III. Prior Investigations
Of The Relevance Feedback Retrieval Algorithm

Rocchio [8,9,10] suggests an algorithm for relevance feedback based on the properties of the distance function used. If the set of relevant documents is known, the query that will be "closest" to this set of documents and furthest from the set of non-relevant documents can be formed. If the cosine correlation is used as a distance function, this ideal query is

$$q = N_s \sum_{1}^{N_r} \frac{r^i}{\sqrt{(r^i)^2}} - N_r \sum_{1}^{N_s} \frac{s^i}{\sqrt{(s^i)^2}}$$

where each r^i is the vector describing a document relevant to the user's query, and each s^i is the vector describing a document not relevant. Thus N_r is the number of documents in the collection that are relevant to the request, and N_s is the number of non-relevant documents, or the remainder of the collection.

This ideal query is useless for retrieval, because if the documents relevant to each request were known, a retrieval operation would not be needed. Rocchio suggests that the ideal query might be approached by iteration. The user is asked to make relevance judgments on a small retrieved set of documents, and this set is used to update the former query as follows:

$$q_{i+1} = n_r n_s q_i + n_s \sum_{1}^{n_r} \frac{r^i}{\sqrt{(r^i)^2}} - n_r \sum_{1}^{n_s} \frac{s^i}{\sqrt{(s^i)^2}} \qquad (A)$$

where n_r and n_s are the numbers of relevant and non-relevant documents retrieved by the previous ~~query.~~ search operation.

Rocchio investigated relevance feedback using formula A and the SMART retrieval system [9]. A set of seventeen natural language search requests and a collection of 405 abstracts of articles published in IRE Transactions on Electronic Computers (March-September, 1958) were indexed using a SMART regular thesaurus (Section II). Relevance judgments for the sample queries were constructed by a manual search of the entire document collection. Average retrieval results for the collection described are improved by two iterations of the relevance feedback process described in formula A. Rocchio suggests construction of multiple queries when the documents desired are not clustered in the document vector space.

Another investigation of a relevance feedback system was based on the "ADI collection", a collection of 82 documents presented at a conference on documentation. Thirty-five queries were constructed for this collection, and the documents considered relevant to those requests were specified by the two originators of the queries. The investigation of relevance feedback in the ADI collection was conducted by Riddle, Horwitz, and Dietz [11]. They used 22 of the 35 queries and studied a slightly different algorithm for modifying the search query. Their formula is:

$$Q_i+1 = Q_i + \alpha \sum_{1}^{n_r} r_i \qquad \text{(B)}$$

Three differences from Rocchio's formula are immediately apparent:

a) The descriptor vectors are not normalized by their length. In Rocchio's formula, the change to the weight of concept "a" in the query depends not only on the weight assigned to concept "a" in a retrieved document vector but also on the length of that document vector; that is, on the number of other concepts and on the magnitudes of weights in the document vector. This is not the case in the Riddle, Horwitz, and Dietz formula. When the latter formula is used, for instance, a document with generally highly weighted concepts changes the query more than does a document with generally lower weighted concepts, the number of concepts being equal. Weight magnitudes being roughly equal, a document with more concepts changes the query more than one with fewer. Rocchio's formula compensates for these effects.

b) The parameter α, which is the one variable in the above formula, is constant for all queries. Rocchio's formula uses a different multiplier for each query; the multiplier being dependent on the numbers of relevant and non-relevant documents retrieved (n_r and n_s).

c) The non-relevant documents retrieved on the previous iterations are not used to update the query. However, Riddle, Horwitz, and Dietz tested a "negative heuristic strategy" which uses the two non-relevant documents first retrieved (the two which the system falsely judges most relevant to the query) to update those queries that retrieve no further relevant documents on the first feedback iteration. For such queries the formula becomes:

$$Q_{i+1} = Q_i + \alpha \sum_{1}^{n_r} r_i - \sum_{1}^{2} s_i \quad (C)$$

The feedback algorithm of Riddle, Horwitz, and Dietz produces an improvement in performance on most of the queries tested. The three experimenters recommend that the variable α in their formula be set to 1 for the first iteration and then increased by 1 for each subsequent iteration (called "increasing alpha strategy"). They also recommend their negative heuristic strategy (formula C).

Crawford and Melzer [12] have tested a relevance feedback strategy that ignores the original query after the initial search if a relevant document is found. Their algorithm is:

If at least one relevant document is retrieved within the first n documents, the original query is ignored and one additional relevant document is used for each iteration:

$$Q_{i+1} = \sum_{1}^{i} r_i$$

But if no relevant documents are retrieved within the first n

$$Q_2 = Q_1 - s_1$$

On iterations after the first, the second formula of their strategy is not used.

Using the Cranfield 200 document collection (Section IV), Crawford and Melzer found their strategy superior to formula B when $\alpha = 1$.

Steinbuhler and Aleta [13] have tested Rocchio's algorithm in the ADI collection, for the 'worst case' when only non-relevant documents are available for feedback after the first search operation. The information from non-relevant documents alone, when used to modify the query according to formula A, gives better retrieval than the initial query.

Kelly [14] proposes an addition to the relevance feedback algorithm when no relevant documents are retrieved. He points out that in these cases no new concepts are added to the query, and recommends adding concepts that occur frequently in the document collection. He tested this recommendation on sets of artificially constructed 'query' and 'document' vectors with success. However, Steinbuhler and Aleta [13] found that adding frequent concepts to the query degraded performance in the ADI collection.

The results reported in Section VI of this study give further insight into the problems investigated by the five earlier studies cited. Section VII uses both the earlier studies and the present report to support recommendations for document retrieval systems.