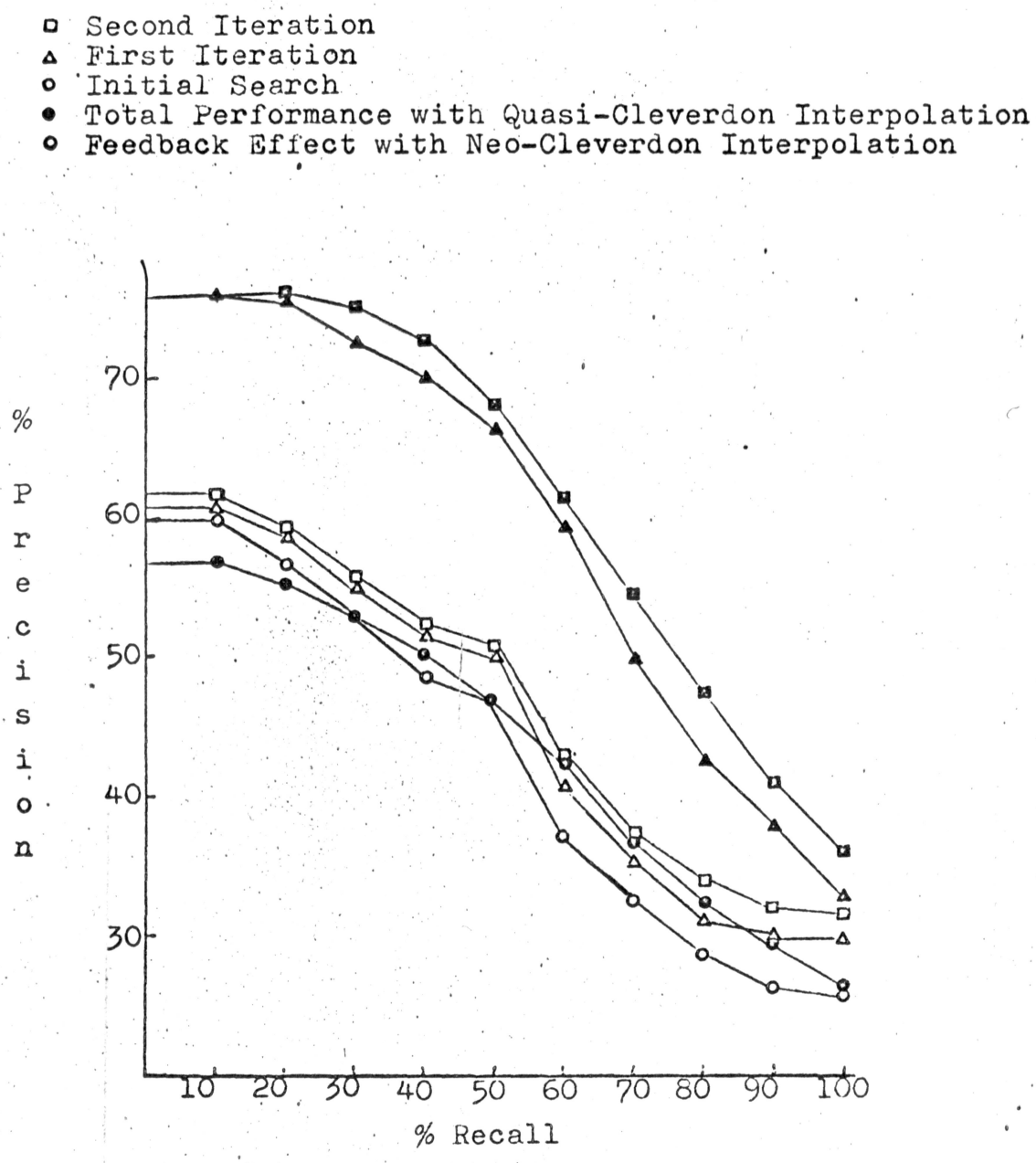
Total performance, when evaluating an interative strategy, answers the question 'How much closer is the modified query vector to the optimum query vector (Section III, Reference 9)? Hall and Weiderman state that "For a relevance feedback system the measure of its effectiveness should be a measure of how many new relevant documents are retrieved as a result of feedback." [17] However, feedback effect evaluation does not measure the variable that Hall and Weiderman propose. Instead, it answers the question 'What is the overall retrieval performance of the system after each iteration from the viewpoint of the user who is interacting with the system?'

The distinction being made above is based on the components of performance that are isolated for measurement. To measure how many new relevant documents are retrieved by feedback, the change in performance caused by the feedback on each iteration must be isolated from <u>all</u> other factors. In feedback effect evaluation, the early retrieval of previous iterations becomes an albatross which

is hung on each new iteration, so that the possibility of change in reported performance becomes less for each iteration. The description of feedback effect in Section V-C makes this point clear.

Before further discussion of the question that Hall and Weiderman ask, the question that Feedback Effect evaluation answers is explored. Figure 35 shows total performance evaluation with Quasi-Cleverdon interpolation and Feedback Effect evaluation with Neo-Cleverdon interpolation for two comparable strategies (Total Performance Q_0 and Feedback Effect Q_0+) with N equal to 5. The difference between the initial search curves is entirely due to the difference in interpolation methods.

The first iteration total performance curve shows that for the average information request the modified query vector is much closer to the optimum query vector than is the original query vector. The change in total performance from first iteration to second iteration indicates much less change in the query vector was caused by relevant documents retrieved on the first iteration than was caused by relevant documents retrieved on the initial search. This smaller change is due in part to the fact that total performance evaluation does not retrieve five <u>new</u> documents for the second iteration query modification. The five documents retrieved on the initial search are new, but the five retrieved on the first iteration probably include all relevant documents retrieved on the initial search.



Comparison of Two Evaluation Methods
Total Performance Evaluation with Quasi-Cleverdon Interpolation
and
Feedback Effect Evaluation with Neo-Cleverdon Interpolation
Comparable Strategies, N = 5

Figure 35

The feedback effect curves show much less change than the total performance curves after the initial search, demonstrating that the user interacting with the feedback system observes little of the effect of the change in the position of the query vector. The largest changes in the feedback effect curves are observed at high recall levels, because the freezing of the early retrieval limits possible gains in precision at low recall levels. The slight improvement observed at low recall levels is probably due to the leftward extension of later precision improvements by Neo-Cleverdon interpolation.

The comparison of total performance curves to Feedback Effect curves shows that great improvement in the position of the query vector is needed before the user at the teletype notices an overall improvement in interactive retrieval. This conclusion is significant in predicting the psychological impact of automatic interactive retrieval on its users.

Two methods of evaluation answer two valid questions about interactive retrieval systems. Yet other questions can be asked, and other evaluation methods can be constructed to answer them. Four examples of possible evaluation methods are presented below, one of which answers the Hall-Weiderman question 'How many new relevant documents are retrieved as a result of feedback?'

The total performance evaluation method is inappropriate for relevance feedback because it does not ensure that the

documents used for feedback have not been encountered previously. This fault in the evaluation method does not invalidate the question it is intended to answer: 'How much closer is the modified query to the optimum query?' The total performance algorithm is here modified to answer this question for more than one iteration of relevance feedback. The modified algorithm flags all documents presented to the user for feedback, and presents N new documents on each iteration regardless of the rank of the N^{th} new document in the list ranked by total performance. The recall-precision curves resulting from this algorithm could be directly compared to those generated by the total performance algorithm in experiments not involving document feedback. However, the document curves are changed in meaning because different queries would assign different ranks to the N^{th} new relevant document. Nevertheless, the modified total performance algorithm is needed to determine the performance increment caused by feedback iterations after the first. Both total performance and feedback effect evaluation limit the attainable performance of later iterations. Both evaluation methods therefore indicate a sharp drop in performance improvement after the first iteration, and both indicate so little improvement between the second and third iterations that third iteration results are not reported in this study. Modified total performance evaluation might show subsequent feedback iterations to be nearly as valuable as the first in moving the modified query toward the optimum query, and thus might

stimulate further study of later iterations of relevance feedback.

Two evaluation methods similar to feedback effect evaluation have been discussed, both of which indicate better performance after feedback than feedback effect evaluation does. One of these methods assigns all previously retrieved relevant documents the highest possible ranks and all previously retrieved non-relevant documents the lowest possible ranks. This algorithm is here called 'best list' because it answers the question 'Using all information available to the system, what is the best ranked list of documents that can be presented to the user after the iteration being evaluated has made a search?' This algorithm would report better performance for iterations after the first than any evaluation method discussed earlier in this report, because it maximizes ranking effect for each query. However, the impressive performance changes would not be informative, because most of the improvement reported by best list evaluation would not be caused by the changes made to the query vector as a result of feedback.

A better alternative that retains an outlook important to the user is here called 'modified feedback effect' evaluation. Feedback effect freezes the ranks of all documents presented to the user on earlier feedback iterations, and assigns the first document retrieved on the ith iteration a rank of $iN+1$, if N documents are used for feedback on each iteration. Modified feedback effect freezes the

ranks of all previously retrieved relevant documents, and assigns the first document retrieved on the ith iteration the rank next below that of the last retrieved relevant document. Non-relevant documents retrieved with ranks higher than that of the last retrieved relevant document retain their earlier ranks, and non-relevant documents retrieved with lower ranks are re-ranked by the modified query. Like modified total performance evaluation, modified feedback effect retrieves N new documents on each iteration regardless of the rank of the N^{th}. It answers the question 'What is the best performance that can be achieved this iteration given the performance indicated by (i.e. without changing the ranks of) the relevant documents already seen by the user?' Hall and Weiderman define ranking effect as "changes in the rank of relevant documents previously seen by the user" (underscore mine) [17]. By this definition, modified feedback effect evaluation is the appropriate measure of feedback effect. Since non-relevant documents retrieved below the last retrieved relevant document are re-ranked rather than being pushed to the bottom of the list, all performance improvement between iterations can be attributed to changes in the query.

Feedback effect evaluation and modified feedback effect evaluation have a common characteristic; performance on each feedback iteration is limited by the early retrieval performance already achieved. Thus neither way of measuring 'feedback effect' directly answers the Hall-Weiderman question 'How many new relevant documents are

retrieved as a result of feedback?' Rephrased in terms of overall performance so that 'retrieved' need not be defined, this question is 'What is the performance of the modified query with respect to the relevant documents that have not yet been presented to the user?' By Rocchio's theory [9], this is equivalent to the question 'How close is the modified query to the optimum query for the documents not yet presented to the user?' Therefore, to answer Hall and Weiderman's question the evaluation method must treat the remainder of the document collection as a complete collection and the remainder of the relevant documents as a complete set of relevant documents, and perform a total performance evaluation of the modified query in this new environment. The evaluation method constructed to answer the three equivalent questions posed above is called 'residual collection' evaluation.

Three problems are encountered in the construction of a residual collection evaluation method. The first and most obvious problem occurs when all relevant documents are retrieved before all requested iterations are completed. This problem is solved by dropping all queries that retrieve all relevant documents from the query sample for later iterations, and reporting the number of queries remaining in the sample for each iteration. This solution has the advantage of eliminating all meaningless information from the evaluation of each iteration to give an unbiased indication of the improvement obtained as a result of feedback. Of course, a user in a real environ-

ment might conduct fruitless searches, not knowing that all relevant documents available had been retrieved. Residual collection evaluation ignores this user's dilemma because there is no unambiguous way to take account of it in the experimental environment.

The second problem occurs in averaging the performance of different queries. Each query may have a different residual collection for a given iteration. This poses no problem unless the number of documents used for feedback is not the same for all queries, in which case the residual collections are not the same size. The variable feedback situation does not change the meaning of normalized measures or of recall-precision curves as long as the appropriate collection size is used for averaging. However, two possible methods of document curve construction exist. Recall and precision could be averaged after an absolute number of documents had been retrieved, or at percentiles of the document collection. Since recall and precision values change rapidly at the higher ranks and since all queries would be averaged into the earliest retrieval points, the absolute number of documents retrieved is an appropriate evaluation dimension for early retrieval. Percentile of the collection is a better evaluation dimension for document curves intended to summarize overall performance.

The third problem involves comparison of results between iterations. For two retrieval algorithms, performance measures obtained by residual collection evaluation

could be directly compared for each feedback iteration. However, if one iteration of a feedback strategy is compared to a previous iteration of the same strategy, the question asked of the comparison must be specified. Direct comparison is appropriate if the question asked is of this type: 'How much more improvement occurred as a result of first iteration feedback than occurred as a result of second iteration feedback?' This is a meaningful question that cannot be asked of other evaluation methods. However, a quite different type of question is often asked: 'Would it be better for the user to perform a second feedback iteration than to look at the later retrieval of the first iteration?' The latter question is not answered by direct comparison of residual collection measures, because it is equivalent to the question 'Is the second iteration query closer to the optimum query <u>for the second iteration residual collection</u> than the first iteration query is?' Thus residual collection evaluation must provide the option of re-evaluating the performance of queries used for previous searches in the residual collection constructed for a later search. This re-evaluation is not difficult if the ranks assigned to relevant documents by earlier iterations are saved. To calculate the performance of the first iteration query in the second iteration residual collection, for example, all relevant documents presented for feedback on the second iteration are deleted from the saved list (or not saved) and the lowest rank assigned by the first

iteration to a document presented for second iteration feedback is subtracted from each rank assigned to a relevant document by the first iteration. (This 'lowest rank' is the number of documents fed back on the second iteration.) The adjusted ranks of relevant documents are used to calculate all measures and the size of the second iteration residual collection is used for averaging.

In spite of the greater complexity of calculation, residual collection evaluation is recommended for future experiments with relevance feedback, because it directly answers a hitherto uninvestigated question considered most relevant to the evaluation of feedback strategies. Moreover, the viewpoint presented by residual collection evaluation is appropriate to other areas of information retrieval research. Some of these areas are discussed in later sections of this report.

An evaluation method has been proposed that avoids the controversy between feedback and ranking effect. The document collection is randomly separated into two halves here called subset one and subset two. The feedback and query alteration are performed based on subset one, then the original query and all altered queries are tested on the documents of subset two. Subset two thus performs the function of a residual collection not containing documents used for feedback. This evaluation method, here called test collection evaluation, has the same advantages as residual collection evaluation. It shares two residual

collection evaluation disadvantages. First, both methods require that previous queries be re-evaluated in the residual or test collection. Of course, residual collection evaluation provides several test collections while test collection evaluation supplies only one. Second, both methods may encounter queries with no relevant documents in the residual or test collection and that therefore must be dropped from the evaluation. This condition is less likely in residual collection evaluation because the residual collection is as large as possible. Test collection evaluation has one advantage and two disadvantages as compared to residual collection evaluation. It has the advantage of using the same collection to test all queries, while residual collection evaluation uses a different residual collection for each query, causing the problems stated earlier. On the other hand, residual collection evaluation has the advantage of providing the largest possible collection for test purposes in every case. Also, residual collection answers directly a question not answered by test collection evaluation, which is the Hall and Weiderman question 'How many new relevant documents are retrieved as a result of feedback'. This question requires the use of different test collections for each query, because the 'new relevant documents' available to each query in the document collection are different.

Test collection evaluation thus provides a method of evaluation distinct from residual collection evaluation and having several advantages over the evaluation methods previously employed. Its major disadvantage is the need to halve the size of the experimental collection. However, test collection evaluation promises to be a useful technique for providing direct comparison between varied feedback strategies, search techniques, vector constructions, and other dissimilar experiments conducted over a long period and using the same large document collection divided into the same subsets. It parallels the commonly accepted procedure of providing a control group and a test group for each experiment, and should be added to the evaluation methods employed in information retrieval as soon as a large enough collection is obtained.

From consideration of the problems encountered in evaluating the experiments reported, five recommendations for evaluation of relevance feedback algorithms are made. First, larger document collections with larger query samples should be obtained and statistical tests should be used to support all average results. Second, weighted recall and precision, summary measures analogous to normalized recall and precision, are recommended to attach greater significance to early retrieval than to later retrieval. Third, Quasi-Cleverdon interpolation is recommended over Neo-Cleverdon interpolation for constructing average recall-precision curves in the experimental environment of this

study, and further investigation of possible definitions of 'equivalent performance' for queries having different numbers of relevant documents is suggested. Fourth, three new evaluation methods are constructed that are more appropriate for relevance feedback study than existing methods. The three methods are called modified total performance evaluation, modified feedback effect evaluation, and residual collection evaluation. Each answers a different question that is relevant to the study of interactive document feedback. Fifth, a previously suggested evaluation method, here called test collection evaluation, is distinguished from residual collection evaluation and is recommended to provide directly comparable studies of different types of retrieval and classification methods.

C. Feedback of Non-Relevant Documents

The results reported in Section VI-D and VI-E indicate that feedback of non-relevant documents provides excellent retrieval for certain queries and very poor retrieval for certain others. Although the causes of this variability are not clear from this study, promising indications for further research are found in Section VI-E. Similar investigation of subgroup properties should be conducted in larger document collections with larger query samples, because the sizes of some subgroups investigated are marginally small for the statistical test used, especially when tied observations occur. With a larger query sample, comparisons within subgroups as well as between subgroups

would be meaningful. Four research areas suggested by results reported in Section VI-E are listed below.

1) Three findings indicate that for the Rocchio strategy, movement of the modified query away from the original

query between the first and second iterations is correlated with poor performance, especially with poor initial search results. This direction of movement could be an effect of inadequate feedback, or it could be an attempt to compensate for a poor original query. Residual collection evaluation could determine whether the movement of the second iteration Rocchio query further from the original query results in performance improvement or performance degradation, and modified total performance evaluation could determine whether the movement is toward or away from the optimum query for all relevant documents. A difference in the results of these two evaluation methods would raise implications for multiple query strategies (discussed in Section VII-D).

2) Recall-precision curves (but not normalized measures) indicate that the queries resulting in performance degradation with the Rocchio strategy give poorer performance on the initial search and poorer performance and less first iteration improvement with the Q_0+ strategy than other queries. If this relationship holds in other collections, this type of query should be studied separately to discover the causes of this poor performance. It is possible that the relevant documents for these queries form two or more separated clusters in the document space. The implications of this possibility are discussed shortly.

3) Further study of those queries that retrieve no relevant documents on the initial search should be conducted in an environment containing more such queries. The

ingenuity of Steinbuhler and Aleta [13] in artifically creating the same retrieval situation by omitting retrieved relevant documents from the collection leads to valid conclusions about negative feedback, but does not provide a valid means of investigating the type of query that results in poor initial retrieval.

4) Finally, the joint relationship of number of concepts and number of available relevant documents to the performance of positive and negative feedback strategies should be explored in several ways. Some relevant questions are:

Does the relationship found in the Cranfield 200 collection hold in other environments? Does it hold for residual collection evaluation?

Is query vector length a better predictor than number of concepts? If so, is the reported relationship caused by the failure of this study to normalize the components of the Rocchio query modification? Does the change in query vector length after feedback have some relationship to the reported phenomenon?

Can the number of documents retrieved on the initial search be used as an estimator of the number of available relevant documents? If so, can the system select an appropriate strategy for each query before iteration? If not, can another estimator be found that is known to the system before iteration?

Do the queries with many concepts and few relevant have similarities to the queries with few concepts and

many relevant other than that of poor performance on the Rocchio strategy? Do these two groups differ in characteristics other than number of concepts and number of relevant? Does the Rocchio strategy fail for the same reason or for a different reason in each group?

A hypothesis is presented that explains some of the observed performance differences between the negative feedback strategies and the positive feedback strategies investigated, and is consistent with all experimental results reported.

Hypothesis:

For most queries, for every vector v contained in the set R of relevant document vectors there exists at least one vector s contained in the set S of non-relevant document vectors such that for some other vector r contained in R, cos(r,s) is greater than cos(v,r). Further, for a significant number of queries the prevalence of such relationships effectively prevents the retrieval of some relevant documents with reasonable precision by any relevance feedback strategy that constructs only one query on each iteration.

This hypothesis states in effect that the documents relevant to a single query are usually found in two or more distinct clusters in the concept vector space, and that these clusters of relevant documents are separated from each other by non-relevant documents. Further, it states that for a significant number of queries this phenomenon will seriously interfere with the retrieval of some relevant documents regardless of the relevance feedback strategy employed. For any collection in which this hypothesis is true, all relevance feedback algorithms tested in this study are inappropriate for a significant percentage of retrieval requests. Algorithms constructing more than one query on each feedback iteration are

necessary in such an environment.

The anomalous results of the reported comparisons of positive and negative feedback support the conclusion that the stated hypothesis is true in the Cranfield 200 collection. Because this collection is a carefully chosen subset of a larger collection representative of a well-defined, technical, limited subject area, this conclusion suggests that multiple query algorithms or other means of simplifying the distribution of relevant document vectors in the vector set being searched will be needed in practical automatic retrieval systems.

The most ubiquitous indication of separated relevant clusters is the typical negative feedback drop in normalized recall on the first iteration. This decrease in normalized recall is coupled with a rise in total performance normalized precision and in both total performance and feedback effect precision at all recall levels. As was stated earlier, this combination of measurements indicates that the Rocchio strategy raises the ranks of some high-ranking relevant documents and lowers the ranks of other low-ranking relevant documents. In fact, figure 21 shows that both negative feedback strategies tested are superior to positive feedback within the top 8% of the ranked collection, but greatly inferior in recall after 20% of the collection has been scanned. Both negative feedback strategies maintain a slight first iteration advantage in precision. The early negative feedback advantage is evident in spite of the freezing of the top ranks for feedback effect

evaluation. Therefore it is evident that the ranks of high-ranking unretrieved relevant documents are being raised more by negative feedback than by positive feedback, but that the ranks of low-ranking relevant documents are being lowered much more by negative feedback to cause a precipitous drop in the average recall difference between negative and positive feedback. Rephrasing the previous sentence in terms of query vector movement, the use of nearby non-relevant documents as well as nearby relevant documents for feedback causes the query to move closer to other nearby relevant documents than to nearby non-relevant documents, but at the same time to move farther from relevant documents already relatively distant than from relatively distant non-relevant documents. Such a description of vector position change is easiest to explain by assuring the presence of non-relevant documents between the 'nearby' and 'distant' groups of relevant documents. In particular, Figure 21 might indicate that the non-relevant documents used for feedgack are between the retrieved relevant documents and the 'distant' relevant documents, and actively push the modified query away from low-ranking relevant documents.

Several characteristics of the groups of queries chosen by strategy in Section VI-E are consistent with the hypothesis of separated relevant clusters. The criterion for selection of the Q_0^+ and Rocchio groups is retrieved within i N documents on the i 'th iteration, i ranging from one to three. The differences in normalized recall and precision between these

groups are caused by the Rocchio strategy. The normalized measures for the Q_o^+ strategy are not significantly different between the Q_o^+ group and the Rocchio group, but in the Q_o^+ group the Rocchio strategy degrades performance and in the Rocchio group it improves performance. In addition the recall-precision curves of Figure 28 and 29 show that the initial search curve of the queries in neither group is the highest, while the initial search curve of the Q_o^+ group is the lowest at low and medium recall levels.

The findings summarized above can be explained in terms of the hypothesis as follows: The strategy differences are caused by the Rocchio strategy because it uses negative feedback to discriminate better between the retrieved relevant and retrieved non-relevant documents. If the retrieved non-relevant documents are badly positioned relative to some unretrieved relevant documents, the Rocchio strategy specifically moves the query away from these relevant documents while the Q_o^+ strategy merely moves toward retrieved relevant documents. Because the Rocchio strategy discriminates better between relevant clusters represented by feedback, it can have inferior <u>retrieval</u> only if the Rocchio query is pushed away from all relevant documents by negative feedback (only 3 cases) or if it moves away from many relevant documents in order to better discriminate between a relatively small relevant cluster and nearby non-relevant documents. Since both strategies use the same relevant documents for feedback on the first iteration, only the hypothesis of separated relevant clusters can explain how negative feedback moves the query further away

from relevant documents than positive feedback. Since the described movement of the Rocchio query cannot occur if the origin query retrieves relevant documents that represent the largest relevant clusters, the Q_o^+ group has poor initial search performance. The Rocchio strategy has better early retrieval if the original query retrieves documents representing the largest clusters without retrieving the entire cluster, that is, if the original query is good but not optimal. If the original query is already near-optimal, both strategies will have equally good performance. This reasoning explains the high average performance of queries in neither group at all recall levels. The Rocchio group and Q_o^+ group are equally low in initial precision at high recall, indicating that in both groups the original query is far from some separated clusters of relevant documents. Also, in Figure 30 the Rocchio strategy in the Rocchio group has lower normalized recall on the first iteration than the Q_o^+ strategy in the Rocchio group, indicating that even when the Rocchio strategy provides better early retrieval, it still lowers the ranks of distant relevant documents relative to the ranks assigned by the Q_o^+ strategy to these documents. If no queries in the Rocchio group had separated clusters of relevant documents, the higher early retrieval of the Rocchio strategy would lead to higher normalized recall also.

The first statement of the hypothesis is thus consistent with reported results. The stronger statement that the presence of separated clusters of relevant documents will prevent full

retrieval for a significant number of queries with any single-query feedback strategy is supported by the low average precision at 100% recall. The highest reported total performance average precision at full recall is 45%, and the highest feedback effect average precision is 33%. To further support this stronger claim, the performance of the individual queries for the Q_0^+ and Rocchio strategies are examined. Twenty-eight of the 42 queries display performance indicating the presence of separated relevant clusters; that is, as the correlation of one relevant document rises, the rank of another relevant document falls. Twenty-two of these queries display this behavior with the Q_0^+ strategy, proving that the phenomenon is not caused only by negative feedback. Eighteen queries seem seriously affected by the presence of separated relevant clusters. For 12 of these queries, one or more relevant documents are not retrieved within 20% of the collection, or 40 documents, by either positive or negative feedback after three feedback iterations. The average precision at 100% recall for these queries is 7.4% when the best strategy is chosen for each query. Six more queries have at best less than 20% precision at full recall and must search at least 10% of the document collection. The average precision of these six queries at full recall is 16% when the best strategy is used for each query. The average rank of the last relevant document retrieved by these queries is 73.5 at best. By contrast, the average precision at full recall of the remaining 24 queries is 52.7% and the average rank of

the last relevant document is 10.4, at best. When the worst strategy is chosen for each query the average final precision only drops to 45.2%. The conclusion that either relevance feedback strategy is inappropriate for 43% of the query sample is inescapable.

Examples of the retrieval behavior caused by separated clusters of relevant documents are given in Figures 36, 37, and 38. Query 9 has only two relevant documents, but these are separated from each other so that as one rises in rank, the other falls. Positive feedback retrieves one of these relevant documents and negative feedback retrieves the other. Figure 37 gives a more complex example. The Q_0^+ strategy uses only document 173 for feedback, thereby raising the ranks of five relevant documents and lowering that of document 174. The second Q_0^+ iteration provides no feedback, so the original query is increased in weight, lowering the ranks of four relevant documents and raising document 174. Feedback of document 172 raises three of the lowered relevant documents, further lowers document 176, and lowers document 174 again. The movement of the Q_0^+ query vector is not consistent in direction, and little overall improvement in performance is accomplished. Negative feedback achieves better early retrieval by retrieving document 176 on the second iteration. All unretrieved relevant documents except the obviously separated document 174 rise in rank after the first and second iterations. However, after retrieval of 141 and 172 the ranks of 171 and 175 are lowered and that of 174 is raised slightly. In Figure 38 the Rocchio query moves immediately to a cluster of relevant documents including 4, 30,

and 32, using only negative feedback. Document 57 drops slightly in rank and document 31 drops considerably. Retrieval of document 57 by positive feedback raises document 31, but is much less effective than negative feedback in raising 4, 30, and 32. Feedback of documents 4, 57, 30, and 32 to the Rocchio strategy is needed to raise the ranks of documents 31 and 33 at the same time; in two other cases the ranks of these two documents change in opposite directions.

The inconsistent changes in rank from one iteration to the next displayed in these three figures are typical, and indicate that neither the Rocchio nor the Q_o^+ strategy is optimal in the experimental collection.

Query 9:

Rank	Q_o^+ Strategy Iteration			Rocchio Strategy Iteration		
	0	1	2	0	1	2
1	179	179	179	179	179	179
2	112	112	112	112	112	112
3	39	39	39	39	39	39
4	42	42	42	42	42	42
5	181	181	181	181	181	181
6	45	45	45	45	25	25
7	62	62	62	62	71	71
8	116R	116R	116R	116R	41	41
9	97	97	97	97	64	64
10	188	188	188	188	3	3
11	31	31	117	31	85	98
12	57	57	3	57	88	178
13	117	117	2	117	23	82R
14	2	2	158	2	101	160
15	25	25	185	25	17	101
33	82R	82R	0	16 0	82R	0
42	0	0	0	18 0	116R	0
47	0	0	82R	20 0	0	116R
				21 0	0	0
				33 82R	0	0

An Example of an Individual Query
With Separate Clusters of Relevant Documents
Q_o^+ and Rocchio Strategies
Figure 36

Query 30: Q_o^+ Strategy					Rocchio Strategy				
Rank	Iteration				Rank	Iteration			
	0	1	2	3		0	1	2	3
1	39	39	39	39	1	39	39	39	39
2	173R	173R	173R	173R	2	173R	173R	173R	173R
3	188	188	188	188	3	188	188	188	188
4	42	42	42	42	4	42	42	42	42
5	7	7	7	7	5	7	7	7	7
6	199	156	156	156	6	199	176R	176R	176R
7	41	41	41	41	7	41	27	27	27
8	23	44	44	44	8	23	97	97	97
9	30	199	199	199	9	30	156	156	156
10	156	23	23	23	10	156	101	101	101
11	178	176R	30	30	11	178	49	96	96
12	181	101	178	178	12	181	134	141R	141R
13	44	118	101	101	13	44	96	44	44
14	131	27	181	181	14	131	31	89	89
15	73	172R	172R	172R	15	73	118	118	118
19	0	0	176R	0	19	0	0	172R	172R
23	172R	0	0	0	20	0	172R	0	0
25	174R	0	0	0	22	0	141R	0	0
27	0	0	0	176R	23	172R	0	0	0
28	0	141R	0	0	25	174R	0	0	0
30	0	0	174R	0	54	0	174R	0	0
32	0	0	0	141R	67	0	0	171R	0
35	0	0	0	171R	69	0	0	0	171R
39	0	174R	141R	0	71	176R	0	0	0
69	0	171R	0	0	103	0	171R	0	0
71	176R	0	0	0	111	0	0	0	174R
77	0	0	0	174R	112	0	0	174R	0
80	0	0	171R	0	115	0	0	175R	0
109	0	0	0	175R	121	0	0	0	175R
118	0	175R	0	0	130	0	175R	0	0
123	0	0	175R	0	198	141R	0	0	0
198	141R	0	0	0	199	171R	0	0	0
199	171R	0	0	0	200	175R	0	0	0
200	175R	0	0	0					

An Example of Complex Retrieval Behavior

Figure 37

Query 3: Rank	Q_o^+ Strategy Iteration			Rank	Rocchio Strategy Iteration		
	0	1	2		0	1	2
1	179	179	179	1	179	179	179
2	42	42	42	2	42	42	42
3	112	112	112	3	112	112	112
4	39	39	39	4	39	39	39
5	117	117	117	5	117	117	117
6	181	181	181	6	181	4R	4R
7	57R	45	45	7	57R	71	71
8	45	57R	57R	8	45	57R	57R
9	152	152	152	9	152	30R	30R
10	62	62	62	10	62	32R	32R
11	182	182	31R	11	182	182	31R
12	153	153	4R	12	153	152	200
13	31R	31R	182	13	31R	43	189
14	43	43	30R	14	43	3	184
15	116	116	189	15	116	199	34
17	0	0	0	20	30R	0	0
20	30R	30R	32R	23	32R	0	0
23	32R	32R	0	25	4R	0	0
25	4R	4R	0	27	0	31R	0
124	33R	33R	0	36	0	0	33R
181	0	0	33R	85	0	0	0
195	0	0	0	118	0	33R	0
				124	33R	0	0

An Example of Good Rocchio Performance
On Separate Clusters of Relevant Documents

Figure 38

In summary, four areas of future research are recommended involving feedback of non-relevant documents. Queries retrieving no relevant documents on the first iteration should be studied, the relationship between the correlation of the modified query to the original query and performance should be determined, and the joint relationship of query size and number of relevant documents to positive and negative feedback differences should be explored. A hypothesis explaining the observed performance differences between positive and negative feedback is presented, and evidence of its validity is found in the reported results. Many queries have separated clusters of relevant document vectors, and are modified by both positive and negative feedback algorithms in such a way as to make early retrieval of some relevant documents impossible. The conclusion that all strategies tested in this study are inappropriate to this retrieval environment because of the prevalence of queries having separated clusters of relevant documents is supported by investigation of individual queries. In Section VII-D, a strategy more appropriate to the environment of this study is proposed. Study of the relative distribution of the vectors describing relevant documents in other collections is recommended.

D. Partial Search and Multiple Query Algorithms

All relevance feedback algorithms evaluated in this study requires a search of the entire document collection for each iteration. In a document collection one hundred times as large as the experimental collection, several full searches per query would be prohibitively expensive and time-consuming on present computers. Since collections of 20,000 documents or more are often encountered in practice, the use of partial search strategies is imperative. No attempt to investigate partial search algorithms is made in this study because the subdivisions of the collection would be far too small to be realistic. However, some of the discussion earlier in Section VII can be extended to partial search algorithm experimentation.

In this section, prior investigations of partial search algorithms in the Cranfield 200 collection are briefly reviewed. Next the evaluation of cluster search techniques is discussed and measures for the evaluation of partial searches and of the general usefulness of a clustering scheme are suggested. Then a new cluster search algorithm is suggested, based on the hypothesis stated in the previous section.

The hypothesis discussed and supported in Section VII-C strongly suggests that an algorithm employing more than one query is needed in the environment of this study. A cluster search algorithm employing relevance feedback and constructing a separate query for each selected cluster is

presented in detail. Then an earlier study of a query splitting algorithm in the Cranfield 200 collection is briefly reviewed. Suggestions for other multiple query algorithms involving relevance feedback are made based on the conclusions of Section VII-C. Finally the clustering of previous requests, suggested by Salton, and the modification of document descriptions based on user requests and relevance judgments are discussed as possible solutions to the problems presented by the hypothesis of that section.

Rocchio [9] proposes an algorithm that assigns every document vector to one or more clusters of similar document vectors, using the distance function that is employed for retrieval in the collection. He suggests that the centroid vectors of the clusters formed by the algorithm be used as a pseudo-collection for a preliminary search, and that only the document vectors in those clusters with centroids nearest the query vector be examined for retrieval. (Hereafter the phrase 'the cluster nearest a query' refers to the cluster with its centroid vector nearer to the query vector than the centroid vector of any other cluster.) Rocchio's clustering algorithm has the following advantage over other methods of partitioning the documents of a collection.

a) Clusters are generated automatically.

b) The cluster size and number of clusters in the collection can be controlled by parameters.

c) A document may be assigned to more than one cluster. This feature allows for documents concerning more than one subject, and may increase the probability that all documents relevant to a query can be found by searching only a few clusters.

Results of two studies of Rocchio's algorithm in the Cranfield 200 collection are here summarized. Salton [22] reports search results after using Rocchio's algorithm to cluster the ADI regular thesaurus vectors and the Cranfield 200 word stem vectors. At all attainable recall levels, precision is lower for the cluster searches than for the full search, except after the 6 clusters nearest the query (30.9% of the document vectors) are searched in the Cranfield 200 collection. Salton concludes that a significant reduction in processing time is achieved with relatively little precision loss (maximum 15%), and recommends cluster search as a money-saving possibility for users not requiring high recall. He also suggests that the queries submitted by previous users be clustered in collections in which either the document space or the subject classifications are subject to rapid change. He proposes a general search algorithm combining cluster search with relevance feedback and other techniques. This algorithm first performs a query cluster search, and then chooses progressively more accurate techniques as needed to retrieve relevant documents. Document vectors for relevance feedback may be selected from the results of a full search or of a partial search.

Leech and Matlack [23] compare the results of clustering the Cranfield 200 regular thesaurus vectors with those of clustering the Cranfield 200 word stem vectors. They conclude that in the regular thesaurus vector collection,

clusters of a size equivalent to five percent of the collection size are optimal, but that larger clusters are needed in the word stem collection. A cluster search of the thesaurus collection gives better recall-precision results than a cluster search of the word stem collection except for large clusters at less than 28% recall. The recall-precision curve generated by searching the two clusters nearest the query using the best set of clusters formed from the thesaurus vectors is slightly higher than the full search recall-precision curve at all recall levels. This result does not indicate that searching two clusters provides better precision than a full search at all recall levels. Because all relevant documents may not be found in the nearest two clusters, some recall levels cannot be achieved for some queries. Extrapolated values for these unattainable recall levels are nevertheless averaged into the recall-precision curve. The average 'recall ceiling', that is the average value of the highest attainable recall level for each query, is 53.4% for the two nearest clusters. On the average, 9.6% of the collection is scanned to obtain this recall ceiling. It appears that performance improvement is achieved for low recall levels and search cost is significantly reduced by a two-level search of Rocchio clusters formed from the Cranfield 200 regular thesaurus vectors.

The evaluation of partial search algorithms presents several problems. In the previous paragraph, difficulty is

encountered in interpreting a comparison of performance measures obtained from a partial search and from a full search. In the SMART system at Cornell [24] the full number of relevant documents is used to calculate all recall and precision measures. Thus the evaluation of partial search results is intended to answer the question 'How well can a partial search retrieve all relevant documents from the total collection?' This question is answered incompletely by partial search recall-precision curves, because these curves give no indication that some recall levels cannot be achieved for some queries, and in fact extrapolated precision values are assigned to unattainable recall levels. The SMART system reports the average recall ceiling for every partial search to give some indication of the recall levels that can be attained. However, because this reported recall ceiling is an average value, some queries may achieve higher recall levels and some may not achieve the ceiling level. Salton [22] and others report partial search results as recall-precision curve segments. For a search of the nearest n clusters, only the curve segment from the recall ceiling of the search of n-1 clusters to the recall ceiling of the search of n clusters is graphed. This type of graph recognizes the recall ceiling problem inappropriately, because some achieved recall levels below and above the bounds of the reported curve segment are ignored. The implied assumption that the performance of the n-cluster search is the same as that of the n-1 cluster search up

to the n-1 cluster recall ceiling is false, because all documents in the n clusters are ranked together by the search, so all documents from the n^{th} nearest cluster are not necessarily retrieved at the bottom of the ranked list. Leech and Matlack [23] report the full recall-precision curve for each partial search and indicate the recall ceiling as a point on the curve. Their solution of the evaluation problem is better than that of reporting curve segments, because no attained recall levels are ignored. However, the problem of distinguishing attainable from unattainable performance remains.

By extension of the discussion of recall-precision interpolation in Section VII-B, the SMART rightward extrapolation method for partial search recall-precision curves defines an equivalence relation between partial search performance and full search performance at all recall levels not attained by the partial search. The results of performance comparisons between partial and full searches are largely dependent on the equivalence relation defined by the choice of a rightward extrapolation method. The definition of an equivalence relation between queries with different numbers of relevant documents by precision interpolation at unattained recall levels seems reasonable. The definition of an equivalence relation between a partial search and a full search of the same query by precision extrapolation at unattainable recall levels is less easily justified. A possible alternative is to refuse to extrapolate to the right, but instead to average at each recall

level only queries that attain equivalent or higher recall. For each point on the recall-precision graph of a partial search, the number of queries attaining that recall level would be reported.* This alternative as proposed above eliminates doubt of the validity of partial search recall-precision curves at high recall levels, but still does not provide direct performance comparison to a full search curve because different queries would be used for averaging the high recall points. It would be possible to construct for each partial search curve a matched full search curve that averages at each recall level the full search precision of the queries attaining equal or higher recall on the partial search. This second alternative gives a directly interpretable comparison between full and partial search recall-precision curves by failing to report all full search results. Each of the proposed partial search recall-precision curves illuminates the experimental situation from a different angle; all three curves may be needed in some cases to provide even and unshadowed lighting.

Though partial search and full search recall-precision performance is difficult to compare, the document curves provide a direct answer to another question relevant to partial search strategies: 'What performance has been

*The SMART system now reports the number of queries achieving a given recall or less without extrapolation so that the extent of leftward extrapolation at low recall levels can be estimated.

achieved by each search after the same percentage of the total collection has been scanned?' The document curves report recall and precision at several possible cut-off ranks, so they can be used to answer questions of the form 'Is it better to give the user all n documents in the nearest cluster or the top n documents of the full search?' These curves provide direct and meaningful comparability between partial and full search strategies and between alternative partitions* of the same collection.

In the preceding discussion the distinction between attained performance and attainable performance arises. Recall ceiling is a measure of the highest recall attainable in a cluster, though that recall may be attained after only part of the cluster has been searched. Since different multi-level search strategies might use the same set of document clusters, attainable performance may provide a better indication of the general usefulness of a given partition of the document collection than the performance attained by one particular search strategy. In a study of clustering in the ADI collection, Grauer and Messier [25] use three measures that are not related to the search strategy employed, but that may be used jointly to indicate the utility of a given partition of a document collection. One of these measures is recall ceiling, an indicator of attainable performance. The other two measures are called 'user percentage scanned' and 'machine

*Hereafter, a set of clusters such that their union includes all document vectors in a collection is called a partition of the collection.

percentage scanned'. These three measures are defined below in terms unrelated to any specific search strategy:

Let N = number of documents in the collection

C = number of clusters in the partition being evaluated

Q = number of queries in a representative query sample used for evaluation

Then given a number of clusters n and a query i, let

R_i^n = the number of documents relevant to query i in the n clusters closest to query i.

D_i^n = the number of documents in the n clusters closest to query i.

R_i = the total number of documents relevant to query i in the collection.

Then

recall ceiling (n) = $\frac{1}{Q} \sum_{i=1}^{Q} \frac{R_i^n}{R_i}$

the average ratio of the number of documents in the nearest n clusters to the total number of relevant documents.

user percentage scanned (n) = $\frac{1}{Q} \sum_{i=1}^{Q} \frac{D_i^n}{N}$

the average ratio of the number of documents in the nearest n clusters to the collection size.

machine percentage scanned (n) = $\frac{1}{Q} \sum_{i=1}^{Q} \frac{D_i^n + C}{N}$

the average ratio of the number of vectors searched by a two level partial search of the nearest n clusters

to the number of vectors searched by a full search of the document collection. Machine percentage scanned is a system-independent indicator of the search time or search cost of a partial search relative to a full search. (For a partial search strategy involving a different number of vectors, the machine percentage scanned strategy could be changed to indicate the changed search cost.)

As Grauer and Messier [25] point out, these three measures do not provide direct comparability between alternative partitions of the collection. The type of question asked of these measures is 'If partition A yields an average recall ceiling of 25% for the nearest two clusters, and these two clusters include 30% of the collection, while partition B yields an average recall ceiling of 35% and the nearest two clusters include 40% of the collection, which partition is better?' An answer to this type of question is here proposed that leads to two directly comparable and meaningful measures of the utility of alternative partitions of a document collection. The first measure is based on the notion of generality number used by Cleverdon and Keen [18]. The generality number of a collection is the ratio of the average number of documents relevant to a query (calculated from a representative query sample) to the number of documents in the collection. In a collection with a higher generality number, precision is generally higher [18]. The goal of a two level search using

a partition of the documents collection is to find the same relevant documents by searching fewer document vectors. Therefore, the partition used should effectively increase the generality number of the searched collection for each query, that is, it should select for each query a subset of documents containing more relevant documents in proportion to the subset size than the entire collection contains in proportion to its size. The 'generality factor' defined below is a strategy-independent measure of the extent of which a given partition of the document collection accomplishes this aim:

$$GF(n) = \frac{1}{Q} \sum_{i=1}^{Q} \frac{R_i^n}{D_i^n} \div \frac{R_i}{N}$$

The average factor by which the proportion of relevant documents to searched documents is multiplied by clustering the document vectors and selecting the n clusters closest to each query.

A second measure, called the cost factor, is based on the comparative cost of a partial search to a full search, as is the machine percentage scanned. The cost factor is defined with the same structure as the generality factor:

$$CF(n) = \frac{1}{Q} \sum_{i=1}^{Q} \frac{R_i^n}{D_i^n + C} \div \frac{R_i}{N}$$

the average factor by which the proportion of relevant documents to searched vectors is multiplied by clustering

the document collection and selecting the n clusters closest to each query. Note that a cost factor greater than 1 indicates that the cost of a partial search is lower than that of a full search.

The generality factor and cost factor each define an equivalence relation between two partitions that may achieve different recall ceilings with document subsets of different sizes.

It is interesting to note that a re-evaluation of the Grauer and Messier [25] results using estimates of the generality factor and cost factor measures clearly shows that clustering the 82 document ADI collection isn't worth the trouble. Only a few runs have generality factors as high as 2.0 and for these runs the cost factor is less than one, indicating a search cost greater than that of a full search.

By contrast, the Leech and Matlack [23] clusters in the Cranfield 200 collection yield estimated generality factors from 3 to 9 and estimated cost factors from 1.5 to 2.6. In larger collections, the difference between the generality factor and the cost factor of a run would probably be smaller. For comparison of different partitions of a document collection, it is suggested that for each partition n (number of clusters searched) be increased until a recall ceiling of 100 is reached, and that the generality factor and/or the cost factor be plotted against the recall ceiling for each possible n.

One further suggestion for cluster search algorithms can be made on the basis of the hypothesis stated in the previous section. The Rocchio clustering algorithm has been used with only one two level search strategy, that of choosing the nearest n clusters and ranking in one search operation all documents in these n clusters. This procedure may not be ideal for most queries. If n equals 2, for example, the centroid of the second cluster may be farther from the original query than that of the first cluster, indicating that in general the documents in the second cluster are farther from the original query than those in the first cluster. It is possible, therefore, that some if not all relevant documents in the second cluster are retrieved later in a joint search of both clusters than are some non-relevant documents in the first cluster. If all relevant documents form a single cluster in

the unpartitioned document space, this problem does not occur. However, according to evidence in Section VII-C the relevant documents are usually separated from each other in the document space.

If each Rocchio cluster is searched separately, however, the user's query is only required to separate the relevant documents in each cluster from the non-relevant documents in the same cluster, rather than to separate all relevant documents from all non-relevant documents in the clusters searched. Within a single Rocchio cluster, the occurence of separated clusters of relevant documents might be less evident than in the full collection. In fact, for some queries each separated cluster of relevant documents might be found in a different Rocchio cluster, thus providing within each cluster a retrieval situation that a single query can resolve.

The foregoing argument suggests a cluster search algorithm that ranks each document relative to other documents in the same cluster, and retrieves the highest ranking documents from each cluster searched. Construction of such an algorithm presents a strategic problem - in what order are the documents to be presented to the user? This problem can be rephrased in terms of performance evaluation - given the ranks of all documents relative to other documents in the same cluster, how are ranks to be assigned to all documents in the collection for comparison with other strategies not using the same partition

of the document space? The simplest method is to assign the first n ranks in rotation to the first document of each cluster searched, and so on. This 'rotation' method of ranking all documents makes no special provision for clusters of different sizes or for clusters that might be expected to contain more relevant documents. Modified rotation methods might be constructed that automatically assign more high ranks to documents in the larger clusters, or to documents in the clusters nearer to the original query. Another alternative worth testing is to rank all documents according to the distance of each document from the original query relative to the distance of the cluster containing that document from the query. Coefficients providing this ranking could be obtained by subtracting from the correlation coefficient of each document the coefficient of the centroid of the cluster containing that document. Because the Rocchio clustering algorithm allows cluster overlap, an overall ranking method must define the rank of a document appearing in more than one cluster. Such a document might be assigned the highest of the possible ranks, or perhaps the rank assigned by its position in the cluster nearer the original query.

Investigation to determine the most appropriate ranking method for combining separate cluster searches should be conducted. Residual collection evaluation, defined in Section VII-B, is a valuable tool for such a study. If each cluster is evaluated separately, the

efficiency of the query in separating the relevant documents from the non-relevant documents within each cluster can be determined, and can be compared to the ability of the same query to separate all relevant from all non-relevant documents in the searched clusters. With this information the feasibility of separate cluster searches, and of some of the possible ways of combining them, can be estimated.

It is evident from Section VII-C that a multiple query algorithm is usually needed to separate all relevant documents from all non-relevant documents in the full collection. The preceding discussion indicates that a partial search algorithm might take advantage of the possibly simplified retrieval task within each selected cluster of documents by searching each cluster separately. However, even if only one query is required for ideal retrieval within each cluster, it is very unlikely that the same query can accomplish this task for every selected cluster. A combination of relevance feedback and cluster search techniques is indicated, to tailor a specific query for each retrieval situation encountered in processing a user's request.

The partial search relevance feedback technique proposed here treats each cluster as a separate document space, and could use any relevance feedback algorithm to construct a query intended to separate relevant from non-relevant documents within that cluster. Any technique

using relevance feedback to construct a single query for each document cluster on the lowest search level of a partial search algorithm is herein called 'cluster feedback'. A detailed description of a general two-level cluster feedback algorithm is presented below. Two considerations in defining this combined algorithm have not been encountered in the cluster search or relevance feedback strategies discussed in this report. The first is the possibility of using relevance feedback to select additional clusters to be searched, seen in steps 6-8 below. The second is the economic need to abandon the search of unproductive clusters as soon as possible. The methods of discarding queries that are incorporated into the suggested cluster feedback algorithm could also be used for full search relevance feedback and for the multiple query feedback algorithms discussed later in this section.

The detailed algorithm description below includes some explanation and lists alternative strategies for critical steps. Figure 39 displays an abbreviated algorithm description in flowchart notation.

A Two-Level Cluster Feedback Algorithm:

Step 1. Search all cluster centroid vectors and select the clusters closest to the original query q_0. The number of clusters selected might be the same for each information request. However, other possibilities should be investigated, such as selecting all clusters with centroid correlations to q_0 greater than some π, or choosing

Step 5. Discard any query constructed in Step 4 that contains fewer than k concepts. Also discard the associated cluster.

This step is optional, and is needed only when negative feedback is used in Step 4. See Step 10b for a related method of discarding unproductive queries.

Step 6. Construct a new centroid search query using the original query, any previous centroid search query, and the documents retrieved from all clusters.

Steps 6-8 optional. The utility of this process in retrieving additional clusters containing relevant documents should be investigated. An experimental system should include the possibility of omitting Steps 6-8 after j iterations.

Step 7. Select the clusters with centroids closest to the centroid search query of Step 6.

The numbers of clusters to be selected in this step may be determined in the same manner or in a different manner than in Step 1. The number of additional clusters selected might be allowed to influence the number of documents to be selected from each cluster in Steps 8 and 9.

Step 8. Search the cluster just selected by Step 7 using the centroid search query constructed in Step 6.

Step 9. Search all other clusters that have not been

The number of documents retrieved from each cluster might be determined in the same manner or in a different manner than in Step 2.

Step 10. Discard any query and associated cluster that does not meet the following criteria as a result of Step 9.

a) All documents in the cluster have been retrieved.

Present the documents last retrieved in Step 9 to the user but do not ask for relevance judgments.

b) Of all unretrieved documents in the cluster, the span between the highest and lowest correlation is less than some d.

This condition indicates that the query is too general to select more documents from the cluster, since all remaining documents are about the same distance from the query. Checking for this condition may make Step 5 unnecessary.

c) The highest correlation of any unretrieved document in the cluster with the original query is less than c.

This condition indicates that the query is too specific, because the cluster contains no more documents similar to it. The later discussion of multiple query algorithms suggests alternate queries for this condition.

Step 11. Obtain relevance judgments on all documents selected in Steps 8 and 9 except those documents selected from clusters discarded in Step 10.

Step 12. Discard any query and associated cluster that has retrieved no new relevant documents in M iterations.

Step 12 may not be needed if all conditions suggested in Step 10 are checked.

Return to Step 4 to search all clusters that have not been discarded, including those new clusters last selected by Step 8, if any.

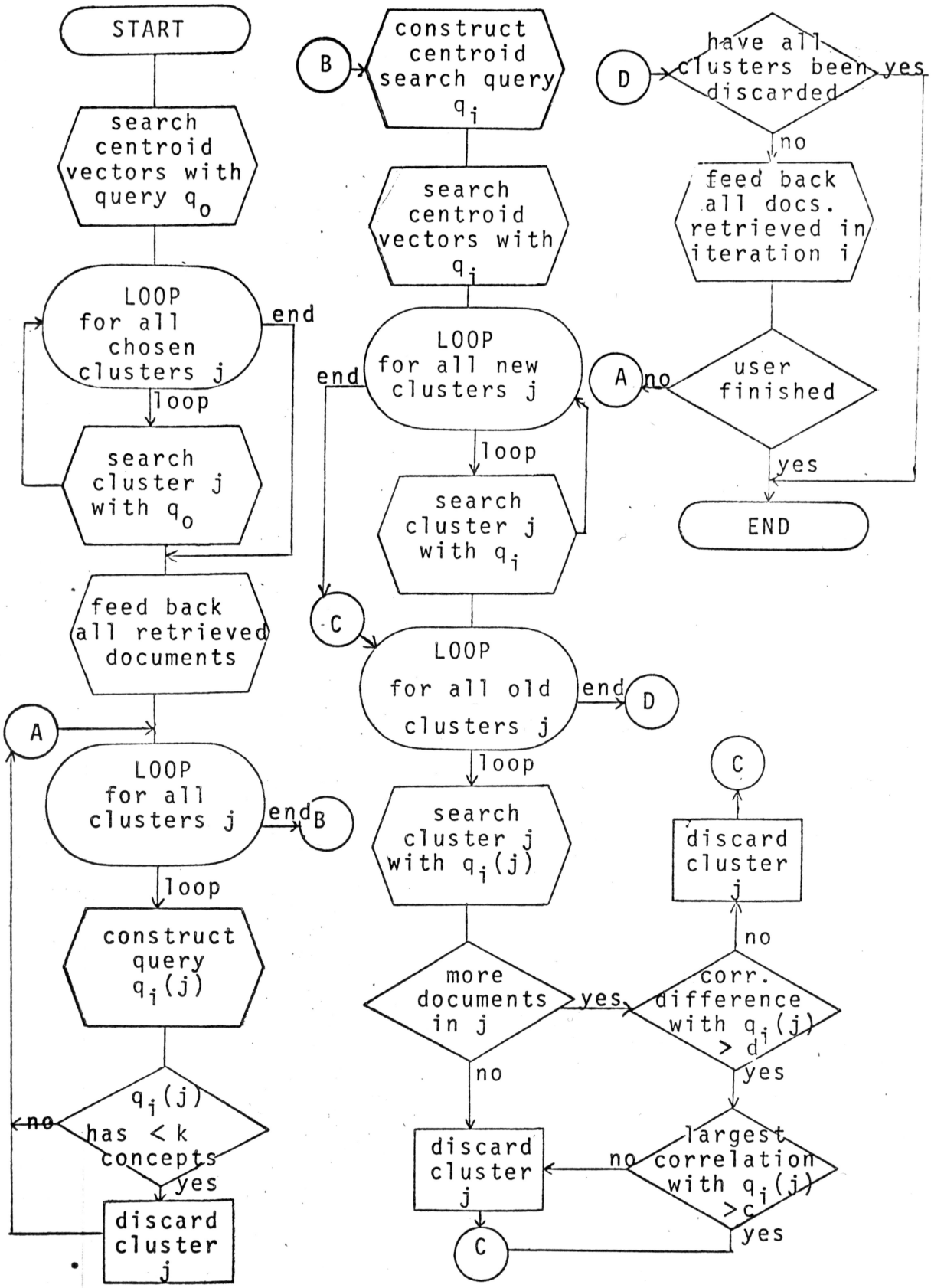
Compared to full search relevance feedback, the above algorithm will provide improved retrieval at decreased search cost if the following conditions are true:

1. The partition of the document space must not overlap so much that more documents are processed by searching the selected clusters separately than by searching the full document collection.

2. The retrieval problem within each cluster must be simpler than the retrieval problem in the full collection. In the ideal case each cluster would require only one query for ideal retrieval.

3. The cluster selection in Steps 1 and 6 must select those clusters containing relevant documents, and must select few unproductive clusters. If unproductive clusters are selected, they must be discarded early in the iterative process.

Condition 1 can be controlled by the Rocchio clustering process. Condition 2 is likely to be true in environments similar to that of the present experiments. Investigation to determine the document vector collections, document space partitions, query types, algorithm variations, and algorithm parameters (k,c,d, etc. in algorithm description) resulting in improved performance at lower search cost should be conducted. Large document and query collections (at least 1000 documents and 500 queries) should



A General Two-level Cluster Feedback Algorithm

Figure 39

be used for all experiments with this algorithm.

Although the retrieval situation within each cluster is probably simplified by cluster feedback, separated clusters of relevant documents might still be encountered, particularly in large document collections divided into relatively large clusters. Therefore, it may still be necessary to investigate the possibility of constructing a query for each separated relevant cluster in a set of documents. In this report, a 'multiple query' relevance feedback algorithm is defined as a strategy that constructs more than one query to search the same set of documents on the same feedback iteration, whether that set of documents is a standard cluster or the full document collection. This definition is used to stress an important distinction between multiple query algorithms and simple cluster feedback, which constructs only one query per iteration to search each selected document cluster. Although cluster feedback constructs more than one query, it may still use the feedback algorithms based on Rocchio's assumption that all relevant documents are grouped together in the document set being searched. Multiple query algorithms are constructed to provide improved retrieval in cases when this assumption is not valid, so such algorithms require the development of relevance feedback strategies radically different from those studied previously.

The only previous investigation of a multiple query algorithm in the SMART system uses the Cranfield 200 col-

lection. Borodin, Kerr, and Lewis [26] study a straightforward technique for constructing multiple queries, called 'query splitting'. Whenever the relevant documents retrieved by some query q form two or more clusters that are relatively far from each other in the document space, each such cluster of retrieved relevant documents is used separately to form a new query. The two highest non-relevant documents retrieved by q are used for negative feedback in forming each new query. If all retrieved relevant documents are near each other in the document space or if no relevant documents are retrieved, only one new query is formed using the Dec 2 Hi strategy. A retrieved relevant document is considered 'far' from another if the correlation between them is less than some constant times the average correlation of q with all documents retrieved by q on that iteration.

The algorithm described is tested on the 24 Cranfield queries that retrieve more than one relevant document on some iteration with N equal to 5. A 'user measure' table details the relative performance of each query for which the retrieval of the first 25 documents is changed by query splitting. Borodin, Kerr, and Lewis conclude that the result in this table 'favor query splitting', and add that the relative performance of their query splitting algorithm would be better in larger collections. They suggest that an additional query formed by negative feedback alone should be constructed for each iteration, and that methods of discarding unproductive queries be included in the algorithms.

The following facts can be ascertained from the data available from the experiments presented herein and the user measure table presented by Borodin, Kerr, and Lewis.

1. The early retrieval of 11 of the 24 queries is changed by query splitting (when 12.5% of the collection has been retrieved).

2. Only 4 of these 11 queries are improved by query splitting. Performance of the other 7 changed queries is degraded.

3. None of the 12 queries for which the Rocchio strategy performs more poorly than positive feedback (the Q_0^+ group) are improved by query splitting. One of them is degraded.

4. Only 2 of the 18 queries seriously affected by the presence of separated clusters of relevant documents are assisted by query splitting. Four of these queries are degraded.

The above findings contradict the conclusion of Borodin, Kerr, and Lewis, and indicate that query splitting does not solve the problem for which it was constructed. The contention of the three authors that query splitting would be more effective in a larger collection is probably true, but it is evident from the previous section of this report that there is considerable room for improvement in the Cranfield 200 collection. The failure of query splitting in the collection studied indicates that the algorithm

tested is inadequate as a solution to the retrieval problems caused by separated clusters of relevant documents.

The query splitting strategy tested constructs a specific query for each relevant cluster represented by retrieved documents. However, it is probable that the queries displaying the poorest performance do not retrieve relevant documents from each separated relevant cluster. Query splitting is still based on the Rocchio assumption found invalid in this collection that the retrieved relevant documents are representative of all relevant documents. Cluster feedback as suggested earlier in this section assumes that separated relevant clusters will not seriously affect retrieval within the standard document clusters used to partition the document space. Unless this assumption can be verified in typical document collections by outstanding cluster feedback results, less optimistic strategies should also be investigated.

Two considerations uniquely characteristic of multiple query algorithms are the basis of the following discussion. The first is the possibility of constfucting more than one query from relevance feedback on retrieved documents. The second is the need to construct useful queries under the assumption that the documents used for feedback are not necessarily representative of the documents remaining in the collection being searched.

Borodin, Kerr, and Lewis [26] compare the correlation between retrieved relevant documents to the average query-document correlation for a given iteration, in order to define clusters of retrieved relevant documents. One query is then constructed using each retrieved cluster. However, the discussion in section VII-C of this report indicates that if negative feedback is used, the distance between two relevant document vectors may be less important than the presence of a non-relevant document vector between them. Therefore, it is suggested that separated rather than separate relevant clusters be sought. Two relevant document vectors r and v would be assigned to different clusters if there exists a non-relevant document n retrieved previously or concurrently such that cos (n,v) is greater than cos (r,v) and cos (n,r) is greater than cos (r,v). Any retrieved relevant document vector that is in this way assigned to more than one cluster could be assigned only to the cluster closest to it, or if the distqnces between alternative clusters

are near equal, could form a separate cluster. It is clear from Figure 40 that the suggested clustering criterion is quite strong. Even though an ideal single query could retrieve all three relevant documents in the situation symbolized, each of them is assigned to a different cluster because the one non-relevant document is closer to each than are the other two relevant documents.

If the clustering criterion suggested above is used, there is a possibility that a defined cluster of retrieved relevant documents could be broken up by a non-relevant document retrieved on a subsequent iteration. The defined relevant cluster could be broken into smaller clusters and a new query formed from each. This re-clustering would require that the algorithm have access to all vectors of relevant documents retrieved on previous iterations and that it determine the relationship of each non-relevant document used for feedback to each document of the relevant cluster defining the query being altered. The clustering criterion defined by Borodin, Kerr, and Lewis [26] does not raise this problem. However, the suggested choice of separated clusters of relevant documents guarantees no feedback conflicts between relevant and non-relevant documents used to alter the same query, and also minimizes the number of queries formed by avoiding the formation of different clusters of relevant documents until such a feedback conflict is likely to occur.

It has been established that the information obtained from retrieved relevant documents may not be sufficient to

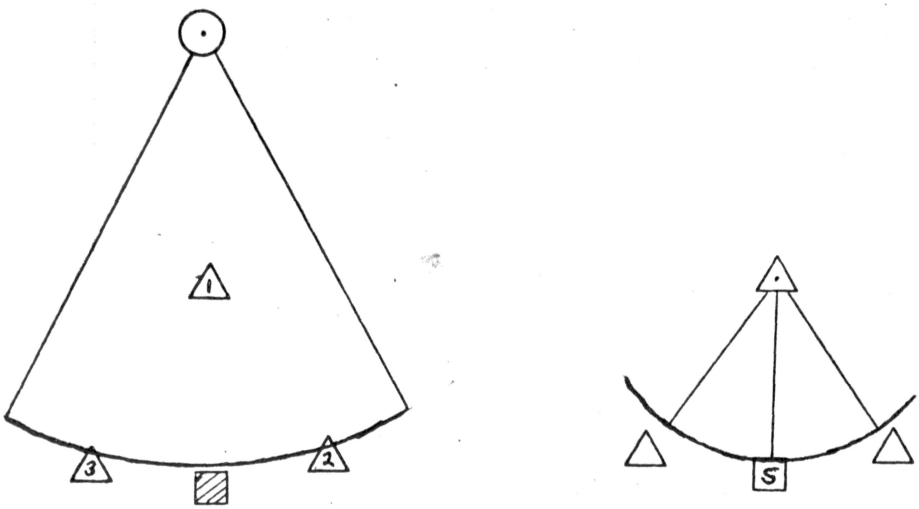
retrieve all relevant documents. In the present Smart system, the only available source of information about relevant documents not represented by retrieved documents is the user's original query. A multiple query strategy should make specific use of the concepts chosen by the user to express his needs, and should ensure that none of these concepts are ignored.

There are three ways in which a concept from the original query can be effectively cancelled from the search by a strategy using positive and negative feedback. First, other concepts found in the retrieved relevant documents may have much larger weights and thus greater effect on subsequent iterations than a user-selected concept not found in the first relevant

○ ideal query for the documents symbolized

△ relevant documents

□ non-relevant documents



All relevant documents are retrieved by one query,

BUT $\cos(r_i,s)$ is less than $\cos(r_i,r_j)$ for all i, j.

Three Relevant Documents Forming Three Separated Clusters

Figure 40

documents retrieved. Strategies using relevant documents only and giving extra weight to the original query do not solve this problem (see Section VI-B) because a concept appearing in the original query and in relevant documents will still outweigh a concept found only in the query. To give an extreme example, a query on 'the aerodynamics of birds' addressed to the Cranfield collection would quickly become in effect a question on 'aerodynamics'. By contrast, a human librarian confronted with this situation might make a special effort to find any document concerning 'birds'.

There are several ways in which similar stress on concepts not immediately found in retrieved documents can be incorporated into an automatic multiple query search. The construction of one query using only negative feedback (also suggested by Steinbuhler and Aleta [13]) would eliminate concepts found in non-relevant documents without disproportionately increasing the weight of any concept. In some cases however, it might be more effective to pinpoint precisely the concepts that are being ignored. In the example given, any query that still contained the word 'aerodynamics' would probably not retrieve a document containing only the rare concept 'birds'. Therefore, a query constructed by subtracting the retrieved relevant documents from the original query might be useful, on the theory that if any user-chosen concepts remain after such

drastic negative feedback, something should be done about it.

The second and third ways in which a user-selected concept can be ignored are caused by negative feedback. The user may employ a concept occurring in the given collection only in documents not relevant to his needs. In this case, of course, negative feedback appropriately eliminates the misleading concept. The third possibility is that an initial query concept is used in the collection with more than one meaning, and thus is relevant in one context and irrelevant in another. If non-relevant documents containing this concept are retrieved first, negative feedback may erase meaningful information. This third case might explain the type of behavior displayed in Figure 37, and the complete query erasure occurring with all negative feedback to query 34. There is no way in the present SMART system to distinguish the third possibility from the second, to judge whether a user concept erased by negative feedback should be re-inserted or forgotten.

Adding information about concept-concept relationships to the system might enable a negative feedback algorithm to distinguish a completely irrelevant initial concept from a concept relevant in the appropriate context. For each concept in the thesaurus, a weighted list of other concepts often occurring in the same document as the given concept should be stored. For a concept used in two ways in the collection, this list would include related concepts from both possible contexts.

The suggested lists could be constructed automatically from the document vectors, perhaps by using the asymmetric coefficient of concept similarity suggested by Salton for automatic hierarchy construction [3]. If a concept from the original query is eliminated by negative feedback, the query could be immediately reformulated by adding some number of related concepts to the previous query and then repeating the same negative feedback. Added concepts appropriate to irrelevant contexts might be eliminated by negative feedback while added concepts from the relevant context might be retained, preserving by context the intended meaning of the eliminated concept in the user's query. This suggestion is a variation of the strategy suggested by Kelly[14] (see Section III), but differs in that concepts closely related to eliminated initial query concepts rather than concepts occurring frequently in the collection are added to the query. The idea detailed above is here called the 'related concept Kelly strategy' and is appropriate to both single and multiple query feedback algorithms.

Another way of supplying context information to the SMART system without requiring permanent storage involves the use of negative query weights. If all document weights are positive, a concept weight below zero in a query tends to indicate that documents containing that concept are not relevant. (It may be important to realize that with negative weights, the cosine correlation coefficient as calculated by the SMART system has a range from -1 to +1 and no longer corresponds to the cosine

of the angle between the vectors.) Negative query weights could be used to supply context information to differentiate possible meanings of original query concepts as follows: A 'non-relevant context' vector is constructed by adding together the vectors of the non-relevant documents retrieved, and then setting every concept occurring in the original query to zero. In other words, concepts contained in the original query are ignored when they appear in non-relevant documents. Then a new search query is formed by multiplying the non-relevant context vector by some constant less than one and then subtracting the resulting vector from the original query vector. The suggested procedure preserves all original query concepts with their original weights. However, any other concepts that occur in non-relevant documents are given a negative weight. Thus if two unretrieved documents contain the same original query concepts with the same weights, the document having the fewest other concepts in common with retrieved non-relevant documents is retrieved first. The use suggested above for negative query weights is here called 'selective negative weighting' and is appropriate to single query or multiple query strategies. Selective negative weighting avoid the negative weight problem encountered by Kelly [14], who did not preserve the original query and found that all cosine correlations with the new query were often negative. If all cosine correlations are negative after selective negative weighting, either the suggested multiplier constant is too large or else no documents

in the collection contain the original query concepts in a context that has not been declared non-relevant by negative feedback. Selective negative weighting may also be used when relevant documents are retrieved, in which case only the concepts from the original query are set to zero in the non-relevant context vector. In this way the deletion of superfluous or misleading concepts from relevant document feedback by negative feedback may still occur. If negative weighting is used with the related concept Kelly strategy, preservation of the original query concepts may not be necessary.

The discussion thus far has described two distinct types of queries that could be constructed by a multiple query algorithm, each type of query serving a distinct purpose. The 'specific' query is a type of query constructed from a cluster of retrieved relevant documents to retrieve similar documents. The 'general' query is constructed to retrieve documents not represented by the retrieved relevant documents. These two types of query contrast in structure as well as in purpose. The specific query is largely constructed from document abstracts and contains many concepts of high weight, at least at first. Because the specific query vector is long and contains many concepts, few document vectors will have high correlations with it. The general query is constructed from the original query and possibly from related concepts, and has fewer concepts with lower weights. Thus in the specific query discarding superfluous and misleading concepts is a prime consideration, while in the general query preserving and clarifying the meaning of the original query is the chief aim.

A multiple query algorithm should therefore use different relevance feedback strategies for general than for specific queries. Some of the consideration important in altering each type of query are listed below:

1. The specific query is intended to select only relevant documents similar to a retrieved cluster of relevant documents. Therefore by Rocchio's theory, the optimum query to differentiate the retrieved cluster from all other documents, including other relevant clusters, should be approached by iteration. That is, the retrieved relevant cluster should provide positive feedback and all other documents retrieved by any query, relevant or non-relevant, should be used for negative feedback. Since general queries have fewer positively weighted concepts, each general query should perhaps be altered only by the non-relevant documents it retrieves.

2. The Crawford and Melzer study may indicate that the original query need not be used in constructing specific queries. By contrast, only the original query is used for positive feedback to general queries.

3. Since a specific query is intended to select documents similar to retrieved relevant documents, it should be discarded quickly if no similar relevant documents are found, or if no spread in query-document correlation is produced (if all remaining documents are roughly the same distance from the query). However, the highest query-document correlation can can be fairly low without indicating that a specific query is useless as a selector. The Rocchio strategy does not necessarily construct a query that is close to relevant

documents, but rather one closer to relevant than to non-relevant documents. In the situation symbolized in Figure 40, the single query that best separates the relevant documents from the non-relevant documents is some distance from each depicted document. Therefore specific queries should be discarded quickly on the criteria described in steps 10b and 12 of the cluster feedback algorithm presented earlier but should not be evaluated by the criterion described in step 10c. On the other hand, a general query is intended to retrieve relevant documents of types not previously encountered. The shorter and less detailed general query typically correlates more strongly with more documents than the specific query, and has a smaller spread in query-document correlations. Therefore, the criteria of steps 5 and 10c are of greater importance in judging the worth of a general query than the criteria of steps 10b and 12.

4. Since each specific query searches a relatively small area in the document space, the immediate construction of a new specific query for any retrieved relevant documents that cannot be added to the cluster defining the retrieving query may not be redundant. However, since a general query might retrieve a wide variety of relevant documents, any relevant document retrieved by a general query that has been previously or concurrently retrieved by any specific query should be ignored in processing the general query. Further, any relevant document retrieved by a general query that can without conflict be added to the cluster defining one and only one specific query should

be so treated rather than being used to form a new specific query.

5. If the retrieved relevant cluster defining a specific query is subsequently separated by a retrieved non-relevant document to be used for feedback to that query, a new query should be formed for each subcluster without using the previous query defined by the original cluster. In this way each specific query is defined by a single cluster of relevant documents and no relevant documents separated from that cluster are included in the positive weight of the defined query.

Although single query algorithms are known to be inadequate for retrieval, it is not clear whether all the complexities suggested in the foregoing discussion are necessary. Before further experimental effort is invested in multiple query algorithms of this type, the related concept Kelly strategy and selective negative weighting could be tested in the Cranfield 200 collection by ignoring retrieved relevant documents as Steinbuhler and Aleta do [13]. If both of these strategies prove ineffective, the usefulness of the general query in a multiple query algorithm is doubtful, unless some other means of clarifying the meaning of the original query is found.

Figure 41 details a multiple query algorithm incorporating separated clusters of retrieved relevant documents, the related concept Kelly strategy, all suggested query deletion procedures, two general queries, and distinct feedback algorithms for specific and general queries. Selective negative weighting

n_r = number of relevant documents retrieved this iteration

n_s = number of non-relevant documents retrieved this iteration

$r_i(s_i)$ = a relevant (non-relevant documents retrieved this iteration

v_i^j = a relevant document in cluster j

cluster j = a cluster of retrieved relevant documents such that there do not exist two documents v_i^j and v_k^j and a non-relevant document s retrieved on any previous iteration by any query such that $\cos(v_i^j,s)$ is greater than $\cos(v_i^j,v_k^j)$ and $\cos(v_k^j,s)$ is greater than $\cos(v_k^j,v_i^j)$.

Q_o = user's original query

G_k = a general query constructed by the algorithm

S_j = a specific query constructed from the documents in cluster j

$n_r^k(n_s^k)$ = number of relevant (non-relevant) documents retrieved by query S_k or G_k on the present iteration.

$r_i^k(s_i^k)$ = a relevant (non-relevant) document retrieved by query S_k or G_k this iteration

d_i^k = any document retrieved by query S_k or G_k this iteration

ng(k) (ng(j)) = an indicator set to 1 the first time query G_k (S_j) retrieves no relevant documents

a,b,c = control parameters for tests of query usefulness

Figure 41 - page 1 of 5

con (v) = number of concepts in vector V

ADCON (G_k, G_l) is a subroutine that constructs a new query by adding related concepts to query G_k for all concepts l on list L and not in query

$$(n_s^k G_k - \sum_{i=1}^{n_s^k} s_i^k).$$

After concepts related to l has been added to one query, concept l is removed from list L. The constructed query is stored in G_l.

CLUSTER clusters all documents on list LR by the criterion described in the definition of 'cluster j', adds the new clusters to the set J of clusters j, and clears list LR.

QUERY forms a specific query S_j for each cluster j on iteration I as follows:

$$S_j = K_I \sum_{i=1}^{n_r^j} r_i^j - n_r^j N_I$$

Figure 41 - page 2 of 5

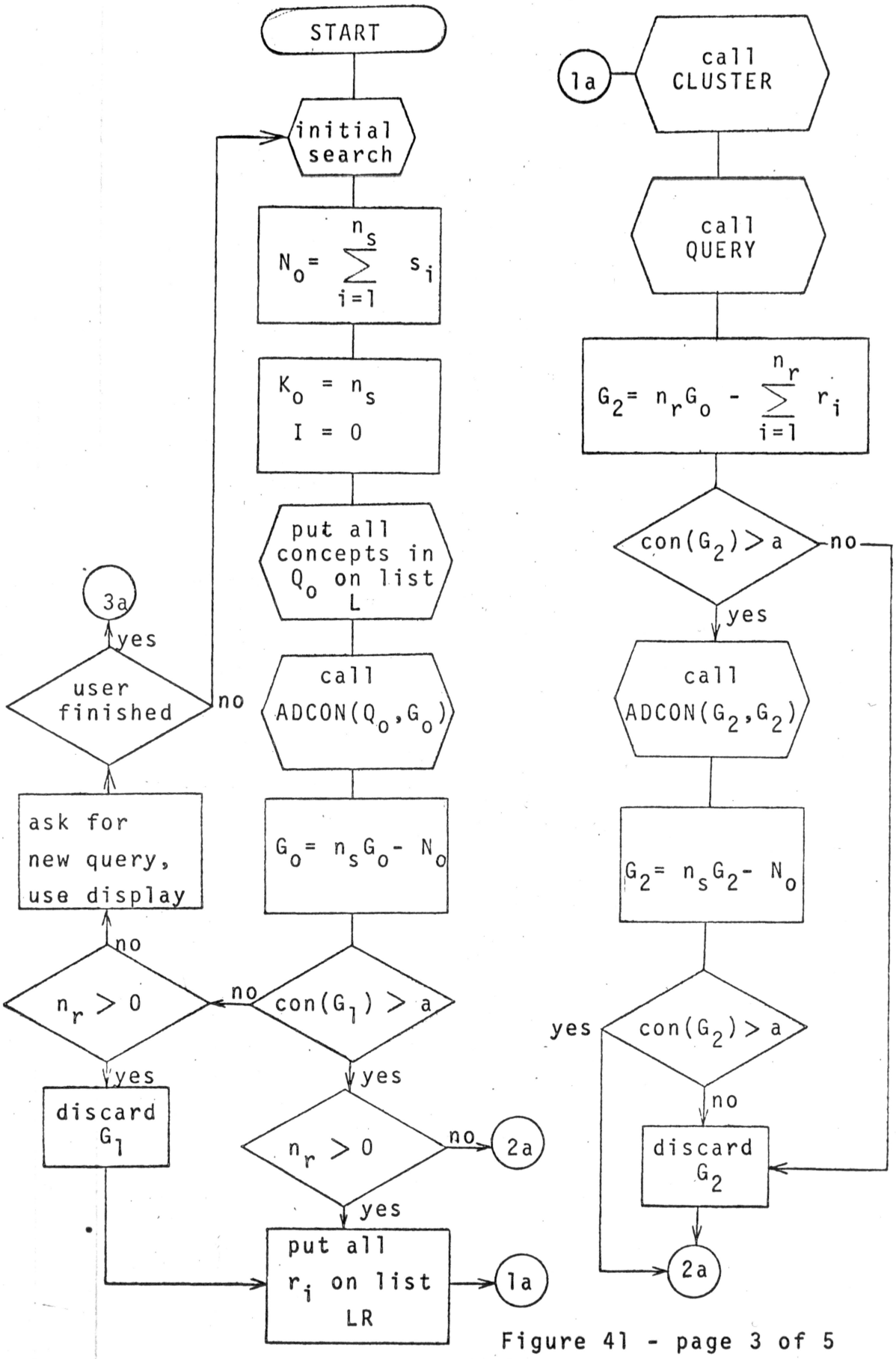


Figure 41 - page 3 of 5

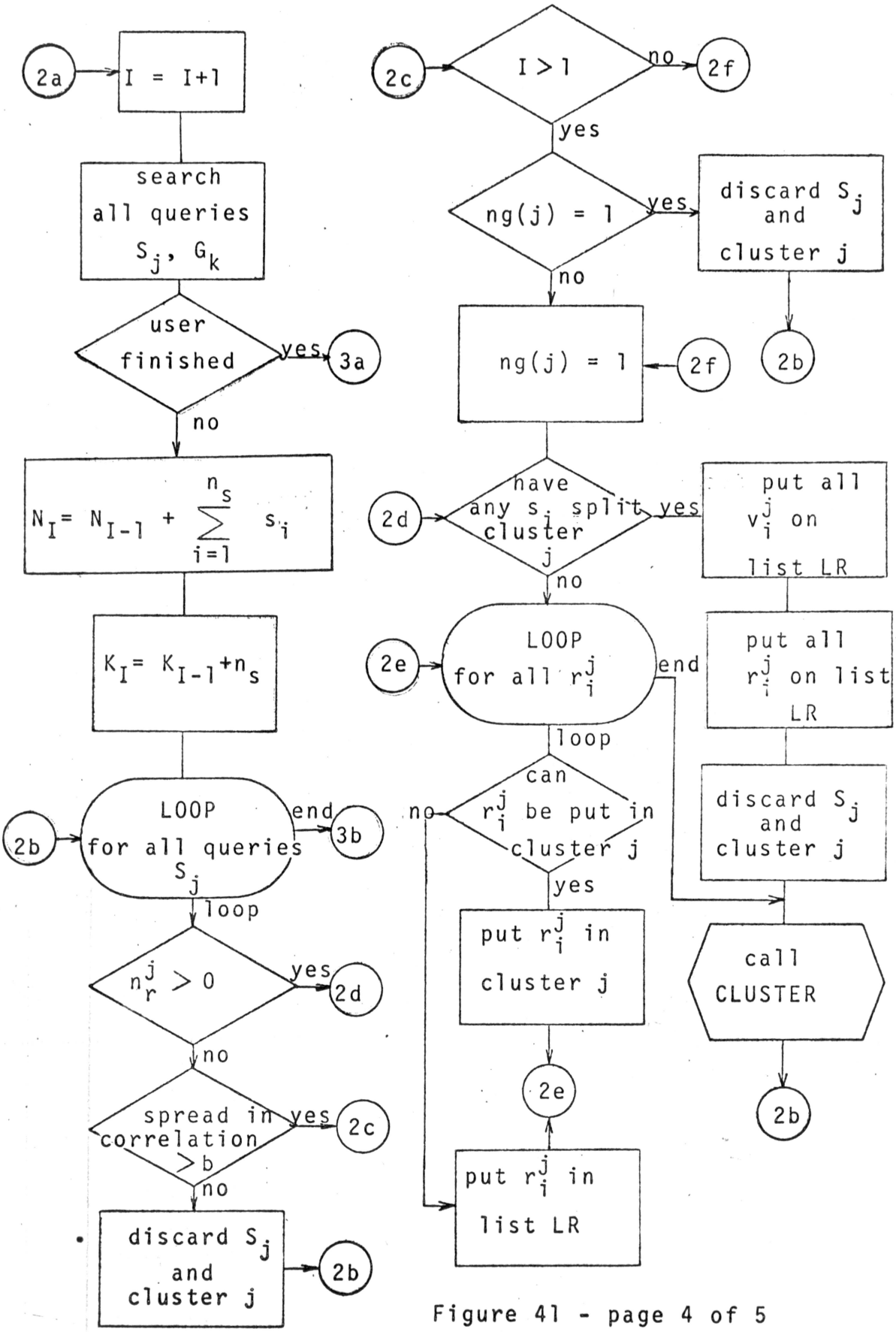
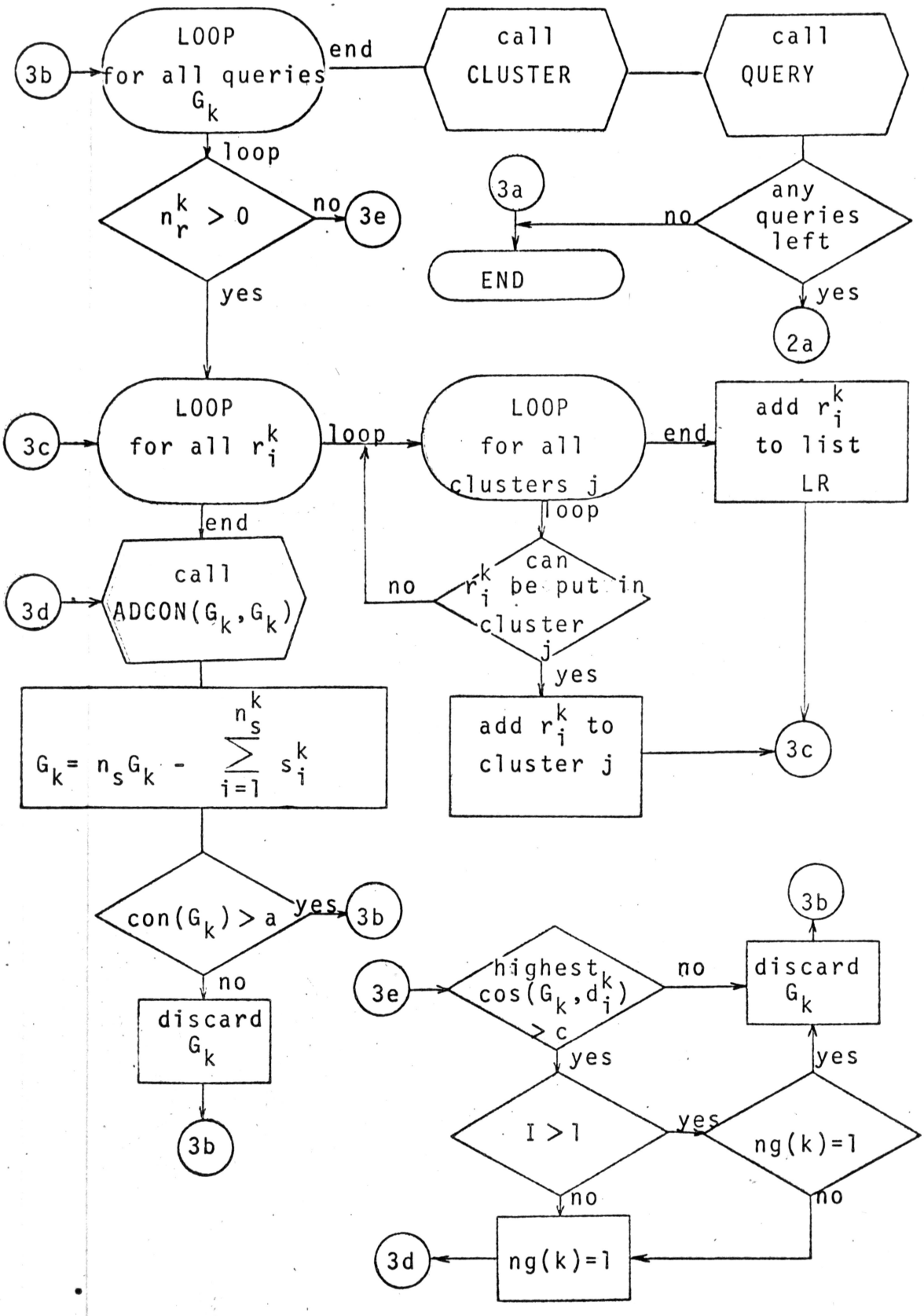


Figure 41 - page 4 of 5



A Multiple Query Algorithm
Figure 41 - page 5 of 5

can be incorporated into this algorithm in a straightforward manner. Some strategic choices in the construction of the charted algorithm were made to simplify programming and to reduce computer time, others were made to illustrate the possibilities for generality and may not be the most efficient choices. The level of detail presented in Figure 41 is intended to aid the serious experimenter in constructing similar algorithms, and may not be of general interest. The algorithm charted may be simple enough to be meaningfully tested in the Cranfield 200 collection. More complex algorithms should be tested with larger query collections so that several examples of each possible alternative in the algorithm are encountered.

Any multiple query search algorithm increases the cost of retrieval by repeated searching of the same set of documents. It might therefore be economical to invest more time in procedures not taking place for each search if this off-line effort would create single query retrieval situations for most users, either in the full collection or in standard subsets of the collection. Better methods of document vector construction and clustering are thus important fields for research.

Two promising off-line techniques that might improve most retrieval situations are discussed briefly. The first of these is the clustering of previous queries suggested by Salton [22]. All original queries submitted to the retrieval system are saved, along with the documents found relevant to each query by the user during the search. When enough queries

have accumulated, the query vectors are clustered by Rocchio's clustering algorithm or a similar method. Then the documents found relevant to the query vectors in each query cluster are grouped and used as a standard cluster for search. The user's query is first compared to the centroid vectors of the query clusters (not of the documents in the cluster) then to the concept vectors of the documents in the standard clusters defined by the query clusters closest to the user's query. Salton mentions that 'request clustering' would provide a means of automatically adjusting the retrieval algorithm to vocabulary shifts in a fast-moving technical field, especially for a document collection that attracts a homogenous user population. Request clustering offers another possible advantage, that the standard clusters of documents are not based on the location of the document vectors in the document space. That is, the documents in the standard cluster defined by a query cluster (the cluster of documents relevant to the queries in the cluster) may not be adjacent in the document space, but may be intermingled with documents from other standard clusters. It now appears that the documents relevant to a query are usually found intermixed with non-relevant documents. Request clustering may offer a greater possibility of simplifying such a retrieval situation than does document clustering.

The idea of request clustering is based on several assumptions. These assumptions deserve review as

indications of initial directions for investigation.

1. It is assumed that the relationship of each document in the collection to one or more request clusters is well-defined. For this assumption to be valid, each document should be relevant to several queries in the clustered sample.

2. For a user environment, it is assumed that relevance information obtained during search is an adequate representation of the needs of the user formulating the query. This may not always be the case, since an inadequate search may fail to reveal relevant documents that the user does not know are available. The appropriateness of the relevance judgments obtained for experimentation is even more important in request clustering than in other retrieval experiments.

3. It is assumed that similar queries have similar sets of relevant documents, and that dissimilar queries tend to have non-overlapping sets of relevant documents. The failure of the first half of this assumption casts doubt on the basic rationale of request clustering. The failure of the second half might mean that a costly degree of standard cluster overlap is unavoidable. Both halves of this assumption can be tested statistically in various document and query collections before request clustering is implemented.

4. If request clustering is used in preference to document clustering, it is assumed that documents retrieved by similar queries are more appropriately related for retrieval purposes than are documents with similar descriptor vectors.

This assumption can only be tested by using the same cluster search algorithm with request clusters and with document clusters. It might be found true for some search algorithms and false for others.

If assumption 4 is true, it suggests that the document vectors should be altered to correspond with the relationships indicated by user requests and relevance judgments. Means of dynamically altering the document space using previous user queries and relevance judgments have been investigated, and two algorithms that permanently alter the document vectors have been suggested. Both algorithms are here discussed.

Davis, Linsky, and Zelkowitz [27] base their approach to document space modification on two assumptions, here quoted:

a) "For a given query, concepts which appear more frequently in relevant documents than in non-relevant documents probably contribute significantly to the relevance of the pertinent documents. The significant concepts are related to one another and often occur in conjunction with one another. Thus by raising the weights of these concepts in all documents within the entire space which contain occurrences of these concepts, similar documents are brought closer together."

b) "Any relevant document (as determined by user feedback) which does not contain an instance of a given concept determined to be significant is likely to contain material which nonetheless relates to this concept. Therefore, this concept is added to that relevant document. It is expected that by increasing the weights of these concepts more relevant documents will be clustered together and ultimately retrieved when a similar query is processed in the future."

The algorithm suggested by Davis, Linsky, and Zelkowitz is almost completely described by their assumptions. From user relevance judgments on the first 15 documents retrieved

by the initial search, a vector of 'significant' concepts is formed by subtracting the sum of the vectors of retrieved non-relevant documents from the sum of the vectors of retrieved relevant documents and setting all negative weights in the resulting vector to zero. This vector is then divided by the sumuof the vectors of all retrieved documents. Each significant concept thus is assigned a positive fractional weight proportional to its significance. Every document in the space is then multiplied by the vector $(\underline{1} + \underline{d})$. That is, every weight assigned to a significant concept i is multiplied by $(1 + d_i)$. Also, each significant concept i is added to every relevant document vector not containing it, in accordance with assumption b).

A closer examination of the assumptions quoted predicts the effects of the resulting algorithm. Assumption a) states that because concept i is important in distinguishing the retrieved relevant documents from the retrieved non-relevant documents, the importance of concept i in the document space should be emphasized by raising the weights of every occurrence of concept i. The algorithm based on this assumption tends to increase the correlation coefficients among all documents containing concept i. It tends to decrease the correlation of any document containing concept i with any document not containing concept i, because concept i appears only in the denominator of the cosine correlation between two such documents. However, since the documents used to select concept i as 'significant' are all retrieved by the user's query, all have relatively high correlations with that query; that is, both the relevant and non-relevant documents retrieved contain a relatively high

proportion of the concepts found in the query. Therefore the concepts that best distinguish between retrieved relevant and retrieved non-relevant documents are not likely to be found in the user's query. The suggested algorithm is thus likely to decrease the correlation of most altered documents with the user's query. Assumption b), by suggesting that concept i be added to every retrieved relevant document not containing it, guarantees that the correlation of every retrieved relevant document with the user's query is lowered.

In fact, Davis, Lin*sky, and Zelkowitz report that while their algorithm does bring relevant documents closer together in the document space, it degrades the retrieval performance of the user's query. The three authors then argue that the resulting clustering of relevant documents is a desirable result and that relevance feedback can be used to overcome the initial degradation of performance and provide ultimately better retrieval. In their examples relevance feedback in the modified document space provides better retrieval than relevance feedback in the unmodified space. However, the examples given of document-document correlations show that while some unretrieved relevant documents are brought closer to some retrieved relevant documents and to each other, these affected documents are moved further away from still other relevant documents. This result indicates that not all relevant documents contain the concepts selected as significant discriminators.

Ignoring for the moment the unfortunate reported effects on initial retrieval and examining only assumption a), the

suggested algorithm can be questioned on theoretical grounds. Assumption a) states that all concepts found useful in distinguishing relevant from non-relevant documents in the existing document space should be emphasized by increasing all weights assigned to that concept. The resulting process is essentially circular in that it uses the characteristics of the document vectors to change the document vectors. Imagine an ideal document collection in which every document is equally needed by users and every concept is equally useful in distinguishing among documents. Given a representative sample of information requests, the suggested algorithm would emphasize each concept in turn, resulting in no effective change to the document space. In a typical collection, this algorithm would eventually eliminate concepts that are either relatively useless for discriminating between documents or that are useful only for discriminating among documents not often requested. Only the first of these effects can be considered useful. In short, the suggested algorithm does not accomplish what should be its prime aim, to alter the document space in such a way as to provide retrieval performance closer to that expected by the users.

Brauen, Holt, and Wilcox [28] suggest a simpler document vector modification algorithm that does accomplish this aim. The concept vector of the user's query is added to the vectors of documents selected by the user as relevant to that query.

The document vector modification formula is:

$$d_i = (1 - \alpha)\, d_i + \alpha \bar{q}_o$$

where $\bar{q}_o$ is the user's query vector normalized to be equal in length to the document vector d_i.

This formula does not change the length of the document vector.

A 425 document subset of the Cranfield 1400 collection, in which each document is relevant to at least one query, is used to test this algorithm. The 155 available queries are partitioned randomly in two ways into an update sample of 124 queries and a test sample of 31 queries. For values of α from 0.05 to 0.4, an average 3.3% improvement in normalized recall and 15.5% improvement in normalized precision are obtained. Every improvement is significant at the 1% level, as measured by the T-test. The changes in α cause no significant change in performance.

In one special experiment the modification algorithm is applied to a document space of zero vectors; that is, document vectors are derived from the queries and relevance judgments only. Results approaching the performance in the original document space are obtained, with normalized recall 1.2% lower and normalized precision 13.4% lower than the original results. Since 425 document vectors are being defined entirely by the information contained in only 124 queries, these results are surprisingly good.

The results reported by Brauen, Holt, and Wilcox indicate that queries and relevance judgments contain information useful

to future retrieval. The special experiment strongly suggests that given a larger query sample to use for document vector modification, this information may in fact be of more value than that contained in the original document vectors. Thus assumptions 3 and 4, stated in the earlier discussion of request clustering, are supported. Two cautions in generalizing these results to practical retrieval systems are necessary. First, Brauen, Holt, and Wilcox force assumption 1 to be true by their intelligent selection of a document collection. In an actual system, there is no guarantee that every document will be relevant to at least one query in a modification sample. Second, the relevance judgments supplied for experimental evaluation are used for document vector modification even though some of the relevant documents would not have been retrieved by an initial search of the user queries. Therefore assumption 2 is not tested by these experiments. Further investigation of these two assumptions in realistic document and query collections is needed. Nevertheless, the results reported by Brauen, Holt, and Wilcox encourage the investigation not only of document space modification but also of request clustering.

An analogy to document space modification is found in the more fully explored field of adaptive pattern recognition. An adaptive system first described by Nilsson [29] and studied by many later experimenters is directly comparable to the SMART system in several meaningful respects. The task of a pattern recognition system is to assign each pattern presented

to the correct class of patterns; for example, to recognize each spoken word as a 'one', 'two', or other single digit. For each class i of patterns to be recognized, a weight vector w_i is constructed. A pattern x is assigned to class if and only if $w_i \cdot x$ is greater than $w_j \cdot x + \theta$ for all classes j not equal to i. The following adaptive algorithm adjusts the weight vectors to a set of patterns used for 'training'.

If a pattern x belonging to class i is presented to the system, and $w_i \cdot x$ is greater than $w_j \cdot x + \theta$ for all j not equal to i, no adjustment to the weight vectors takes place. However, if for some k not equal to i the dot product $w_i \cdot x$ is less than $w_k \cdot x + \theta$, the pattern x is added to the vector w_i and subtracted from the vector w_k. The parameter θ is greater than 1 and is called the 'training threshold'.

The concept vector of a user's query is analogous to the input pattern in such a pattern recognition system. The query 'pattern' is assigned to a 'class' by the SMART system when a document is selected as relevant to the query. The document vectors thus correspond to the weight vectors w_i. Just as similar patterns are assigned to the same class by the pattern recognition system, similar queries select similar documents in the SMART system. In fact the vector dot product $w_i \cdot x$ equals the sum $\sum_j w_i^j x^j$, and therefore corresponds exactly to the cosine correlation coefficient whenever the two vectors are of length 1.

The one weakness in the suggested analogy is obvious - each query is expected to select more than one document as 'relevant', while each pattern is assigned to only one class. Nevertheless, a 'training algorithm' for document vectors can be constructed

that should improve this 'multi-class assignment' in the same way that the adaptive pattern recognition algorithm improves single class assignment. Such an algorithm would modify the SMART document vectors using queries as patterns and relevant documents as 'correct responses'. An adaptive algorithm for document space modification is here stated in information retrieval terminology:

Given a set of user queries with relevance judgments, the document vectors are altered as follows:

If for all i such that document i is relevant to the submitted query q_o, and for all j such that document j is not relevant to q_o,

$\cos(d_i, q_o)$ is greater than $\cos(d_j, q_o) + \theta$, no adjustment to the document vectors is made. However, if this condition does not hold each vector d_i denoting a relevant document i is processed in order of its correlation with q_o as follows:

If there exists a document k such that vector d_k has not yet been adjusted by this query q_o and k is not relevant to q_o, and $\cos(d_k, q_o) + \theta$ is greater than $\cos(d_i, q_o)$, then the query q_o is added to the vector d_i and subtracted from the vector d_k having the highest correlation with q_o. If there exists no non-relevant document meeting all these requirements, butthere exists a document k with previously adjusted vector d_k meeting the other requirements, the query q_o is added to d_i but not subtracted from d_k. The suggested

order of processing insures that a different non-relevant document is decremented for every relevant document incremented whenever this is possible. The suggested multi-class adaptive algorithm is more cautious than the single-class adaptive algorithm in that the vector associated with each correct response is incremented only once while in the pattern recognition algorithm the vector associated with the single correct response is incremented once for each incorrect response that is decremented. Also, the single-class adaptive algorithm decrements all incorrect responses that are stronger than the correct response. In a document retrieval situation, a similar procedure could decrement every non-relevant document in the space. The algorithm suggested above limits the number of document vectors decremented to the number of relevant document vectors incremented. An alternative way to limit negative document vector adjustment is to decrement the vectors of all non-relevant documents retrieved within the first n. If computing time allowed, this document space modification algorithm could adapt during any relevance feedback, algorithm suggested in this study by incrementing the retrieved relevant document vectors and decrementing the retrieved non-relevant document vectors. Only the initial query rather than the queries modified by relevance feedback should be used to adjust the document vectors.

In a document retrieval application, some means of controlling the length of the modified vectors is needed. A

decrementing formula analogous to the length-preserving incrementing formula suggested by Brauen, Holt, and Wilcox, is $d_i = (1 + \alpha)\, d_i - \alpha \bar{q}_o$.

The further investigation of the adaptive document space modification algorithm suggested is encouraged by two findings, the sucdessful performance of the analogous single-class adaptive algorithm in many different pattern recognition applications, and the improved results reported by Brauen, Holt and Wilcox, whose algorithm increments relevant document vectors in the same manner as the suggested algorithm without adjusting non-relevant document vectors. Since the algorithm suggested discourages incorrect responses as well as encouraging correct responses, it should be even more effective than the Brauen, Holt, and Wilcox algorithm in adjusting the responses of the retrieval system to the expectations of its users.

In this final section of this thesis, implications for future research are drawn from the conclusion reached in Section VII-C that the documents relevant to one query are not normally clustered in an exclusive area of the document space. With reference to partial search algorithms, new measures for evaluating the potential usefulness of a given partition of the document collection regardless of the search algorithm used are suggested. The cluster search algorithm is shown to be inappropriate in environments similar to the experimental collection, and a better cluster search algorithm is proposed. A combination of cluster search with relevance feedback that constructs a separate feedback query to search each cluster is supported as a possible solution to the problem

posed by separated groups of relevant documents. If cluster feedback is found inadequate, strategies that construct more than one query to search the same set of documents are shown to be necessary. Several suggestions for the design of such multiple query algorithms are made, culminating in a detailed flowchart of an algorithm that can be meaningfully tested in the Cranfield 200 collection. Finally, request clustering and permanent document space modification are discussed as ways of possibly providing single query retrieval situations for most users by investing time in off-line processing rather than lengthening the search process. An algorithm for adaptive document space modification using queries and relevance judgments is constructed by analogy to a well-tested method that performs a similar function in adaptive pattern recognition systems.