

CHAPTER 4

THE QUERY-DOCUMENT MATCHING FUNCTION

1. The Comparison of Structural Operands

In selecting references from a library collection, the user matches his information needs against the discernable information content of source documents (or tokens representing them). In a mechanized document retrieval system an analogous process is implemented using the formal representations (index transforms) of the user's information requirements and the content of reference documents. It is difficult to characterize precisely the nature of the comparisons which the user has at his disposal because of the richness in information carrying elements present in the natural language and because of the complexity of human decision making. In automatic systems, however, comparison operations are closely related to the structure of the data representations of the compared items. In an automatic document retrieval system, then, the criteria for selecting reference documents in response to user queries are directly related to the data structures produced by the index transformation.

A variety of data structures has been considered for information representations in document retrieval (see Chapter 2). Perhaps the simplest of these is the unordered collection of elements such as results with a keyword representation of document content. When both the query and document representations are sets from a finite

collection of keywords, the comparison operation may take a variety of forms, as summarized in Table 4.1. In general, selection of reference documents based on an equality match between the query and document set images is too restrictive for practical consideration. Partitioning a reference collection into retrieved and rejected subsets by the inclusion relation, however, has been used in many practical retrieval systems. In this case selected documents (members of the retrieved subset R) are defined by:

$$R = \{d_i\} : d_i \supseteq q$$

where d_i and q are keyword sets, and R is a subset of the source collection of document images D . Figure 4.1(b) illustrates this process for the collection and query of part (a).

In many instances it is desirable to have the response of the retrieval system be an assignment of values to all documents in the collection, where the values reflect relevance to the query. Both the overlap and metric distance functions are typical of the matching operations of this type. The overlap coefficient merely measures the number of common elements in the two object sets, whereas the distance function (developed by Rial, reference 1) induces a measure with the metric properties of ordinary distance. Figure 4.1 (c) and (d) provides an illustration of values assigned by each of these comparison operations.

An extension of the above matching operations on set-represented operands can be made by exploiting the isomorphism of a

Comparison Operation	Definition
Equality	$A = B \quad a \in A \Leftrightarrow a \in B$
Inclusion	$A \subset B \quad a \in A \Rightarrow a \in B$
Overlap Correlation	$\rho(A, B) = n(A \cap B)$
Metric Distance	$\mathcal{D}(A, B) = 1 - (n(A \cap B) / n(A \cup B))$

Comparison Operations on Set Represented Operands

Table 4.1

Boolean algebra to the partially ordered system formed by the subsets of the keyword set and the set inclusion relation. This allows one to structure the representation of a search request in the form of a Boolean combination of keywords, i.e. $q = \omega(k_i)$ as opposed to using an unordered keyword set representation. Let column i of the keyword document matrix (Figure 4.1(a)) represent the document subset of the i th keyword. The retrieval operation, then, consists in generating the retrieved subset R by replacing each keyword in the Boolean query polynomial by its keyword set and substituting set intersection for Boolean "and" and set union for Boolean "or". With this transformation the subset R is specified by:

$$R = \omega'(K_i)$$

		Keywords						
		a	b	c	d	e	f	g
documents	d_1	1		1		1	1	
	d_2		1	1		1		1
	d_3	1	1	1	1	1		
	d_4	1		1	1			
query	q		1	1		1		

- a) Query and document keyword images represented by a binary occurrence matrix.

$$\begin{array}{l}
 d_1 \not\supset q \\
 d_2 \supset q \\
 d_3 \supset q \\
 d_4 \not\supset q
 \end{array}
 \quad \therefore \quad R = \{d_i\} : d_i \supset q \Rightarrow R = \{d_2, d_3\}$$

- b) Retrieval by set inclusion matching.

	$n(q \cap d)$
d_1	2
d_2	3
d_3	3
d_4	1

	$n(q \cap d)/n(q \cup d)$
d_1	2 / 5
d_2	3 / 4
d_3	3 / 5
d_4	1 / 5

- c) Relevance values assigned by overlap correlation.

- d) Relevance values assigned by metric corr. (1-metric dist.)

Set Image Matching Operations

Figure 4.1

Figure 4.2 illustrates this process for two Boolean queries and the collection described in Figure 4.1(a).

Another of the operand structures useful for document and query representations is the N-dimensional cartesian vector. Table 4.2 characterizes some of the vector comparison operations of interest. Equality, as in the case of set represented operands, is too restrictive a criterion for selecting source documents in response to an input query. The vector difference assigns a vector quantity to each query-document pair, but its magnitude could be a useful matching criterion. In most cases, however, and particularly in the case of the index images derived by a frequency counting technique (see Chapter 2), the information in the vector image of interest is contained in the relative magnitude of its components rather than in their absolute magnitudes. This results from the direct dependence of the absolute magnitude on the number of words in the input text. With this assumption, the angular distance function provides the most suitable matching operation for vector structured information representations.

Data representations with structures considerably more complex than set or vector operands have also been considered for automatic document retrieval systems. Hierarchical arrays,² tree structures,³ and abstract graphs,⁴ are among these. With information representations of these types, matching operations are considerably more complex than those described above (see for example Sussenguth, reference 4, for a detailed account of graph matching procedures). The price paid, then, for the additional information which can be

$$q_1 = b \wedge c \wedge e$$

$$R = \{d_2, d_3\} \cap \{d_1, d_2, d_3, d_4\} \cap \{d_1, d_2, d_3\}$$

$$R = \{d_2, d_3\}$$

a) Retrieval on the Boolean Polynomial $q_1 = b \wedge c \wedge e$

$$q_2 = e \wedge (f \vee g)$$

$$R = \{d_1, d_2, d_3\} \cap \left[\{d_1\} \cup \{d_2\} \right]$$

$$R = \{d_1, d_2\}$$

b) Retrieval on the Boolean Polynomial $q_2 = e \wedge (f \vee g)$

Retrieval Operations with Boolean Query Images

Figure 4.2

Comparison Operation	Definition
Equality	$\bar{a} = \bar{b} \quad a_i = b_i \text{ for } i = 1, N$
Vector Difference	$\bar{d} = \bar{a} - \bar{b} \quad d_i = a_i - b_i \text{ for } i = 1, N$
Magnitude of the Vector Difference	$\mathcal{J} = \bar{d} = \left[\sum (a_i - b_i)^2 \right]^{\frac{1}{2}}$
Angular Distance	$\Theta = \cos^{-1} \frac{\bar{a} \cdot \bar{b}}{ \bar{a} \bar{b} }$

Comparison Operations on Vector Represented Operands

Table 4.2

carried by these more complex operand structures is the increased cost in the required comparison operations necessary to specify a retrieved subset or to assign a value indicative of document relevance. The discussion here will be primarily concerned with vector operands; however, certain of the results derived will be a function not of the operand structures but of the matching function itself, and will, therefore, be applicable to matching functions of the type considered regardless of the operands to which they are applied.

2. Storage Organization

In principle, an automatic document retrieval system can be characterized independently of any parameters of storage organization. Given a description of the document and query representations and of

the matching function used to implement selection or ranking, a retrieval operation is uniquely specified. Each input query is matched with every reference document to specify the retrieved output. Matching a user's search request against the full store of document index images exploits, in effect, the maximum capabilities of the system. For any but limited collections, however, the complexity of effective matching operations make a full search impractical. Useful retrieval systems are then required to impose some organization to the document store so as to limit the scope of the search to a document subset of manageable size.

The necessity for storage organization in fact is likely to become more stringent as research on automatic document retrieval progresses. Advances in the techniques of automatic content analysis are likely to lead to more complex index representations capable of carrying more information. Such index representations, while allowing for finer retrieval distinctions necessarily require more time for each basic comparison operation. In addition, the introduction of operationally effective time-shared computer systems is likely to produce significant changes in the organization of document retrieval systems. In a real time environment the response time of the system to the user's demand plays a critical role on overall system performance. As the time per query-document comparison increases due to increased information in the index representations, the number of comparisons possible per unit time decreases. Thus, even with the increasing speed of information processing equipment, these factors suggest that some

form of storage organization or document classification will be necessary to achieve economic retrieval from large collections with response times fast enough for a real time environment.

Classification may be regarded as a part of the general problem of content analysis. When a document is classified under some given subject heading, its information content has been found to be related to that area of discourse. A classification system, however, is rarely used for retrieval in the sense that a user can be satisfied by all the references assigned to some given category. The classification schedule in general provides a means of storage organization which allows a user to limit the scope of his search. In this sense the process of document classification is analagous to the document indexing process. The index image of a document characterizes the information content of that document while a classification category normally characterizes the information content of some area of discourse in the general field of knowledge. The assignment of some set of documents to a category then, in effect, creates an index image for the information content of the entire set. The user matches his information needs against the categories of the classification system to select subsets of documents in the same way in which his search request is matched with individual document representations to select particular references. Thus in automatic document retrieval systems, as in conventional library systems, document classification provides the key for a storage organization which can effectively limit the number of references which must be examined in detail in a given retrieval operation.

To formalize the effect of storage organization on the document retrieval operation some definitions are required. In the base retrieval system (full query-document searching) a total of N comparison operations are required for each input query, where N is the number of documents in the reference collection. Let the number of comparisons required per input query with a classification induced storage organization be N_c . The relative search efficiency with classification may then be defined as the ratio N/N_c . Further, assume that the document set retrieved with a limited search R_c is a subset of the retrieved set R produced by a full search. This is a natural consequence of assuming that the retrieval criterion applied to the query-document comparisons is the same for both systems. Thus some documents which would satisfy the retrieval criterion in a full search may not be examined in the reduced search mode and therefore cannot be members of R_c . In general, then, there is a cost associated with a limited search in terms of documents which would be retrieved by a full search but which are not retrieved by the reduced search (members of the set $R - R_c$).^x More precisely, if D_R is the document set relevant to the input query, the cost of an increase in search efficiency is a function of the size of the set $D_R \cap (R - R_c)$, i.e. the number of relevant documents lost.

Statistically, then, a retrieval system with a classification induced storage organization may be characterized by an expected

^x $R - R_c$ is defined as $\{d_i \in R : d_i \notin R_c\}$

increase in search efficiency of N/\bar{N}_c where \bar{N}_c is the average number of comparisons required per input query. In addition, an expected relative loss of relevant documents equal to the expectation of $n(D_R \cap (R - R_c))/n(D_R)$ will accompany the increase in search efficiency. If the relative cost of a query-document comparison is c_1 , and the relative value of retrieving a relevant document is c_2 the search cost per input query of the base system may be expressed as:

$$C_s = c_1 N - c_2 n(D_R \cap R) ;$$

whereas with reduced searching, the search cost per query is:

$$C_s' = c_1 N_c - c_2 n(D_R \cap R_c) .$$

That system of storage organization which minimizes the expectation of C_s' over the population of input queries may be defined as the optimal reduced search strategy. Further, any reduced search strategy for which the expectation of C_s' is less than the expectation of C_s , may be used to provide a net gain to the retrieval system user.

The above cost expressions are oversimplifications since the relative costs are subjectively variable from user to user and since the total costs are probably not linear functions as assumed. But this formulation provides basic insight into the potential gain which may be realized from a classification induced storing organization in a document retrieval system.

3. Automatic Document Classification

Traditionally the creation of classification schedules aims at producing a logically consistent, intelligible structuring of human knowledge, wherein the organization and structural relations among subject categories reflect meaningful relations among the fields of discourse which they represent. Research in automatic classification is generally limited to a much narrower set of goals. In particular, automatic classification techniques have, in general, been based on the state or content of a given collection rather than on the state of knowledge in a given field. In this sense, then, the object of automatic classification has been to generate a set of categories which are in some sense optimal for the collection at hand.⁵

The emphasis of this chapter is placed on the relation of automatic classification to the problem of search optimization in an automatic document retrieval system. To this end, then, the basis for establishing the set of classification categories of a given collection is specifically identified with increasing the search efficiency of retrieval operations.

Previous investigations into the feasibility of automatic classification have regarded the generation of a set of classification categories, or the automatic assignment of documents into an existing classification schedule, or both, as primary goals.^{5,6} The interest here, however, is not in the classification system as an end in itself, but rather as an adjunct to an automatic retrieval system.⁷ For present purposes, then, there need be no a priori constraints on the nature of

classification categories. Thus for example, there may be no requirements for such categories to be intelligible to the users of the system if, in effect, they are used for purely internal storage organization. The tailoring of the classification process to the internal document searching operations of the retrieval system offers, then, an increased degree of flexibility which can be exploited to optimize the overall search strategy.

For present purposes, the following assumptions summarize the role of automatic document classification in a mechanized retrieval system:

1. The discernable information content of source documents which serve as the basis for classification is contained in the collection of index images to be used for detailed query-document comparison.
2. The objective of classification is to induce a storage organization which allows a limited search to retrieve the same documents as would be retrieved by a search of the full source collection.
3. The characteristics of the classification should be such that it jointly maximizes the search efficiency of the system and minimizes the associated loss of relevant documents.

On the basis of these assumptions it is clear that the nature of the query-document matching function is critical to the automatic classification process. In particular, to satisfy the objective

stated in assumption 2, a classification category should be an equivalence class with respect to the retrieval function.

Let \mathcal{R} be a relation on the set of document images such that

$$d_i \mathcal{R} d_j$$

if and only if every query which retrieves d_i also retrieves d_j .

Assume then that the nature of the query-document matching function is such that the relation \mathcal{R} defined above is an equivalence relation (reflexive, symmetric, and transitive). Further, let the equivalence classes it induces on the document set be identified as classification categories. Under these circumstances it is necessary only to match an input query against a single member of each equivalence class, and to retrieve all the members of those classes which satisfy the matching function. In this manner, a reduced search retrieves exactly the same document subset as a full search.

The only matching functions considered here, which result in \mathcal{R} being an equivalence relation are set equality for set represented index images and vector equality for vector index images. The equivalence classes for these comparison operations, however, result in a trivial partition of the reference collection since each class is a singleton. Thus the search on equivalence classes in this case is identical to a full search over all reference documents.

Consider, for example, the set inclusion matching function for which the retrieved set is defined by:

$$R = \{d_i\} : d_i \supseteq q .$$

Consider two document images such that:

$$d_1 \supseteq q \text{ and } d_2 \supseteq q$$

for arbitrary q . Then if $d_1 \neq d_2$ either:

$$d_1 \supset d_2 ; \text{ or } d_2 \supset d_1 ; \text{ or } d_1 \not\supset d_2 \text{ and } d_2 \not\supset d_1$$

The equality case has already been considered; any of the other three possibilities lead to the existence of some query which will retrieve one but not the other of the documents d_1 and d_2 . Thus in general, the relation \mathcal{R} induced by set inclusion matching is not an equivalence relation.

In the case of metric distance matching (of set or vector represented index images), define the retrieved set R by the condition:

$$R = \{d_i\} : \delta(q, d_i) \leq \delta_0$$

Now consider two documents, d_1 and d_2 , such that:

$$\delta(d_1, d_2) = \varepsilon$$

where ε is the smallest distance possible in terms of the quantization

employed in the index space. Under these circumstances there always exists some query q_0 such that

$$\mathcal{J}(q_0, d_1) = \mathcal{J}_0$$

and

$$\mathcal{J}(q_0, d_2) = \mathcal{J}_0 + \epsilon$$

so that regardless of how close two document images are, they do not belong to an equivalence class with respect to retrieval unless they are in fact identical.

Under these circumstances it is clear that in order to reduce the number of comparisons required in a retrieval operation, it will be necessary to introduce some finite probability of error. Thus, since the classification categories cannot be identified with equivalence classes under matching functions of interest, a limited search strategy may fail to retrieve some documents which would be retrieved by a full search over the entire collection. The design of a classification system, then, must involve a tradeoff between the total number of comparisons (search efficiency) and the probability of loss of relevant documents (versus retrieval by a full search).

4. Classification and Metric Searching

The two previously considered metric query-document matching functions did not lead to an equivalence class partition of the reference collection. Metric comparison measures do, however, have a

particular significance to automatic classification of the type being considered. The similarity measures:

$$\mathcal{J}(q,d) = 1 - \frac{n(q \cap d)}{n(q \cup d)}$$

for set represented index images and $|\Theta|$ where

$$\Theta(\bar{q}, \bar{d}) = \cos^{-1} \rho(\bar{q}, \bar{d}), \quad -180^\circ \leq \Theta \leq 180^\circ \quad (4.1)$$

and

$$\rho(\bar{q}, \bar{d}) = \frac{\bar{q} \cdot \bar{d}}{|\bar{q}| |\bar{d}|}$$

for vector represented index images are special cases of a class of matching functions which possess the so-called "metric" property of ordinary distance. A metric is characterized by:

- (i) $\mathcal{J}(\alpha, \beta) = 0$ iff $\alpha = \beta$ while $\mathcal{J}(\alpha, \beta) > 0$ if $\alpha \neq \beta$
- (ii) $\mathcal{J}(\alpha, \beta) = \mathcal{J}(\beta, \alpha)$ (symmetry)
- (iii) $\mathcal{J}(\alpha, \beta) + \mathcal{J}(\beta, \gamma) \geq \mathcal{J}(\alpha, \gamma)$ (triangle inequality).

(Since the index images of two distinct documents can in theory be identical, the distance function in this case is more precisely characterized as being a pseudo-metric, i.e. a function satisfying metric properties (ii) and (iii) for which $\mathcal{J}(\alpha, \alpha) = 0$.)

A metric matching function is well suited for automatic classification since it effectively introduces sufficient structure on the index space to allow groups of related documents to be identified. Since the same structure has been assumed for document

index images as for query images, document-document distance is defined and possesses the same properties of query-document distance. By virtue of metric property (iii), the triangle inequality, a search request which is close (related) to a given document d must necessarily be close to all documents which are themselves close to d .

Let a set of documents D_c , grouped as a classification category, be confined to a region of the index space such that:

$$\mathcal{J}(\bar{d}_i, \bar{c}) \leq \mathcal{J}_c, \quad \text{for all } \bar{d}_i \in D_c,$$

where \bar{c} is an arbitrary vector in this region. Let the distance from a query \bar{q} to the vector \bar{c} be

$$\mathcal{J}(\bar{q}, \bar{c}) = \mathcal{J}_0.$$

The metric properties of the distance function allow the distance between \bar{q} and the members of D_c to be bounded as follows:

$$\max [0, (\mathcal{J}_0 - \mathcal{J}_c)] \leq \mathcal{J}(\bar{q}, \bar{d}_i) \leq \mathcal{J}_0 + \mathcal{J}_c,$$

for all $\bar{d}_i \in D_c$. Thus the single distance $\mathcal{J}(\bar{q}, \bar{c})$ provides a bound on the set of distances from \bar{q} to the members of the document set D_c . The following discussion is limited to the vector indexing model and angular distance matching with the understanding that it is generally applicable to any system employing a metric similarity measure.

In the vector model, document or query index images are treated

as N-dimensional Cartesian vectors. Using the angular distance similarity measure, it is clear that a classification category should consist of a set of document images confined within a localized hypercone of the index space. Alternatively, if the index images are pictured as unit vectors terminating on the unit N-sphere, a classification category should consist of a set of documents represented ~~ing~~ ^{by} index vectors terminating within some local area on the surface of the unit N-sphere. In these terms the problem of automatic document classification is to define the characteristics of such areas and to establish a procedure for identifying and representing them.

5. A Heuristic Classification Algorithm

A. Basic Concepts

Associated with an arbitrary set of document index vectors D , a classification vector c is defined by the equation

$$\bar{c} = \frac{1}{n} \sum_{i=1}^n \frac{\bar{d}_i}{|\bar{d}_i|} \quad (4.2)$$

where $D = \{\bar{d}_1, \bar{d}_2, \dots, \bar{d}_n\}$. The vector \bar{c} is the centroid or center of gravity of the set of unit vectors $\bar{d}_i/|\bar{d}_i|$ derived from the elements of D and represents, then, a vector with an orientation for which

$$\sum_{i=1}^n \Theta(\bar{c}, \bar{d}_i) = 0$$

where Θ is defined according to equation (4.1). The classification vector \bar{c} (or more precisely its orientation) is the best single representation for all of the elements in the set D under the assumption that the information carried by an index vector is contained in its angular position.

In the geometrical interpretation, the vector $\bar{c}/|\bar{c}|$ terminates at the centroid of the point distribution on the unit N -sphere representing the vectors $\bar{d}_i/|\bar{d}_i|$. In particular, then, if the elements in D are sufficiently close to one another, \bar{c} must be close to all of them. With respect to the classification problem, if the members of D are to be grouped into a classification category, \bar{c} can be considered to be the best classification "head" or representation for the category. This property of the centroid vector together with the metric properties of angular query-document matching will be used as a basis for an automatic classification algorithm suitable for storage organization in the vector indexing model.

B. Description of the Classification Algorithm

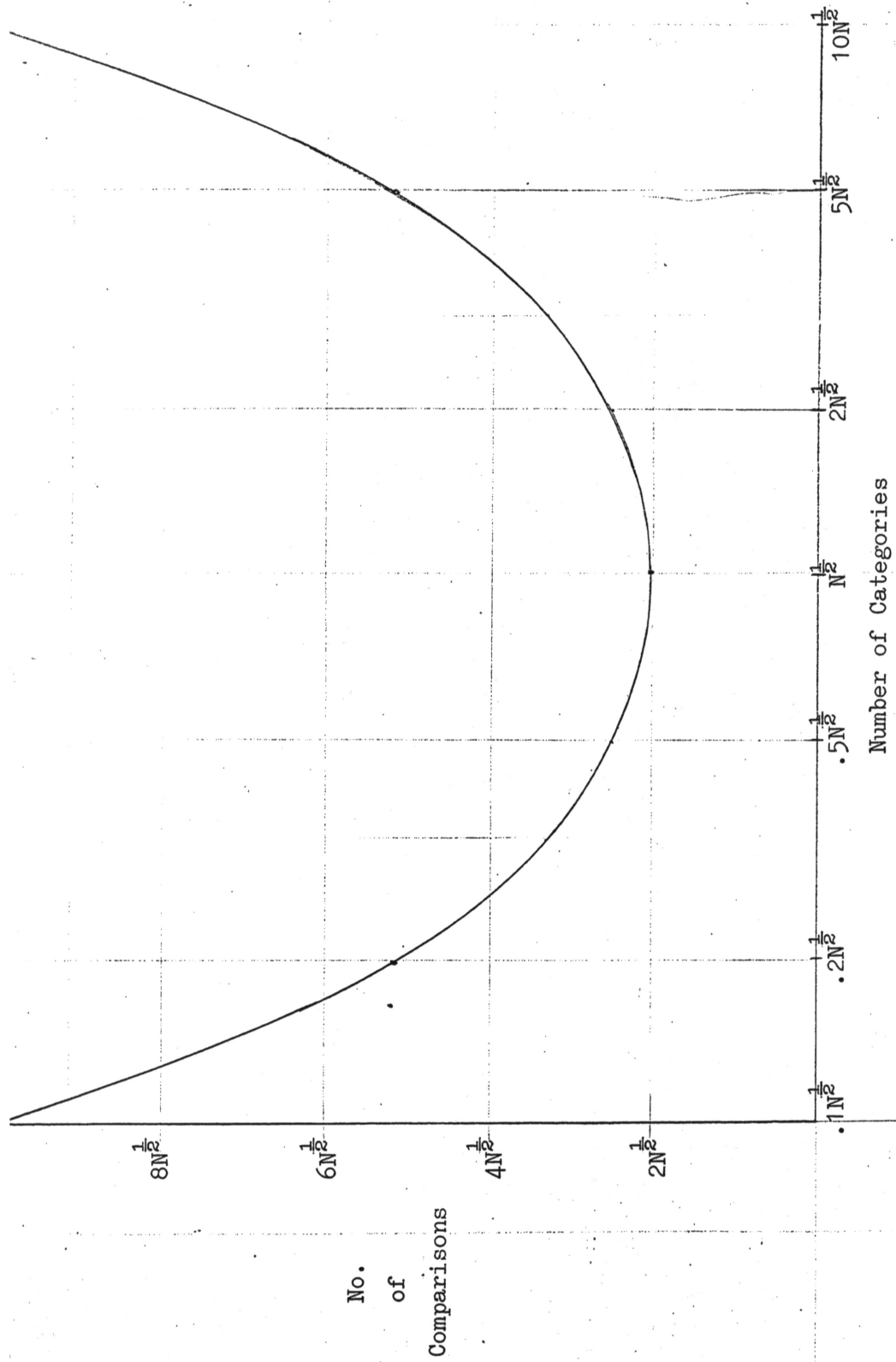
The objective of the classification process is to generate a set of categories or document subsets, each represented by a classification vector (equation (4.2)) from the source collection. The properties of the classification system should result in increased search efficiency in a document retrieval system. The storage organization induced by a classification of this type leads to a two-level search algorithm. Consider an input item which is to be compared with each member of a collection of N elements so that those elements which

satisfy some comparison criterion may be determined. Assume further that the elements can be grouped such that a comparison of the input item with the representation of each group will determine whether any of the group members can satisfy the comparison condition. Under these circumstances, assuming also that k groups are searched in detail, and that the groups are of equal size, the total number of comparisons required, N_t may be written:

$$N_t = x + k \frac{N}{x} ; \quad (4.3)$$

where x is the number of categories and (N/x) is the population of each category. Assuming that all elements of the collection have equal a priori probability of matching an input item, the total number of comparisons N_t is minimized for $x = (kN)^{\frac{1}{2}}$. Thus in an ideal two level storage organization scheme $2(kN)^{\frac{1}{2}}$ comparisons are required versus N for single level searching. The variation of N_t with the number of categories (for $k=1$) is shown in Figure 4.3 on a semi-log plot. Note that the minimum of the total number of comparisons with number of categories in the classification is relatively broad.

The foregoing analysis is applicable to the document retrieval case since all documents can be considered to have equal likelihood of being relevant to an arbitrary search request (at least in the absence of any evidence to the contrary). The classification categories should, therefore, contain approximately the same number of document images. To this end the criteria for identifying a suitable



Total Number of Comparisons Required vs. the Number of Categories

Figure 4.3

subset of index vectors for category formation are based on the number of elements in the subset as well as the mutual distance among the elements. Under these conditions a region of the index space with a high density of document vectors will yield categories in which all the documents are closely related (via the distance function) whereas in regions of relatively low density, categories covering a wider scope will be formed. Note that as the mutual distance among the members of a classification category increases, the classification vector becomes less representative of the group as a whole. There is therefore a definite tradeoff in category formation between producing categories of equal population on the one hand, and maintaining control of the distance relation among category members on the other.

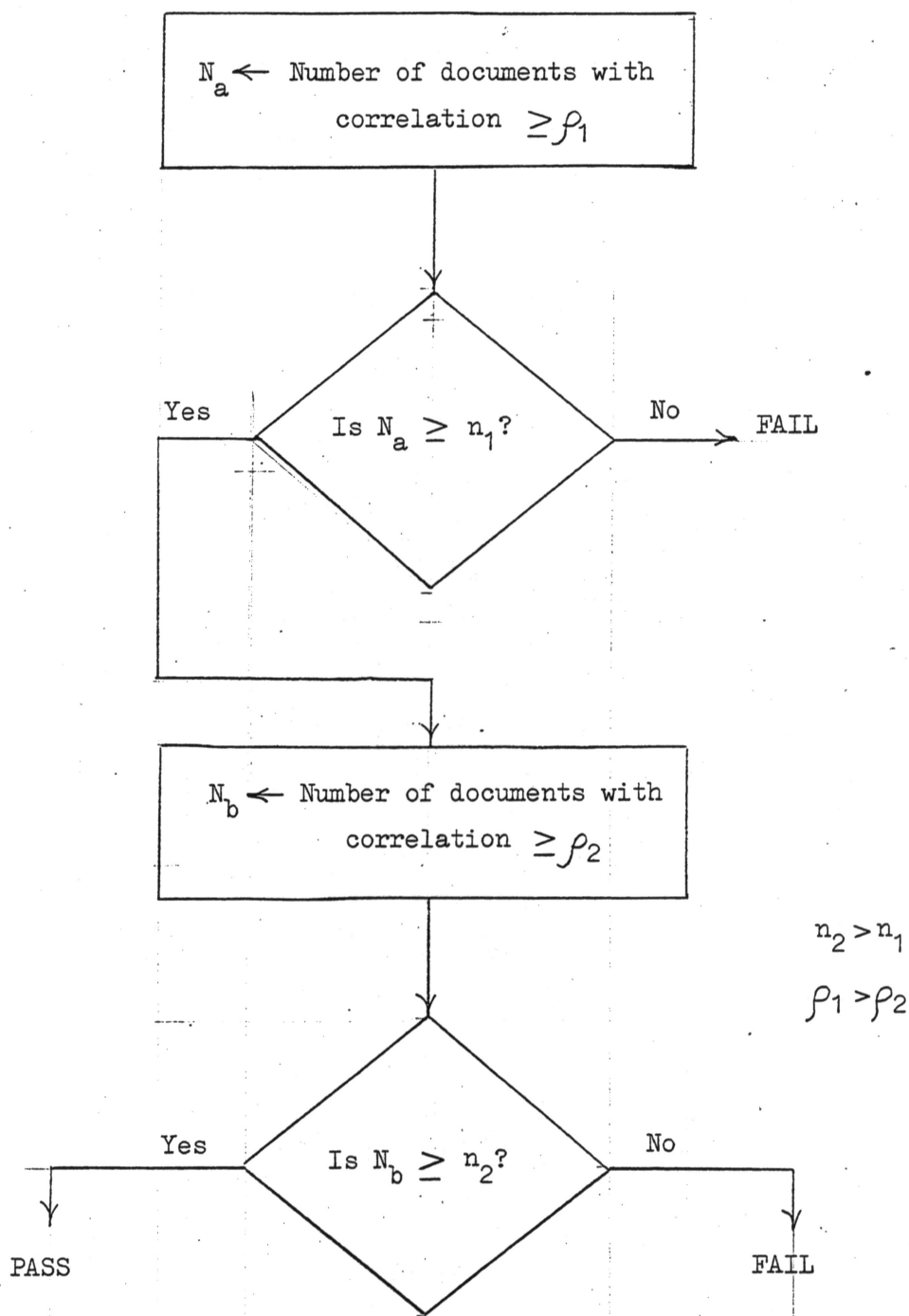
Control of the classification categories is achieved by a set of input parameters to the algorithm which specify:

1. The number of categories desired.
2. A lower and upper bound on the number of elements to be included in any classification subset.
3. An upper bound on the distance (lower bound on the correlation coefficient) between a document and a classification vector such that the document is still considered to be associated with that vector.

In the course of the classification process each document may be associated with one of three possible states. Initially, all documents are considered to be "unclustered", implying that they have not been assigned to any classification category, nor is anything

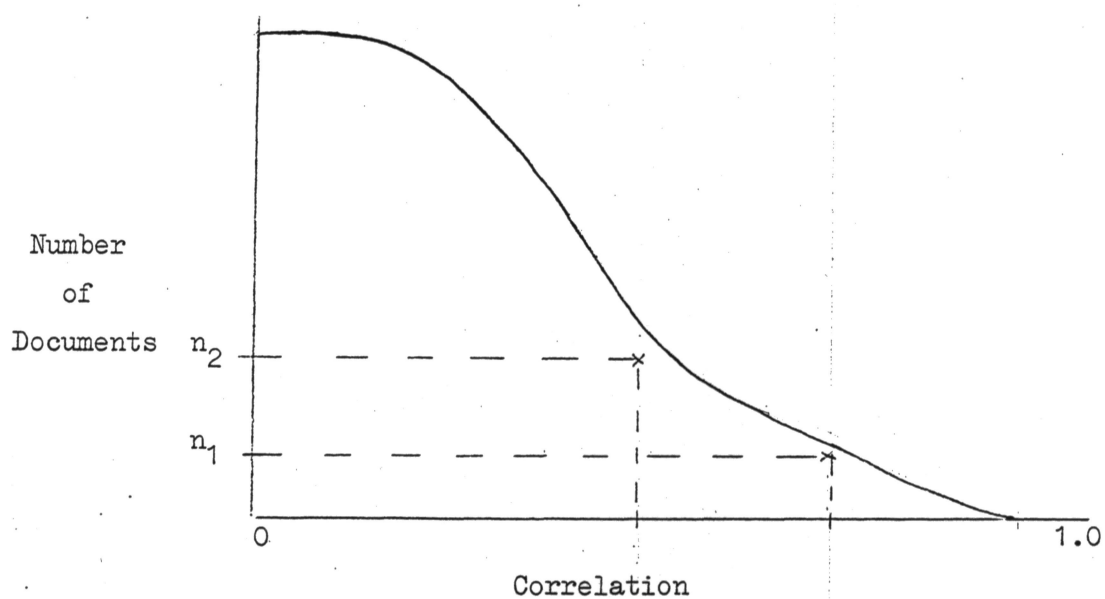
known about their relative position in the index space. As the process develops, a document may become "clustered" i.e. associated with a particular classification vector, or may be identified with the "loose" state indicating that it has been found to be oriented in a region of low density in the index space. Unclustered documents are considered in sequence and the first step consists in generating a measure of the distribution of document images around the document being considered. This is accomplished by correlating this document with all documents except those which are in the clustered state. The resulting correlation distribution is sorted into descending order (note that the correlation is inverse to the angular distance metric), i.e. into order of increasing distance and a density test is applied to determine if the region being considered (defined by the object document plus those unclustered documents in its immediate vicinity) is dense enough for category formation. The density test employed (a flowchart is given in Figure 4.4) requires that the correlation distribution exceed two test points as illustrated by Figure 4.5. This test was chosen heuristically after experimenting with typical document-document correlation distributions.

If the density test fails, the document under consideration is marked "loose" and control returns to step 1 to consider the next unclustered document. If the density test is satisfied, a cutoff correlation is determined as a function of the category size limits and the distribution of correlation values. The cutoff-determining algorithm is illustrated in Figure 4.6. As the documents above the

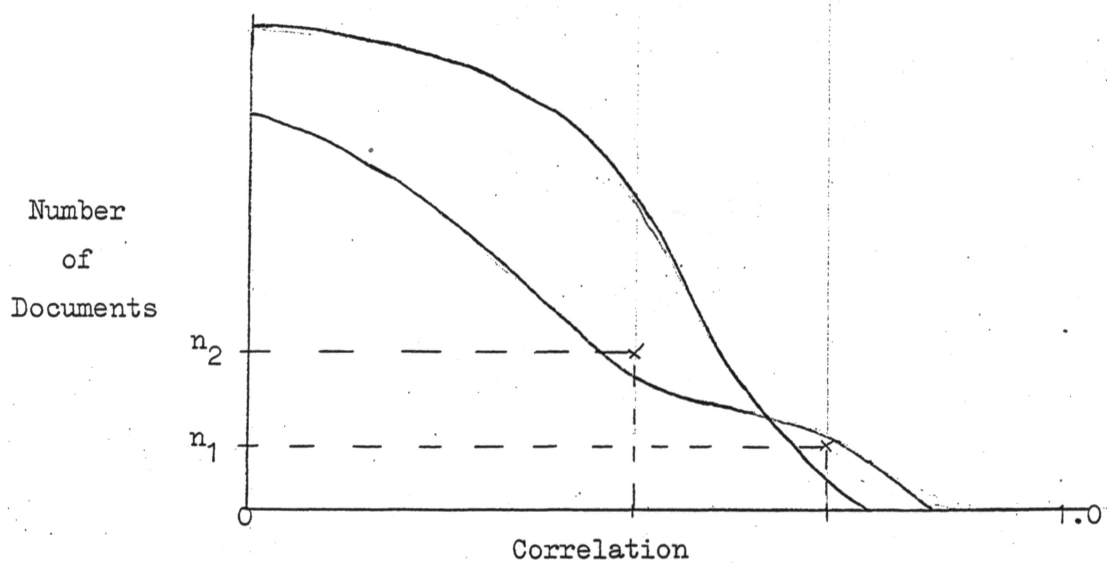


Flowchart of Region Density Test

Figure 4.4



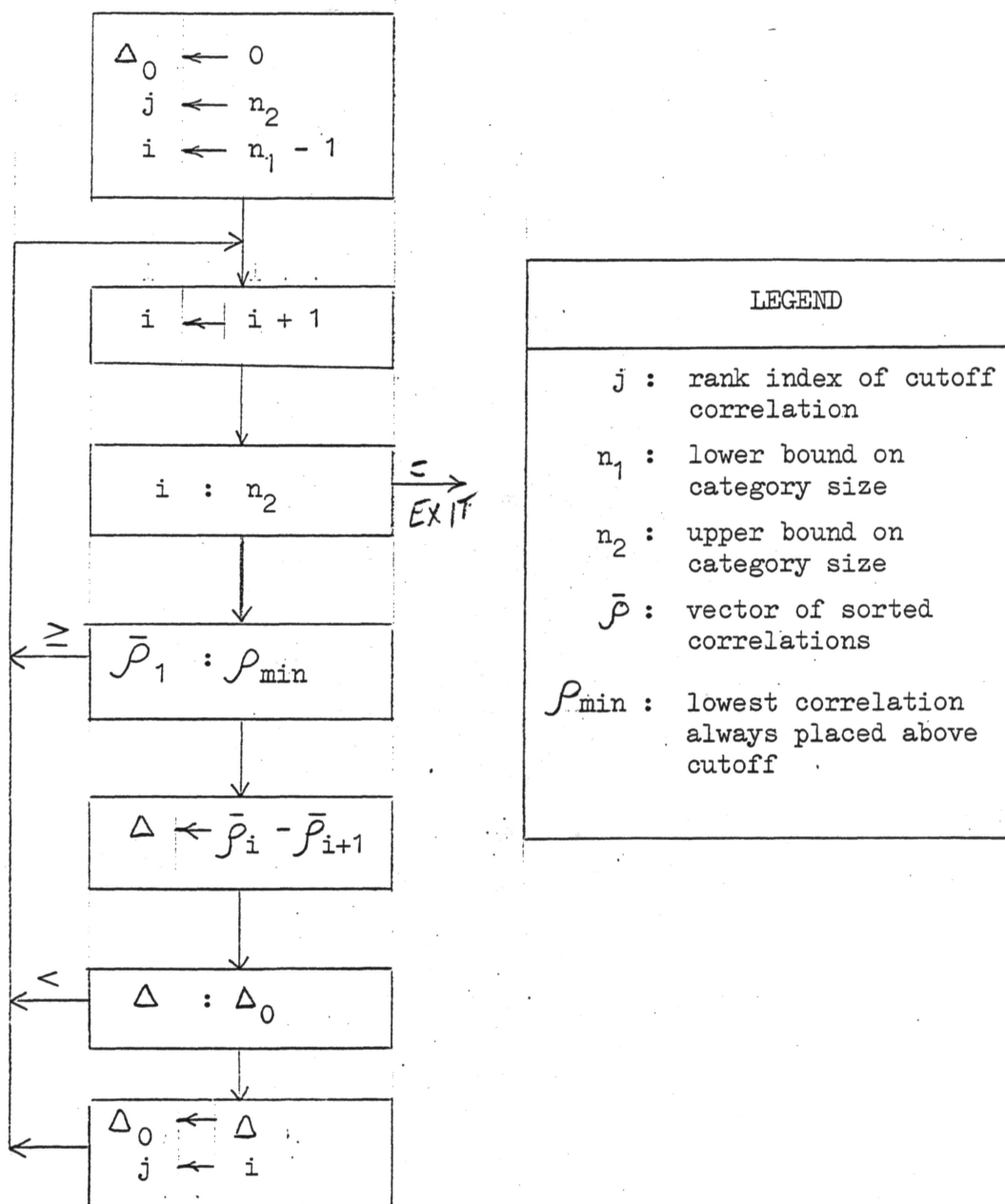
a) Hypothetical Doc.-Doc. Correlation Distribution
Showing Density Test Points



b) Two Distributions which Fail the Density Test

Graphical Illustration of the Density Test

Figure 4.5



Program for Specifying Cutoff Correlation

Figure 4.6

cutoff are to be used as an initial category, this algorithm must accomplish several objectives. First, the subset must be constrained by the maximum and minimum size limits. Further, a region of high document density should yield a larger subset than a region of low density. Thus within the size constraints, documents with correlation above ρ_{\min}^x are automatically placed above cutoff. If the correlations fall below ρ_{\min} before the size limit is exceeded, the cutoff is chosen at the greatest correlation difference (Figure 4.6) in the distribution. This produces, in effect, the sharpest boundary between the identified subset and neighboring unclustered documents. A classification vector following equation (4.2) is now formed for the subset so identified, and a scaled, truncated version of it is then correlated with the entire source collection, thereby identifying documents centered around it. The resultant correlation distribution of the classification vector is sorted into descending order and the cutoff algorithm is reapplied. In this case all documents above cutoff with correlation greater than the minimum clustering correlation (an input parameter) are marked clustered. Documents above the cutoff but with correlation lower than this minimum are marked loose. This prevents such documents which are clearly related to the category just formed (i.e. they are above the cutoff) from becoming candidates for new cluster centers at this stage of the process. At this point control passes to step 1.

This first pass through the collection ends when all documents

ρ_{\min}^x represents a correlation significantly above the average document-document correlation of the collection.

have been either clustered or marked loose. Note that as the documents are clustered, they are removed from the process of identifying initial category subsets. This strategy prevents the generation of classification subsets with large overlap and materially reduces the number of correlations required. Since it is reasonable to expect that some documents should be multiply classified, the classification vectors are themselves correlated with the entire collection. In this manner, previously clustered documents can appear above the cutoff for a given classification vector and thus be associated with more than one category. Figure 4.7(a) illustrates a correlation distribution of unclustered documents which leads to a classification vector (shown in Figure 4.8(a)) and part (b) shows a part of the correlation distribution of this classification vector with the entire collection.

Since there is no a priori way to establish exactly how many categories will be formed by this initial pass through the collection, a second pass is used in case the number formed is less than specified. (Note that more than the specified number of categories could be formed during pass 1, but this would imply that the density test could be made more restrictive or that the category size limit could be increased.) During pass 1, the initial part of the sorted correlation list for all documents failing the density test is saved on tape. In pass 2 this list is scanned and a measure of the unclustered document density around such documents is computed. The maximum values of this measure (which is just the sum of a fixed number of the sorted correlations) are used to ^eselect additional classification regions until the specified number of categories has been formed. The algorithm

Doc. No.	Doc. No.	Doc. No.	Doc. No.	Doc. No.	Doc. No.
52 311 77 53 264 171 174 123 259 47 309 236	Corr. 1.00 .57 .49 .48 .36 .34 .33 .33 .32 .32 .31 .30 cutoff	52 * 177 264 171 123 77 174 * 263 * 127 259 53 * 361 47 225 340 311 * - previously clustered	Corr. .71 .70 .69 .67 .66 .62 .62 .61 .58 .58 .57 .55 .58 .54 .57 .55 .54 .54 cutoff	52 264 171 123 77 174 127 259 53 47 225 311 403 336 345	Corr. .71 .69 .67 .66 .62 .62 .58 .58 .58 .57 .55 .54 .49 .40 .38
a) Initial classification subset. Correl. of doc.#52.		b) Correl. distribution of vector formed from subset of part (a).		c) Partition class of class. vector formed from subset of part (a).	
					d) Correl. distribution of class. vector formed from subset of part (c).

Progression of Categories and Correlation Distributions

Figure 4.7

11 (1)	12 (1)	13 (2)	30 (1)	32 (5)
39 (4)	46 (3)	53 (1)	55 (1)	57 (3)
59 (1)	64 (1)	68 (1)	71 (1)	73 (1)
80 (1)	90 (1)	91 (1)	92 (2)	107 (1)
110 (24)	119 (1)	122 (1)	137 (3)	142 (2)
143 (1)	149 (3)	155 (4)	167 (1)	173 (1)
176 (1)	196 (1)	198 (2)	202 (1)	209 (2)
240 (1)	252 (1)	257 (1)	290 (1)	320 (1)
322 (2)	324 (9)	325 (2)	341 (5)	342 (1)
346 (3)	353 (4)	359 (2)	360 (1)	437 (2)
444 (1)	448 (1)	496 (4)		

- a) Classification vector (in condensed format) of the subset of Figure 4.7 (a). The initial part of the correlation distribution of this vector is shown in part (b) of Figure 4.7.

12 (1)	13 (2)	16 (1)	30 (1)	32 (5)
39 (5)	46 (2)	55 (1)	57 (4)	58 (1)
59 (1)	68 (1)	72 (1)	90 (1)	91 (1)
92 (5)	107 (2)	110 (24)	112 (1)	121 (1)
122 (1)	137 (2)	142 (2)	143 (1)	149 (3)
155 (3)	167 (1)	176 (1)	196 (1)	198 (3)
202 (2)	209 (2)	214 (1)	240 (1)	250 (1)
252 (1)	290 (1)	320 (1)	321 (1)	322 (3)
324 (6)	325 (1)	341 (3)	342 (1)	345 (2)
346 (2)	353 (4)	359 (1)	362 ()	437 (3)
444 (2)	487 (1)	496 (3)		

- b) Classification vector of the partition class shown in Figure 4.7 (c). The initial part of the correlation distribution of this vector is shown in part (d) of Figure 4.7.

Classification Vectors

Figure 4.8

proceeds exactly as in pass 1 except for bypassing the density test. At the end of pass 2, therefore, at least the required number of initial categories have been formed.

It should be clear that at the end of pass 2 not every document has necessarily been used as an element of a classification subset. However, those which have not can be assumed to be document images which are relatively isolated in the index space. In general there are several alternatives for dealing with such documents. In a dynamic environment, i.e. one in which the collection is growing, there will be new documents not yet classified. Isolated documents, then, could be grouped with these in a category which is always searched in detail for all input queries. At periodic intervals all such documents would be entered into the classification system with the possibility of generating new categories as the size of the collection increases. For the current study, however, the elimination of those documents which are in effect hard to classify would bias the evaluation of the overall effectiveness of the technique. The objective here then is to produce a set of categories suitable for all documents in the test collection. To this end a third pass was incorporated into the classification process.

At the completion of pass 2 each source document is assigned to the classification vector with which it has the highest correlation. This assignment induces a partition of the collection such that partition class i contains all documents which are closer to classification vector i than to any other classification vector. In pass 3 each partition class is used as the classification subset for a

with the one on which it is based. Each of these final classification vectors is again correlated with the entire document collection to define the resultant set of categories. At this point a document is associated with a category if it is above the cutoff of the classification vector of that category, or if it is not above any cutoff but is closest to said classification vector. Figure 4.7(c) illustrates the partition class which results in the classification vector of Figure 4.8(b); the correlation distribution of this vector, which specifies the final category, is shown in Figure 4.7(d).

At the end of the classification process, then, each classification vector represents all the documents with index vectors within the angular distance corresponding to its cutoff correlation, and additionally, a few documents outside this radius. Documents of the latter type however, are closer to the vectors to which they are assigned than to any others of the set. Note that the final classification vectors are not necessarily the centroid vectors of the vector subset they represent since the final categories are not in general identical to the partition class from which the centroid vector was formed. However, the final categories generally contain the members of the partition class in addition to documents which are multiply classified. This strategy provides a convenient means for generating multiple classifications for some documents, while maintaining a set of categories balanced over the entire collection. Table 4.3 summarizes the main parts of the classification algorithm and an overall flowchart is given in Figure 4.9.

1. Identify a dense set of unclustered document images.
2. Form the classification or centroid vector for this subset.
3. Identify all documents in the vicinity of the classification vector. Define a category by choosing a cutoff, and cluster documents in the category.

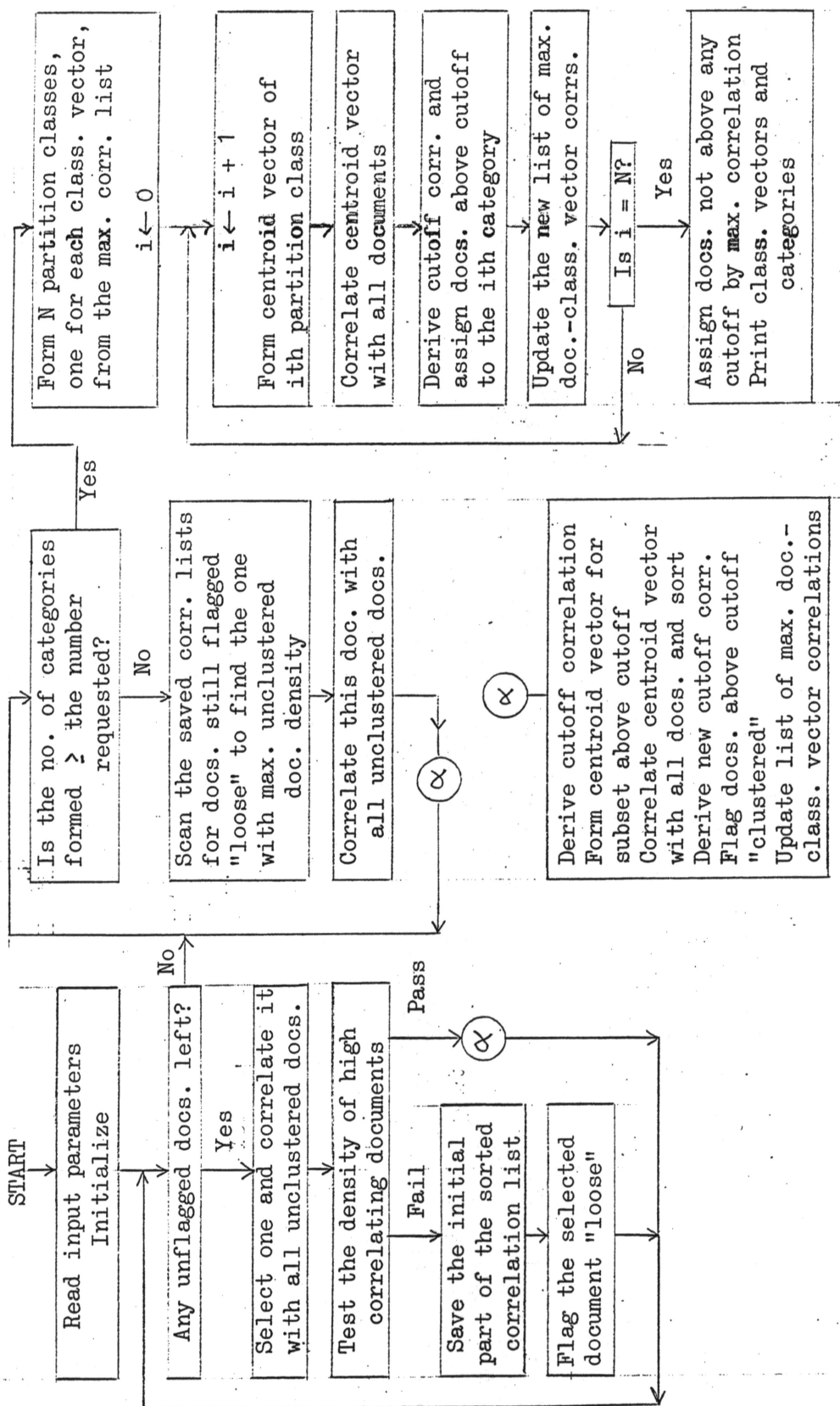
Passes 1 and 2

4. Partition the source collection on the basis of association with the set of classification vectors formed above.
5. Form the classification or centroid vector for each partition class.
6. Define the final set of categories for these classification vectors by correlation with the document collection and cutoff. Assign documents below all cutoffs on the basis of maximum association.

Pass 3

Summary of the Steps of the Classification Algorithm

Table 4.3



Flowchart of the Classification Algorithm

Figure 4.9

6. Experimental Results

The classification algorithm described above establishes both a set of document categories and a representation for each such category in the form of its classification vector. Since these vectors are suitable for direct comparison with search requests (i.e. they are identical in form to document index images^x), the implementation of a two level search in the retrieval system is quite straightforward. Thus a user's search request is first matched with the set of classification vectors to determine which categories are most likely to contain documents which will be sufficiently close to the query to be retrieved. Depending on the correlation distribution of the query with the classification vectors, the documents in one or more categories can be retrieved and individually correlated with the search request to produce the final retrieval output.

Assume that the retrieved output for a query \bar{q} produced in the full search mode is the document subset R where:

$$R = \{ \bar{a}_i \} : \delta(\bar{q}, \bar{a}_i) \leq \delta_r.$$

Let a search over the classification categories produce a set of query-classification vector distances:

$$\delta(\bar{q}, \bar{c}_j) = \delta_j, \quad j = 1, n_c;$$

^x For processing purposes, an integer version of the normalized classification vector $\bar{c}/|\bar{c}|$ of equation (4.2) is produced by scaling and truncation.

where n_c is the number of categories. Assume for simplicity that each category subset C_j contains only documents for which:

$$\delta(\bar{c}_j, \bar{d}_i) \leq \delta_0$$

where \bar{c}_j is the representation for C_j . The distance from the query \bar{q} to any member of the set C_j may be bounded by:

$$\max [0, (\delta_j - \delta_0)] \leq \delta(\bar{q}, \bar{d}_i) \leq \delta_j + \delta_0.$$

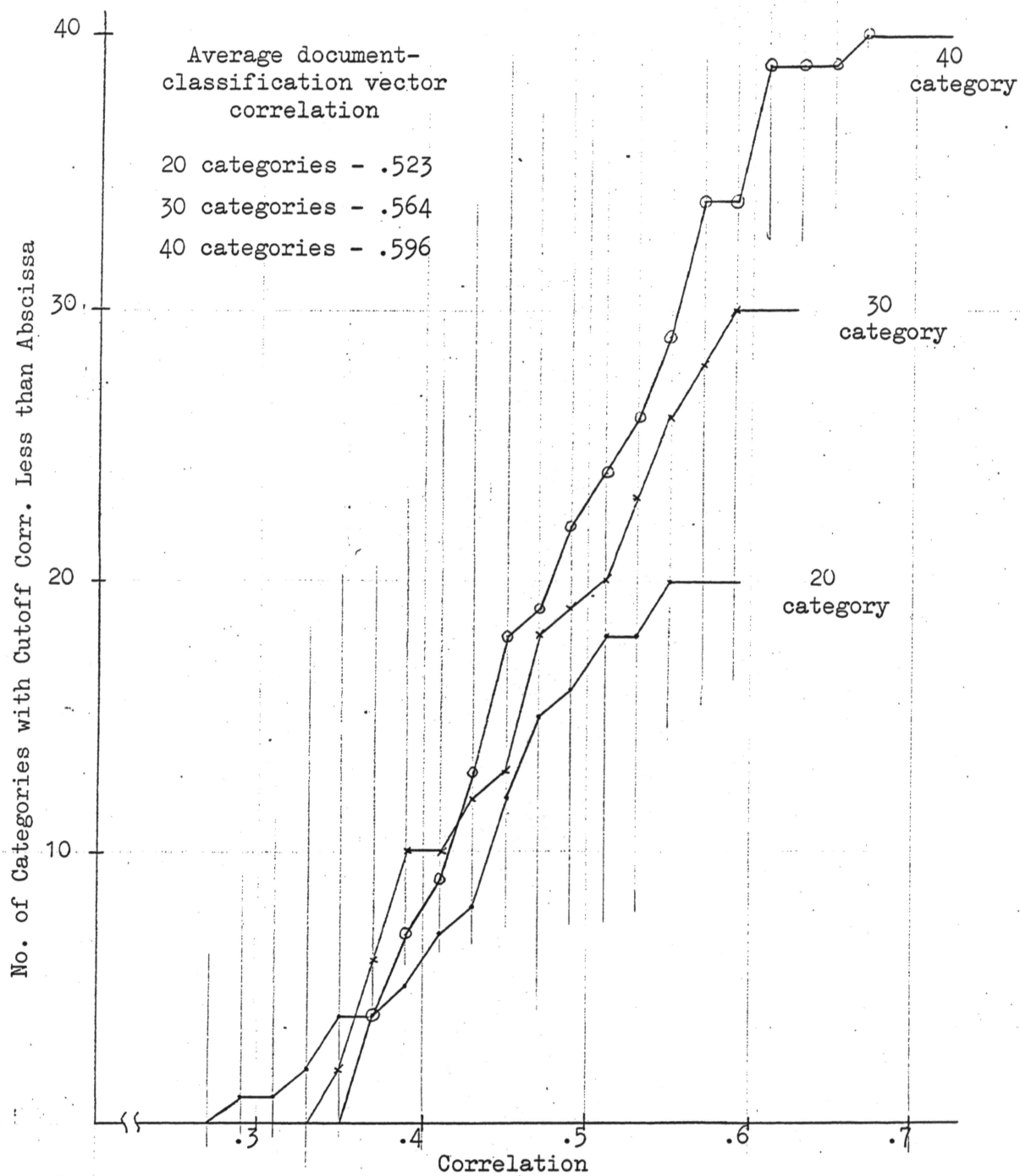
On a probabilistic basis, then, the category for which δ_j is minimum is clearly most likely to contain documents close enough to the query to satisfy the retrieval criterion. Thus the ordering of categories by increasing query-classification vector distance dictates the sequence in which individual query-document comparisons should be made.

To test the characteristics of this system of query-document searching, the classification algorithm was programmed in Fortran and run on the IBM 7094 to produce several classifications of the document set of 405 IRE abstracts discussed earlier. Retrieval results based on a full search of this collection for a set of 24 sample search requests were available from previous experiments conducted with the SMART system. The objective, then, is to compare the retrieval characteristics resulting from the classification induced search system to those obtained in the full search mode. Equation (4.3)

indicates that 20 categories would be about optimal for a collection of 405 documents (assuming only a single category is searched in detail), however classifications of 20, 30, and 40 categories were experimentally produced for comparison purposes. The algorithm required from about 6 to 8 minutes, respectively for these classifications and could undoubtedly be speeded up if it were reprogrammed for this purpose.

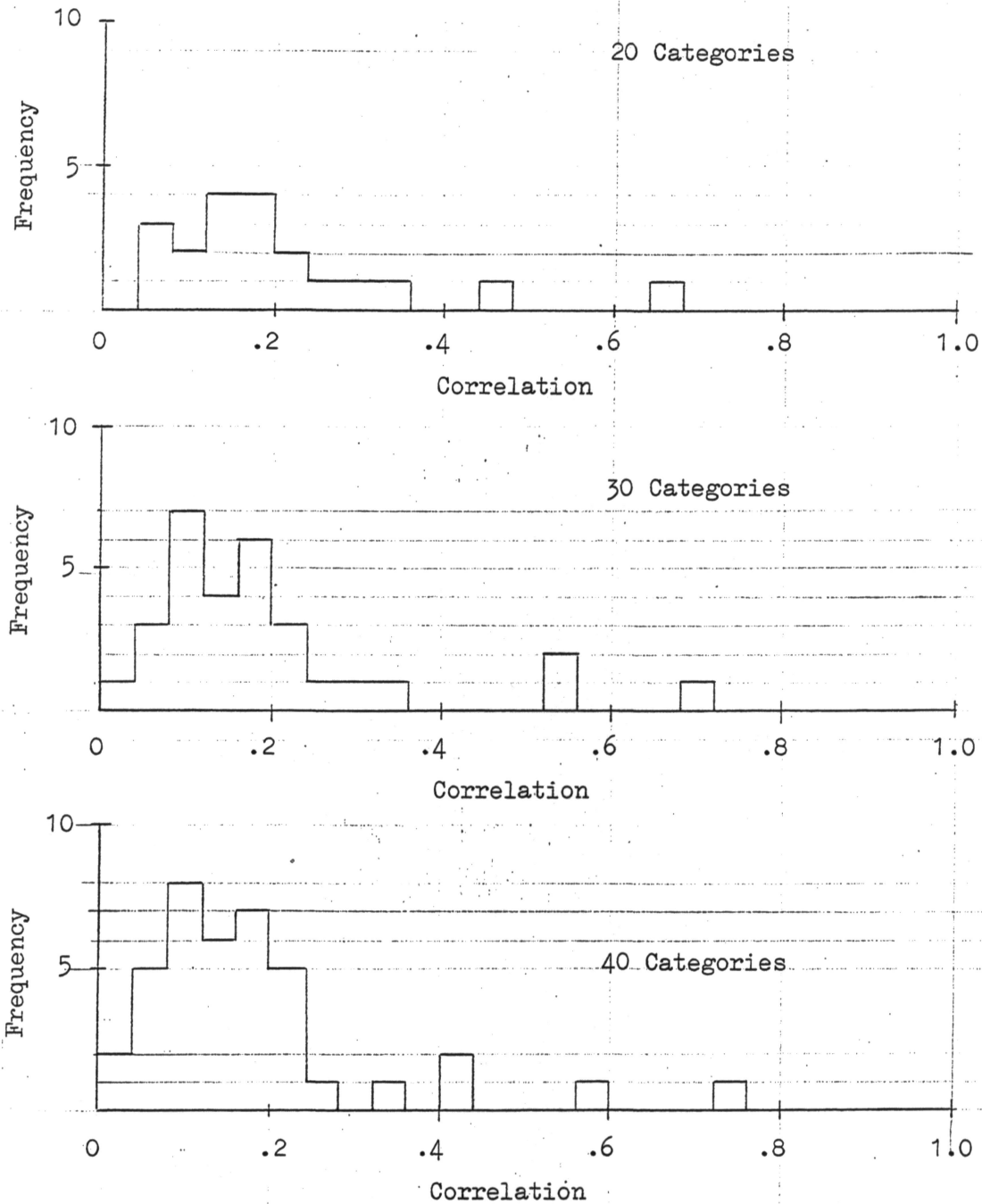
Descriptive parameters of the classifications include the distributions of the cutoff correlations and the average document-classification vector correlations which are shown in Figure 4.10. To evaluate the effectiveness of the search optimization based on the classification induced storage organization, the parameters of interest are: 1.) the consistency of retrieval with respect to all documents, i.e. does the reduced search lead to retrieving the same documents as the full search, and 2.) the consistency of retrieval with respect to relevant documents, i.e. is the retrieval of relevant documents altered by the reduced search? To this end each of the sample search requests was correlated with the set of classification vectors for the three classifications. Figure 4.11 shows the correlation distributions for one of the test queries with the vectors of each of the classifications.

For each of the classifications (20, 30, and 40 categories) the five highest correlating categories for each query were recorded. The documents contained in the union of the first through fifth of such categories were then compared with the first 15 and first 30 documents retrieved by a full search. In addition the number of relevant documents in each of these category retrieved subsets was computed. Assuming then that from 1 through 5 categories would be searched in



Cumulative Frequency Distribution of the Cutoff
Correlation for Three Classifications

Figure 4.10



Query-Classification Vector Correlation
Distributions for Request "Core Memory"

Figure 4.11

détail for each query the following parameters could be produced:

- 1.) The total number of documents in the union of the retrieved categories.
- 2.) The overlap correlation of the category retrieved subset with the first 15 and first 30 documents retrieved by a full search. (The overlap correlation between sets A and B is defined by $n(A \cap B) / \text{minimum}(n(A), n(B))$.)
- 3.) The category recall or percentage of relevant documents in the category retrieved subset to the total number of relevant documents.
- 4.) The normal recall or percentage of relevant documents retrieved to the total number of relevant documents, assuming the same total number of documents retrieved as contained in the category retrieved subset.

It should be noted that this method of evaluating the classification based search is somewhat unfair on two counts. First, it does not consider the correlation distribution of the search requests with the category vectors. Thus when a query has high correlation with only one or two category vectors, only these should be searched. Some queries, however, will not correlate very well with any of the category vectors; and in this case, one should expect to have to search a larger number of categories in detail to do as well as a full search. Queries of this latter type in effect do not fit the classification structure. Second, the degree of association between each classification vector and the documents it represents (as reflected by Figure 4.10 is sufficiently small such that a wide

range of query-document correlations is possible even for the documents in the category with the highest query-category vector correlation observed. Thus in the comparison of recall values it would be fairer to eliminate the low correlating documents from the category retrieved subset before comparison with the full search results. Once these comments are noted, however, it is felt that the evaluation parameters described above are useful in judging the performance of the two level search scheme.

A program was written to produce the evaluation parameters, and the results for a sample search request "Core Memory" are shown in Figure 4-12. From part (a) of this figure, one can see that all the relevant documents can be retrieved by searching only the first two categories; thus 100% recall results with a total of 69 comparisons: 20 for category matching and 49 for document matching. Figure 4.13 shows the evaluation parameters averaged over the set of 24 search requests for each of the classifications. Even though the results are not as good as for the single query shown, it is nevertheless clear that for a relatively small cost (in terms of missing associated associated documents) a large increase in search efficiency can be gained.

On the basis of the experimental evidence gained with this small collection it can be concluded that:

- 1.) A metric query-document matching function enables an automatic classification of the type considered to be easily produced.
- 2.) Such classification schemes are likely to be more

	Category Number	Corr. with Query	No. of Elements	Cumulative No. of Elements	Overlap Corr. 1st 15	Overlap Corr. 1st 30	Category Recall	Normal Recall
20 ✓ Categories	5	.66	29	29	.60	.62	.86	.86
	2	.46	23	49 ✓	.80	.77	1.00	1.00
	7	.34	35	84	.87	.83	1.00	1.00
	4	.32	22	105	1.00	.97	1.00	1.00
	17	.28	27	124	1.00	1.00	1.00	1.00
30 Categories	21	.70	21	21	.67	.76	.71	.71
	2	.56	15	30	.87	.70	.71	.86
	5	.53	20	48	1.00	.90	1.00	1.00
	11	.32	21	69	1.00	.93	1.00	1.00
	22	.28	25	90	1.00	.97	1.00	1.00
40 Categories	8	.73	14	14	.71	.86	.43	.71
	5	.58	16	30	.87	.70	.71	.86
	34	.43	9	39	.93	.80	.86	1.00
	2	.43	11	47	.93	.87	.86	1.00
	12	.35	18	64	1.00	.90	.86	1.00

(Category numbers have no relation from one classification to another.)

Evaluation of Two Level Searching for Test Query "Core Memory"

Figure 4.12

	Category Rank	No. of Elements	Cumulative No. of Elements	Overlap Correlation 1st 15	Overlap Correlation 1st 30	Category Recall	Normal Recall
20 Categories	1	27	27	.54	.52	.55	.85
	2	24	49	.71	.61	.71	.92
	3	26	72	.79	.71	.78	.95
	4	25	95	.84	.79	.88	.97
	5	25	116	.88	.83	.91	.98
30 Categories	1	18	18	.54	.63	.51	.79
	2	18	34	.71	.57	.69	.87
	3	18	51	.78	.67	.82	.92
	4	18	68	.84	.72	.83	.95
	5	17	82	.86	.77	.86	.97
40 Categories	1	12	12	.50	.71	.42	.72
	2	13	25	.60	.56	.63	.82
	3	12	37	.73	.60	.72	.89
	4	13	49	.78	.68	.78	.92
	5	12	61	.84	.73	.82	.94

Average Evaluation Results for 24 Search Requests

Figure 4.13

attractive as the collection size increases on two counts: first, because a larger collection should lead to better defined categories, and second, because the ratio of comparisons required with classification to the number required with a full search ($2N^{\frac{1}{2}}/N$) is a decreasing function of N .

REFERENCES

1. Rial, J.F., "A Pseudo-Metric for Document Retrieval Systems," Report W-4595, The Mitre Corporation, Bedford, Mass.
2. Mooers, C.N., "A Mathematical Theory of Language Symbols In Retrieval," Proceedings of the International Conference on Scientific Information, Washington, D.C., 1958
3. Salton, G., "Manipulation of Trees in Information Retrieval," Communications of the ACM, Vol. 5, No. 2, February 1962
4. Sussenguth, E.H., "Structure Matching in Information Processing," Ph.D. Thesis, Harvard University, April 1964
5. Borko, H. and Bernick, M., "Automatic Document Classification," Journal of the ACM, Vol. 10, No. 2, April 1963
6. Baker, F., "Information Retrieval Based Upon Latent Class Analysis," Journal of the ACM, Vol. 9, No. 4, Oct. 1962
7. Goffman, W., "A Searching Procedure for Information Retrieval," Information Storage and Retrieval, Vol. 2, No. 2, July 1964