Inform. Stor. Retr. Vol. 7, pp. 217–240. Pergamon Press 1971. Printed in Great Britain

THE USE OF HIERARCHIC CLUSTERING IN INFORMATION RETRIEVAL

N. JARDINE and C. J. VAN RIJSBERGEN

King's College Research Centre, Cambridge

Summary—We introduce information retrieval strategies which are based on automatic hierarchic clustering of documents. We discuss the evaluation of retrieval strategies and show, using a subset of the Cranfield Aeronautics document collection, that cluster-based retrieval strategies can be devised which are as effective as linear associative retrieval strategies and much more efficient. Finally, we outline how cluster-based retrieval may be extended to large growing document collections and indicate some ways in which the effectiveness of cluster-based retrieval strategies may be improved.

WE CONSIDER very simple conditions for experimental document retrieval. Each of a collection of documents is described by presence or absence of each of a set of index terms. Each of a set of test requests consists of a subset of the index terms which describe the documents. Both documents and requests can therefore be specified by indexed binary strings. A 1 in the ith position of the string indicates presence of the ith index term in the document or request, and a 0 indicates its absence. The binary strings we shall call the *representatives* of documents and requests; although where the sense is obvious we shall for brevity often write of comparison of documents with documents, and the comparison of requests with documents, where strictly we ought to write of their representatives. Similarly, where the sense is obvious, we shall write of retrieval of documents by automatic retrieval strategies when in fact document references are retrieved.

It has been determined in advance whether each document is relevant or non-relevant to each request. In response to each request a *retrieval strategy* produces a set of documents. The proportion of the relevant documents in the collection which are retrieved is called *recall.* The proportion of the retrieved documents which are relevant is called *precision.* In practice it turns out that attempts to achieve high recall tend to produce low precision and vice versa. Retrieval strategies are therefore sought which trade precision against recall in some optimal manner. The performance of a retrieval strategy measured in terms of precision and recall we call its *effectiveness.* Effectiveness is not, however, the only criterion which guides us in our choice of a retrieval strategy. Some strategies are more costly than others in terms of computer store and time. This aspect of retrieval strategies we call *efficiency.* In the first section we describe how effectiveness and efficiency of certain retrieval strategies may be compared.

Often it may be sensible to consider more complicated experimental systems than that outlined above. For example, relative frequencies of incidence of index terms may be taken into account in representing documents; differential weighting of terms may be allowed in requests; retrieval for a single request may be carried out more than once, the retrieval strategy being modified in the light of the user's response to the first set of documents obtained; requests may be processed in batches; relevance may be measured on an ordinal or numerical scale; and so on. We have focused attention on a very simple system for two reasons. Firstly, the data-base on which we have carried out our experiments is in this simple form. Secondly, the retrieval strategies which we describe are most readily evaluated and compared with other strategies using a simple experimental system.

The majority of the automatic document retrieval experiments which have been described in the literature depend on comparison of requests with individual documents. The experimental strategy generally called *linear associative retrieval* proceeds as follows. A measure of association between request representatives and document representatives is defined. Associations between each request and each document are calculated. The documents are then ranked in order of decreasing association with each request. Rank positions $i > j > \ldots > n$ are selected, and for each request precision and recall values are calculated for the i most highly associated documents; the j most highly associated documents; and so on. The precision and recall values at each rank position averaged over all requests may be plotted as a precision/recall graph (as in Fig. 4). Some workers have chosen instead of rank positions particular thresholds of the association measure.

As it stands, linear associative retrieval is not a complete retrieval strategy. In order to obtain a retrieval strategy from it we must decide for each request the rank position or association measure threshold at which to select a set of documents. This is the *cut-off* problem. Choice of a cut-off position or threshold may be based on learning from past effectiveness at various rank positions or thresholds. It may involve evaluation by a user of an initial set of documents, or of a sequence of sets of documents, retrieved in response to his request (interactive retrieval). Surprisingly little attention has been paid to this aspect of linear associative retrieval strategies.

Calculation of association values between each request and each document is laborious, and so is the subsequent ranking of documents in order of decreasing association. If the document collection is of size n, the procedure has a dependence of order $(n+n \log n)$. Clustering of documents has been used by several workers to increase the efficiency of linear associative retrieval without serious loss of effectiveness. Association values between pairs of document representatives are calculated. On the basis of these values documents are assigned to clusters. Each cluster is then represented in some way, for example by a typical document or by a centroid. A request is first matched with the representatives of each cluster and the best matching cluster is selected. Linear associative searching is applied to the documents in the best matching cluster and may be extended to the next best matching cluster, and so on. In section (iii) we review briefly some of the cluster methods which have been used for this purpose.

The use of automatic clustering of documents which we shall investigate involves a complete break with linear associative retrieval. A hierarchic system of document clusters is constructed and retrieval strategies are devised which match requests with representatives of clusters at successively lower levels in the hierarchy. The cluster with which the best match is achieved is the set of documents retrieved by the strategy.

It is obvious that such strategies will be much faster than any strategy based on linear associative searching. Consider a collection of n documents. Order n^2 work is used to construct a hierarchic clustering, and there are ways of reducing this for large collections. We show that in return for this initial investment the retrieval of documents in response to each request is reduced from the order $(n+n \log n)$ dependence of a linear associative search to order $(\log n)$.

However, it is at first sight surprising that cluster-based retrieval can equal, let alone improve on, the effectiveness of linear associative retrieval. Before describing in detail the setting up and operation of cluster-based retrieval strategies and investigating their effectiveness experimentally, we indicate informally (and without rigour) reasons for supposing that document clustering can in principle improve retrieval effectiveness. It is intuitively plausible that the associations between documents convey information about the relevance of documents to requests. This hypothesis, which we call the *cluster hypothesis*, can easily be checked on a particular document collection. For each request we calculate the associations between all pairs of documents both of which are relevant, and the associations between all pairs of documents one of which is relevant and the other non-relevant. The distributions of the two sets of association values may then be compared as in Fig. 1.

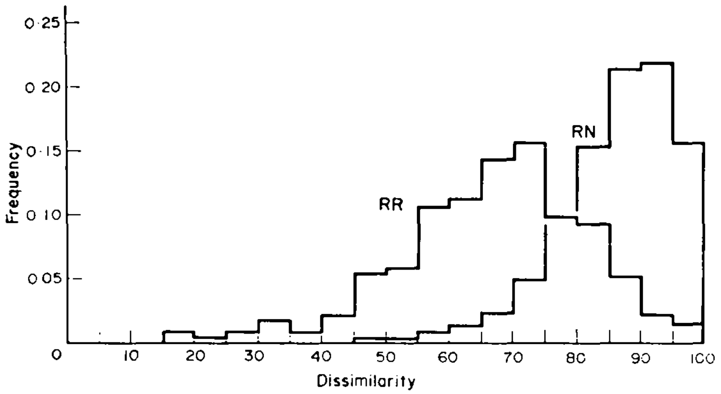

FIG. 1. Comparison of the distribution of association values between pairs of documents relevant to the same request (RR), and the distribution of association values between pairs of documents one of which is relevant and the other non-relevant to a request (RN). The association coefficient used is described in section (ii).

Each of the clusters of documents used in cluster-based retrieval consists of a set of documents which are relatively isolated from documents not in the cluster. The clusters are arranged in a hierarchic system in which clusters of highly associated documents are nested within clusters of less highly associated documents (see Fig. 7). The retrieval strategies first match a request with clusters at the top of the hierarchy and choose the best matching cluster C. They then match the request with representatives of the clusters immediately included in C, and so on, working down the hierarchy until an optimal match is achieved. The effectiveness of this procedure depends on the fact that closely associated documents tend both to belong to the same clusters and (as suggested by Fig. 1) to be relevant to the same requests.

We now outline the layout of the paper. We first discuss the evaluation and comparison of effectiveness and efficiency of document retrieval strategies. Sections (ii–iv) which are fairly technical describe the measurement of association between documents, methods for clustering document collections, the algorithm devised by C. J. van Rijsbergen for implementing hierarchic document clustering, and ways of representing document clusters and setting up search strategies. In section (v) we report the results of some cluster-based strategies on the Cranfield data-base, and in section (vi) we suggest ways in which the effectiveness of our cluster-based retrieval strategies could be improved, and ways in which cluster-based retrieval can be extended to large growing document collections.

The data-base which we used is derived from the Cranfield Aeronautics document collection. It was made available to us by Dr K. Spärck Jones, and full details of it have been published in Spärck Jones and Jackson [1]. Full details of the Cranfield Aeronautics document collection from which our data-base is derived are given in Cleverdon, Mills and Keen [2]. Its vital statistics are as follows:

Number of documents:	200
Number of requests:	42
Number of terms:	712
Average number of index terms per document:	31
Average number of index terms per request:	6
Average number of relevant documents per request:	4

The technical symbols used in the following sections are:

$\| \|$	for the number of elements in a set,
$\cap$	for the intersection of sets,
$\cup$	for the union of sets,
Δ	for the symmetric difference of sets.

(i) *Evaluation and comparison of information retrieval strategies*

(a) *Effectiveness.* We denote the set of documents relevant to a request by A and the set of documents retrieved by a retrieval strategy by B. *Precision* is the proportion of the retrieved documents which are relevant, that is

$$\frac{|A \cap B|}{|B|}.$$

Recall is the proportion of relevant documents which are retrieved, that is

$$\frac{|A \cap B|}{|A|}.$$

When linear associative retrieval is used experimentally it is usual, after ranking the documents in order of decreasing association with each request, to calculate precision and recall values achieved for each request at selected rank positions. The values of precision and recall at each rank position are then averaged over requests and displayed by a precision/recall graph (see Fig. 4). Other effectiveness measures for experimental systems based on precision and recall have been discussed by Keen [3]. Effectiveness measures of other kinds have been described by Swets [4] and Robertson [5]. Some of the problems which arise in evaluating effectiveness of operational retrieval systems have been discussed by Lancaster [6] and Cleverdon [7].

Cluster-based retrieval strategies have built into them a process analogous to choice of a cut-off position in linear associative retrieval, so that only a single set of documents is produced in response to each request. Precision/recall graphs cannot, therefore, be used to evaluate their effectiveness. It is necessary to find a more general evaluation method which can be used to measure and compare the effectiveness of retrieval strategies.

We seek a measure of effectiveness F which takes real values on sets of retrieved documents. For convenience we bound F between 0 and 1. Denoting precision P and recall R, the following conditions on F are self-explanatory:

(i) F is a function of P and R
(ii) $F=0$ if $R=0$ (note that if $R=0$ then $P=0$)
(iii) $F=1$ if $R=1$ and $P=1$
(iv) if $P'>P''$ and $R'=R''$ then $F'>F''$
(v) if $R'>R''$ and $P'=P''$ then $F'>F''$.

If F is to be used to measure the effectiveness of retrieval strategies for a user we require it also to be a function of the relative importance which is attached by the user to recall and precision. There are many ways in which we might define "relative importance" in this context. One way of defining it is as follows. We stipulate that a user who attaches *equal* importance to recall and precision is one who when $P=R$ is prepared to trade a given increment in recall for an equal loss of precision. In this case F must satisfy the condition

$$\text{if } R/P = 1 \text{ then } (\partial F/\partial P)_R = (\partial F/\partial R)_P.$$

We define the relative importance β attached to recall and precision by a user as the recall/precision ratio at which he is prepared to trade a given increment in recall for an equal loss of precision. In general therefore, we require the function F to satisfy the condition:

$$\text{(vi) if } R/P = \beta \text{ then } (\partial F/\partial P)_R = (\partial F/\partial R)_P.$$

A function F which satisfies conditions (i)–(vi) is

$$\frac{(\beta^2+1)PR}{\beta^2 P+R}.$$

In practice we have used $E=1-F$ as a measure of effectiveness. In case $\beta=1$, E reduces to

$$\frac{|A \Delta B|}{|A|+|B|}$$

which is a normalized symmetric difference of A (the set of relevant documents) and B (the set of retrieved documents). The behaviour of the function E for selected values of β as recall and precision are varied is shown in Fig. 2.

Effectiveness of a retrieval strategy is investigated by calculating values of E (for selected values of β) achieved on the sets of documents retrieved in response to test requests. In practice we usually first calculate E for $\beta=1{\cdot}0$. Corresponding values of E for $\beta=0{\cdot}5$ and $\beta=2{\cdot}0$ can be seen on Fig. 2.

Values of E achieved in response to a test set of requests can be used in various ways to compare effectiveness of retrieval strategies. In this paper we have plotted for each retrieval strategy the cumulative frequency distribution of values of E (see Fig. 3). Alternative presentations include frequency distributions in histogram form and scatter diagrams. It is difficult to find an adequate single measure of overall effectiveness for comparison of search strategies on a test set of requests. The standardized difference between the mean values of E achieved on a test set of requests is not in general a satisfactory measure of difference in overall effectiveness, because there is no reason to suppose that values of E will have distributions of any simple form. The variation distance

$$\int |p_1(x)-p_2(x)|\,dx$$

where p_1, p_2 are distributions of values of E, may be a more satisfactory measure.

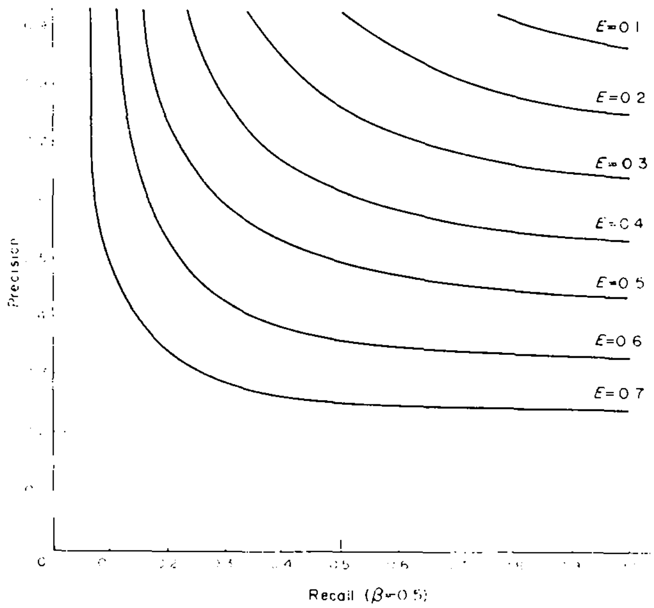
E=0·1
E=0·2
E=0·3
E=0·4
E=0·5
E=0·6
E=0·7
Precision
0
0·4
0·5
0·6
Recall (β=0·5)

FIG. 2 (a).

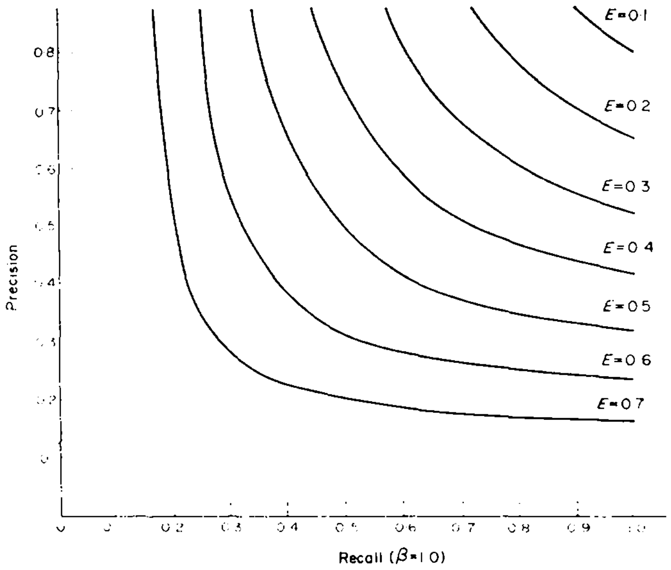
E=0·1
E=0·2
E=0·3
E=0·4
E=0·5
E=0·6
E=0·7
Precision
0·8
0·7
0·6
0·5
0·4
0·3
0·2
0
0·2
0·3
0·4
0·5
0·6
0·7
0·8
0·9
Recall (β=1·0)

FIG. 2 (b).

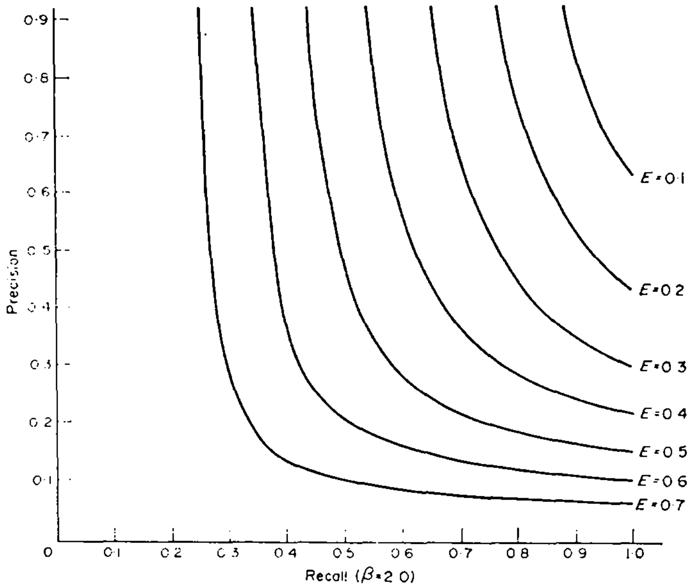

FIG. 2 (c).

FIG. 2. Behaviour of the effectiveness measure E for selected values of β as recall and precision are varied. (a), $\beta=0{\cdot}5$. (b), $\beta=1{\cdot}0$. (c), $\beta=2{\cdot}0$. See text for fuller explanation.

It is likewise difficult to find appropriate tests for significance of difference in overall effectiveness of retrieval strategies. Parametric tests are inappropriate because of our ignorance about the form of distributions of values of E. Non-parametric tests such as the sign test and the Kolmogorov–Smirnov test may be more appropriate (see SIEGEL [8]). However, it is reasonable to doubt whether any statistical significance test is appropriate. The test requests are unlikely to be independent and it is probably unreasonable to regard them as random samples from a population of requests. The same difficulties arise in comparing precision/recall graphs (see SALTON [9] p. 311).

In order to obtain *benchmarks* against which to compare experimentally the effectiveness of cluster-based retrieval strategies we carried out the following evaluations using the Cranfield data-base:

*MK*1. *Measurement of the maximum effectiveness which is theoretically attainable by a cluster-based retrieval strategy which selects a single cluster in response to each request.* For each request the cluster of documents is selected which yields the least value of E (for chosen values of β). The method used to generate document clusters is described in section (iii).

*MK*2. *Measurement of the maximum effectiveness which is theoretically attainable using linear associative retrieval at a single rank position.* The single rank position is selected at which the sets of documents retrieved yield the least average value of E (for chosen values of β).

*MK*3. *Measurement of the maximum effectiveness which is theoretically attainable using a linear associative retrieval strategy.* For each request the rank position at which the set of documents retrieved gives the least value of E (for chosen values of β) is selected.

The cumulative frequency distributions of values of E achieved by the "ideal" strategies $MK1$, $MK2$ and $MK3$ on the Cranfield data-base are plotted for $\beta = 0{\cdot}5$, $\beta = 1{\cdot}0$ and $\beta = 2{\cdot}0$ in Fig. 3.

It is evident that when equal importance is attached to precision and recall the effectiveness of $MK1$, the ideal cluster-based retrieval strategy, is substantially greater than the effectiveness of the ideal linear associative retrieval strategies, $MK2$ and $MK3$. When more importance is attached to precision than to recall, the effectiveness of the ideal cluster-based strategy relative to that of the ideal linear associative retrieval strategies is increased. However, as progressively more importance is attached to recall than to precision, the relative effectiveness of the ideal linear associative retrieval strategies approaches that of the ideal cluster-based strategy.

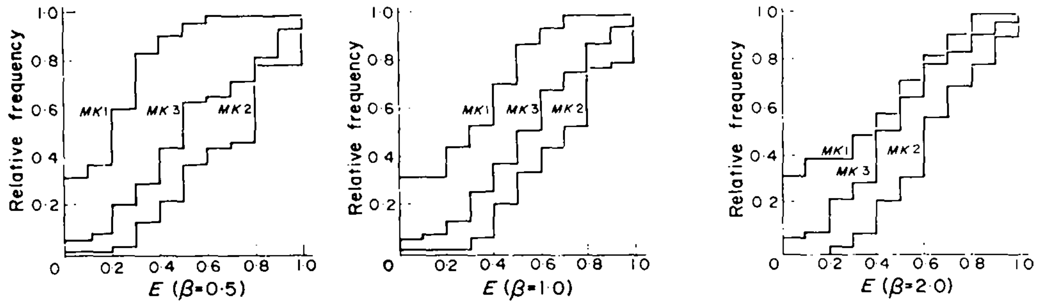

Fig. 3. Comparison of the maximum effectiveness theoretically attainable by two kinds of linear associative retrieval strategy and the maximum effectiveness theoretically attainable by a cluster-based strategy. See text for fuller explanation.

It should be noted that the cumulative frequencies of values of E plotted in Fig. 3 are benchmarks. They set upper bounds to the effectiveness of certain kinds of retrieval strategies. In order to achieve the effectiveness of $MK3$ in practice, a linear associative strategy would have to incorporate an infallible method for learning the optimal threshold at which to retrieve documents for each request. It is very difficult to envisage such a learning procedure. In order to achieve the effectiveness of $MK2$ in practice, a linear associative strategy would have to incorporate an infallible method for learning the threshold which is on average optimal for requests. $MK2$ is a more realistic estimate than $MK3$ of the effectiveness which may be achievable in practice by a linear associative strategy, because it is possible to envisage appropriate learning techniques. In order to achieve the effectiveness of $MK1$, a cluster-based retrieval strategy would have to search the hierarchy of clusters in a way which infallibly finds the optimal cluster.

(b) *Efficiency*. The primary motive in considering cluster-based retrieval strategies is to improve *efficiency* in document retrieval so as to make effective retrieval a feasible proposition for large collections. Cluster-based strategies need substantial initial investment of computer time to construct the system of document clusters. This initial investment pays dividends in reduction in the effort involved at search time in retrieving documents in response to each request or batch of requests.

In Table 1 we contrast the efficiency of linear associative retrieval based on a document file with the efficiency of hierarchic cluster-based retrieval. We have given only rough

TABLE 1. COMPARISON OF EFFICIENCY OF LINEAR ASSOCIATIVE RETRIEVAL AND HIERARCHIC CLUSTER-BASED RETRIEVAL

	Linear associative retrieval	Hierarchic cluster-based retrieval
Computer time for generation of data-structure		order (n^2) [this can be greatly reduced for large collections; see text]
Computer store for data-structure	order (n)	order (n)
Search time for each request	order ($n+n \log n$)	order ($\log n$)

estimates of efficiency, because there are many factors affecting the efficiency of an operational retrieval strategy which we cannot easily anticipate from consideration of experimental strategies.

Some of the estimates given in Table 1 require comment. The effort needed to construct a hierarchic clustering is order n^2. However, in the final section we show how it is possible to reduce this greatly for large document collections by constructing a hierarchic clustering of a small subset of a collection and allocating the remaining documents to this by a fast allocation procedure. If the total collection is of size n and the clustered subset is of size m, the dependence of this method is order $(m^2+(n-m) \log m)$. The storage requirements are self-explanatory (for hierarchic clustering note that the number of distinct clusters in a hierarchy on n objects is order n).

Search of a document file requires n operations to compare a request with each document and $n \log n$ operations to rank documents in order of decreasing association with a request. Under certain circumstances the efficiency of linear associative retrieval can be increased by combining a document file with an inverted file (in which documents in which each index term occurs are listed). A single scan of an inverted file can eliminate all documents which share no index terms with a request, and a linear associative search can be applied to the remaining documents using a document file. But, of course, the combined files require much more store. The retrieval time for a hierarchic document clustering using the simple downward search strategies described in section (iv) is of order $\log n$ (see WINDLEY [10] for details of search efficiency for hierarchic structures).

(ii) *Measures of association*

Both document and request representatives are binary strings in which a 1 in the ith position indicates occurrence of the ith index term, and a 0 indicates its absence.

Numerous coefficients of association between binary strings have been described. See, for example, GOODMAN and KRUSKAL [11, 12], KUHNS [13], SOKAL and SNEATH [14] and CORMACK [15]. We give only a brief guide to the selection of an appropriate coefficient.

Let X and Y be the sets of index terms occurring in two document (or request) representatives. The simplest of all association coefficients is

$$\text{(i)} \quad |X \cap Y|$$

which is the number of shared index terms. This coefficient does not take all the information contained in X and Y into account. The following coefficients which have been used in document retrieval take account of all the information contained in X and Y:

$$\text{(ii)} \quad \frac{2|X \cap Y|}{|X|+|Y|} \quad \text{Dice's coefficient}$$

$$\text{(iii)} \quad \frac{|X \cap Y|}{|X \cup Y|} \quad \text{Jaccard's coefficient}$$

$$\text{(iv)} \quad \frac{|X \cap Y|}{|X| \times |Y|} \quad \text{Cosine coefficient, Salton [9]}$$

$$\text{(v)} \quad \frac{|X \cap Y|}{\min(|X|, |Y|)}$$

Overlap coefficient, Salton [9].

These may all be considered as normalized versions of (i). In Fig. 4 we give precision/recall graphs for linear associative retrieval on the Cranfield data-base using (i) and (ii) to show the effect of normalization.

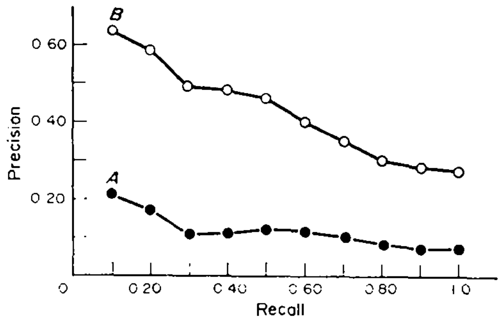

Fig. 4. Precision/recall graphs showing the effect of normalization of the association coefficient on the effectiveness of linear associative retrieval. A, not normalized. B, normalized.

Failure to normalize the association coefficient has an equally drastic effect on cluster-based retrieval. If (i) is used as an association coefficient, documents described by relatively large numbers of index terms tend to form a single cluster.

Some workers have based their choice of an association measure on a model for requests and document representatives. Thus Salton [9] considered requests and document representatives as vectors imbedded in a k-dimensional Euclidean space, where k is the total number of index terms. (iv) above, then has an interpretation as the cosine of the angular separation of two vectors (see also Switzer [16]). He showed that in linear associative searching, (iv) is more effective than (v). This is probably because $\min(|X|, |Y|)$ is an inappropriate normalization. Other workers have sought to measure association in terms of deviation from independence in the occurrence of index terms in two representative strings [17, 13, 18]. Most of the measures proposed have taken as a null hypothesis independence and equal probability of occurrence of the terms in each string. This is obviously unrealistic. Unfortunately if the unequal probabilities of occurrence of terms and the statistical dependences between terms were taken into account any such measure would be computationally formidable.

The cluster method which we shall apply to measures of association between document representatives depends only on the rank-ordering of association values. Measures (ii) and (iii) are monotone, so that the clusters obtained are unaffected by choice between them. Because of its (slightly) greater computational convenience we have used Dice's coefficient subtracted from 1.

$$1-\frac{2|X \cap Y|}{|X|+|Y|}=\frac{|X \Delta Y|}{|X|+|Y|}$$

which is a normalized symmetric difference of X and Y.

(iii) *Document clustering*

Just as many measures of association have been proposed, so numerous methods of automatic classification have been described. For surveys see SOKAL and SNEATH [14], BALL [19], HARRISON [20], JARDINE and SIBSON [21], SPÄRCK JONES [22] and CORMACK [15].

In Fig. 5 we show some of the kinds of classification which may be obtained from an association or dissimilarity coefficient on a set of documents.

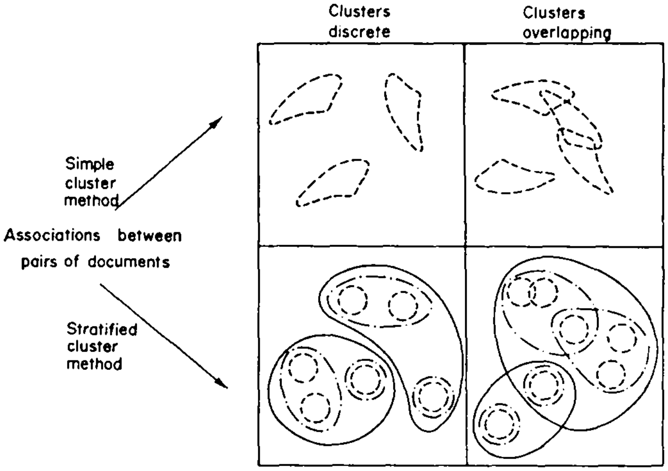
Clusters discrete
Clusters overlapping
Simple cluster method
Associations between pairs of documents
Stratified cluster method

FIG. 5. Kinds of classification derivable from an association coefficient on a set of documents.

FAIRTHORNE [23] and GOOD [24] were amongst the first to suggest that automatic classification might prove useful in document retrieval. Several workers have used simple cluster methods in order to reduce the extent of linear associative searches [9 Ch. 7] [25]. Simple cluster methods have also been used for clustering of index terms in automatic thesaurus construction [26, 1, 27]. Two approaches to simple clustering have been suggested. One approach starts by generating an initial set of clusters and then seeks to improve the clustering by reallocation of objects. The methods of ROCCHIO [28] and NEEDHAM [29] are of this kind. Both methods lead to clusters which satisfy a homogeneity criterion, and Rocchio's method allows initial specification of the number of clusters and bounds on their sizes. DATTOLA'S [30] method starts with an initial partition of the documents and reallocates documents using comparison of document representatives with the frequency distribution of index terms in each initial cluster (the cluster profiles). The algorithm is of interest because of its efficiency. It has an order $n \log n$ dependence whereas cluster algorithms which operate on an association measure have dependence of at best order n^2. The results obtained by these methods are not independent of starting point. The other approach uses algorithms which find all clusters which satisfy a cluster definition. The graph-theoretic methods of GOTLIEB and KUMAR [31], AUGUSTSON and MINKER [32, 33] and VASWANI [34] are of this kind. The results obtained by these methods are independent of starting point. BONNER'S [35] method forms initial clusters by the second approach and then allows reallocation of documents so as to adjust the sizes of clusters. BORKO and BERNICK [36] applied factor analysis to index terms and used the factor-loadings of the documents as a basis for construction of simple clusterings.

The cluster methods which are appropriate for retrieval strategies based entirely on matching of requests with clusters are very different from those which are appropriate for term clustering or for document clustering to limit the extent of search in linear associative retrieval. *Stratified* systems of clusters are appropriate because the level of a cluster can be used as a parameter in retrieval strategies analogous to rank position or association measure threshold in linear associative retrieval. Retrieval of a cluster at a low level in the hierarchy which matches a request well tends to produce high precision but low recall; just as cut-off at a low rank position in linear associative retrieval tends to yield high precision but low recall. Similarly, retrieval of a cluster at a high level in the hierarchy which matches a request well tends to produce high recall but low precision and this is analogous to cut-off at a high rank position in linear associative retrieval. *Hierarchic* systems of clusters are appropriate for two reasons. First, very efficient strategies can be devised to search a hierarchic clustering. Secondly, construction of hierarchic systems is much more efficient than construction of non-hierarchic systems of clusters (see JARDINE and SIBSON [21], appendices 3, 4 and 5).

DOYLE [37] and LITOFSKY [38, 39] have described and applied methods for hierarchic document classification. Doyle used a method due to WARD [40] which, like the single-link method described below, operates on an association coefficient on documents, and is independent of starting point. He considered document classification as a technique for organizing and indexing a document collection, rather than specifically as a basis for automatic document retrieval. Litofsky's method operates directly on document representatives without calculation of an association coefficient. The algorithm used by Litofsky seeks to minimize the average number of distinct index terms which occur in the documents in a cluster at the lowest level of the hierarchy. Each cluster in the hierarchy is then represented by the index terms which are common to all its members and which are not common to all members of the cluster immediately containing it. Litofsky applied the method to a collection of 45,000 documents, and showed that the representation obtained is useful for browsing, and could be used to limit the sets of documents searched in response to requests. The method appears inappropriate for retrieval based entirely on matching of requests with clusters.

The single-link method of hierarchic clustering was selected for the following reasons:

1. The clustering obtained by single-link depends only on the rank-ordering of values of the association coefficient on the documents.

2. The single-link method is stable in the sense that small errors in values of the association coefficient lead to correspondingly small changes in the system of clusters.

3. A set of documents in which all pairs are linked at some value θ of the association coefficient (a *completely linked set*) is never assigned to more than one cluster at level θ.

4. The single-link method produces a clustering which is unlikely to be altered drastically when further documents are incorporated. A new document is assigned to a cluster at level θ if it has association value θ or more with any document in the cluster. If a document is assigned to two or more clusters at level θ they are amalgamated. If it is assigned to no cluster at level θ it generates a new cluster at level θ.

A full account of the properties of the single-link method and other hierarchic cluster methods is given in JARDINE and SIBSON [21, 41].

The first property is important because absolute numerical values and ratios of values of an association coefficient are rarely significant. There is usually no statistical justification for using the particular values obtained rather than, for example, their logarithms or square-roots. The second property is important because it ensures that the document clustering obtained by single-link cannot be greatly affected by minor errors in compilation of index terms or by minor computational errors in calculation of an association coefficient or in the subsequent clustering. The effects of errors in indexing on values of association coefficients have been discussed in detail by JACKSON [42]. The third property is important because if the cluster hypothesis is correct, completely linked sets of documents are likely to be relevant to the same requests; so that a cluster method which breaks up completely linked sets would not serve as a basis for effective retrieval. The fourth property is important, because it makes it relatively easy to update the system of clusters obtained by the single-link method when representatives of new documents in a growing collection are considered.

The way in which single-link clusters are related to an association coefficient on a set of documents can be pictured as follows. Suppose we select a threshold θ of the association coefficient. We may then draw a graph with vertices representing documents and edges joining just those pairs of documents with association θ or more. The single-link clusters at level θ are the connected components of the graph. (See Fig. 6.)

A hierarchic system of single-link clusters can be displayed geometrically as in Fig. 7. This kind of representation of a hierarchic system of clusters is generally called a *dendrogram*. The level at which a cluster last appears as one moves down the hierarchy is called its

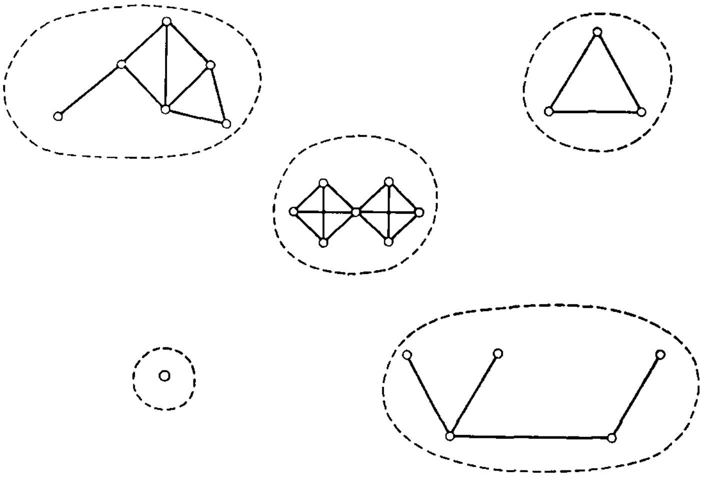

FIG. 6. Single-link clusters at level θ in a hierarchic clustering displayed on a graph which represents an association coefficient thresholded at level θ. See text for fuller explanation.

splitting-level. Alternatively, we may display a single-link clustering as a tree in which each node represents a cluster at its splitting-level. (See Fig. 7.) A tree description is useful both for visualizing the data-structure which is used to store a single-link clustering in the computer, and in formulating search strategies. The clusters which are immediately included in a cluster C constitute the *filial set* of C; conversely, C is the *parent* of each cluster in its filial set.

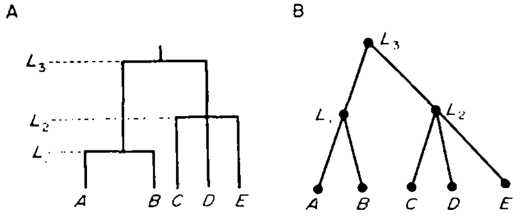

FIG. 7. A, a hierarchic system of clusters displayed as a dendrogram. B, the same system of clusters displayed as a tree in which nodes represent clusters at their splitting-levels.

In the data-structure (described in Appendix 1) two kinds of information about clusters are stored. The positions of clusters in the system of clusters are indicated by their parents and filial sets. This information is needed for the search strategies described in the next section. The information needed for computing values of a matching function between clusters and requests is also stored.

The DYNALINK algorithm which constructs this data-structure was described by VAN RIJSBERGEN [43]. We given here only an informal sketch of its operation. The values of the association coefficient are considered in a single scan in arbitrary order. At each stage of the operation of the algorithm (except the last) the association values already considered constitute a subset of the entire set of association values. At each stage the algorithm constructs a provisional representation of a single-link hierarchy which is consistent with the subset of the association values considered so far. As new association values are considered, the provisional representation is updated until finally the representation of the single-link hierarchy based on the entire set of association values is obtained.

When a large document collection is clustered, substantial computer store can be saved by implementing single-link in this way. Storage of the entire association coefficient can be eliminated by calculating association values sequentially and inputting them directly to the algorithm. Updating of the system of clusters as new documents are considered is readily carried out by the algorithm.

(iv) *Cluster representation and cluster search strategies*

Hierarchic cluster-based retrieval strategies produce in response to each request a single set of documents which constitute a cluster at some level in the hierarchy. In section (i) we defined a function F which evaluates retrieved sets of documents in terms of recall, precision, and a parameter β which measures the relative importance attached to recall and precision. Given a choice of value for β we may define the cluster *optimal* with respect to a request as the cluster on which F achieves a maximum value (and $E = 1 - F$, a minimum value). The ideal cluster-based search strategy is that which always finds the optimal cluster. In section (i) we showed that on the Cranfield data-base the effectiveness of an ideal cluster-based strategy is substantially greater than is attainable by any linear associative strategy. In constructing real cluster-based retrieval strategies we aim to approximate to this ideal.

Cluster search strategies are based on matching functions between requests and clusters. A *global* cluster search strategy calculates a matching value between a request and each cluster and selects the cluster on which the highest matching value is achieved. Global cluster searches are very inefficient: substantially less efficient than linear associative searches. The most efficient cluster search strategies are *simple downward searches.* A request is first matched with each of the clusters at the top of the hierarchy (or at some selected starting level) and the best matching cluster is selected. The request is then matched with each of the clusters which are immediately included in the best matching cluster (its filial set), and again the best matching cluster is selected, and so on. The *simple stopping rule* terminates the search when at some stage the best match which is currently achieved is worse than the best match achieved at the preceding stage. The operation of a simple downward search with the simple stopping rule is illustrated in Fig. 8.

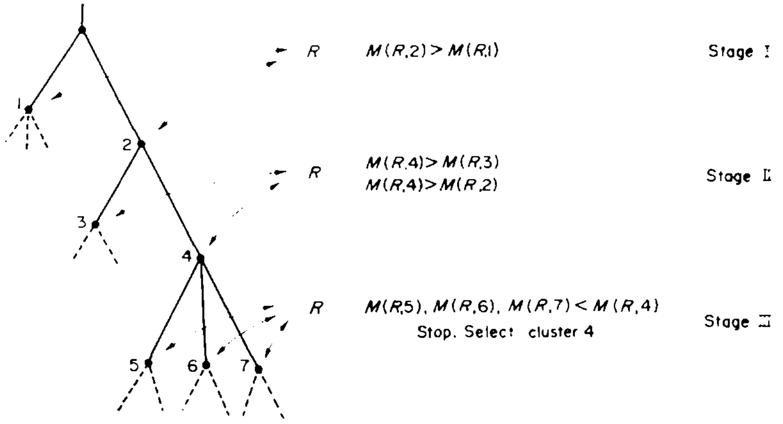

FIG. 8. The operation of a simple downward cluster search using the simple stopping rule. $M(R, C)$ is the value of a matching function between a request R and a cluster C.

Choice of a matching function between requests and clusters is guided by two requirements. First, the highest value of a matching function should ideally always be achieved on the optimal cluster. Secondly, in the interest of efficiency, the amount of information about the clusters which must be stored in order to calculate matching values should be as small as possible.

We have experimented with two ways of representing clusters for matching with requests. One depends on a choice for each cluster of a *typical* document. We represent a cluster at level θ in the hierarchy by the representative of a maximally linked document; that is, a document which is linked to a maximum number of other documents at a threshold θ of the association measure (see VAN RIJSBERGEN [43]). Where there are two or more maximally linked documents an arbitrary choice is made. (See Fig. 9.) Other ways of selecting a typical member as representative of a cluster have been described by HYVARINEN [44] and by SILVESTRI *et al.* [45] (in the context of choice of a typical specimen to represent a group of organisms). In the other approach the representative of each cluster is some function of the representatives of its component documents. It may also be a function of such parameters of the cluster as its size. The latter approach requires storage of more information about clusters and their members in the data-structure, but in our experiments on the Cranfield data-base it led to more effective retrieval.

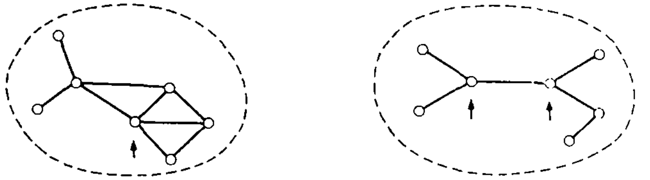

FIG. 9. Selection of maximally linked documents to typify clusters. In the graph vertices represent documents and edges join pairs of documents with association value θ or more, whether θ is the splitting-level of a cluster. In cluster A there is a unique maximally linked document. In cluster B there are two maximally linked documents and an arbitrary choice has to be made.

(v) *Experimental evaluation of some cluster-based strategies*

The data-structure used in our experiments is described in Appendix 1. Our initial experiments using the entire single-link hierarchy achieved very poor effectiveness. At the highest levels in the hierarchy, the majority of the clusters obtained on the Cranfield database consist of single aberrant documents which are closely associated with no other documents. In many cases a downward search strategy retrieved a single one of these documents in response to a request, when all the relevant documents were to be found in a cluster at a much lower level in the hierarchy. The problem of aberrant individuals which form small isolated clusters arises in many classification problems. Such individuals are sometimes assigned to a separate "garbage" cluster (RUBIN [46]). In the following experiments we deal with aberrant documents by truncating the hierarchy at a high level and searching only clusters below this level. Our evaluations of effectiveness are therefore conservative since for each experiment we evaluate retrieval effectiveness for the whole collection, not just for the subset which is searched. The effect can be seen in Fig. 10 which shows the effectiveness of one of the search strategies both for the whole collection and for the subset which was searched. In practice it appears that the best way to deal with aberrant documents may be to reallocate each of them to the cluster below the truncation level which it matches best. We have not done this in the following experiments because there is some doubt whether

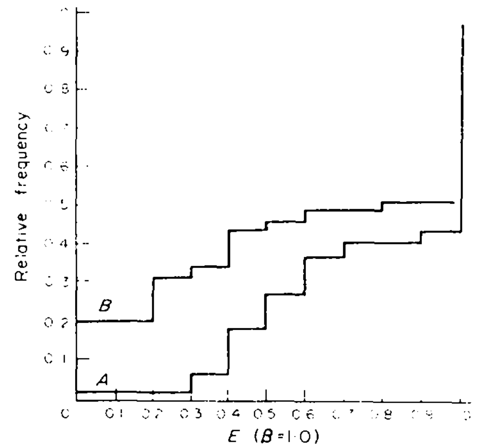

FIG. 10. Cumulative frequency of values of E achieved by the cluster-based strategy $E3$. A, evaluated on the entire document collection. B, evaluated on the subset of the document collection which remains when aberrant documents have been rejected by truncating the hierarchy of document clusters.

this would be the best way to deal with aberrant documents for document collections which show different patterns of clustering. On the Cranfield data-base, reallocation of aberrant documents would yield a substantial improvement in the effectiveness of retrieval, but it is dangerous to attach too much significance to improvement in effectiveness achieved by *ad hoc* measures applied to a single document collection.

The cluster representatives used in the retrieval experiments were as follows:

A. By a binary string in which a 1 in the *i*th position indicates occurrence of the *i*th index term in a maximally linked document of the cluster;

B. By a binary string in which a 1 in the *i*th position indicates presence of the *i*th index term in more than one document in the cluster;

C. By a binary string in which a 1 in the *i*th position indicates presence of the *i*th index term in more than $\log_2 |C|$ documents, where $|C|$ is the number of documents in the cluster.

The matching function used with criteria A, B and C is

$$\frac{|X \Delta Y|}{|X|+|Y|}$$

where X is the set of index terms in the cluster representative and Y is the set of index terms in the request representative.

The cumulative frequency distributions of values of E achieved by simple downward searches using matching functions based on criteria A–C are shown in Figs. 11–13, respectively. In Fig. 14 we show the results achieved by a global cluster search using criterion C for cluster representation.

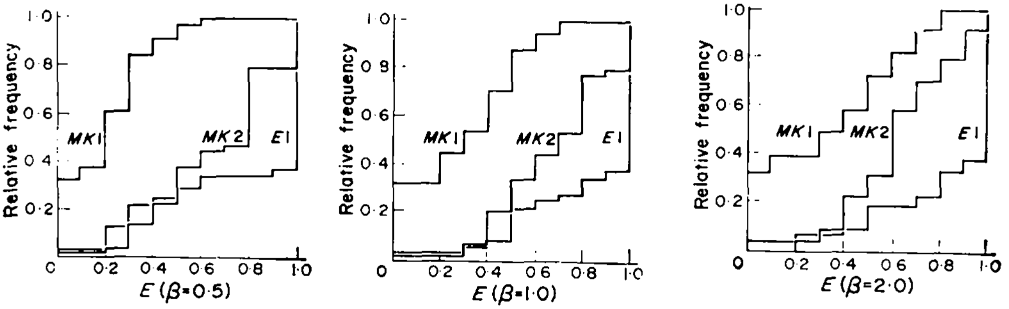

FIG. 11. Comparison of the effectiveness of the simple downward cluster search strategy using criterion A for cluster representation ($E1$) and the ideal strategies $MK1$ and $MK2$.

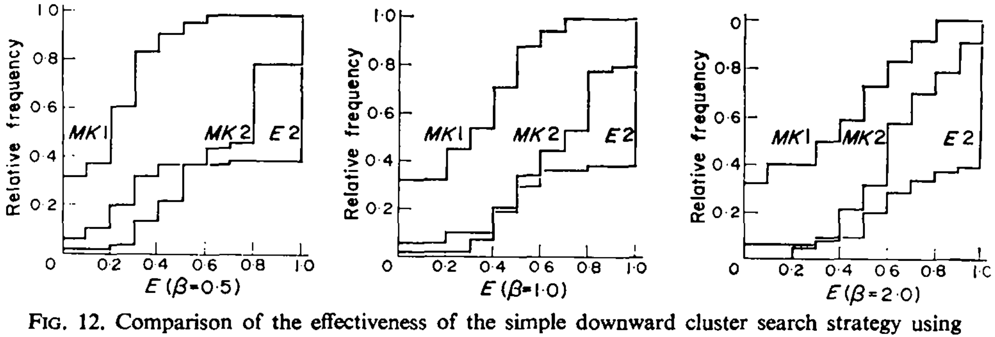

FIG. 12. Comparison of the effectiveness of the simple downward cluster search strategy using criterion B for cluster representation ($E2$) and the ideal strategies $MK1$ and $MK2$.

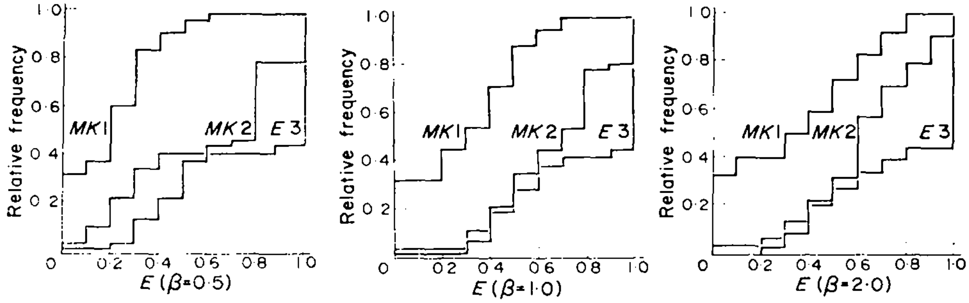

FIG. 13. Comparison of the effectiveness of the simple downward cluster search strategy using criterion C for cluster representation (*E*3) and the ideal strategies *MK*1 and *MK*2.

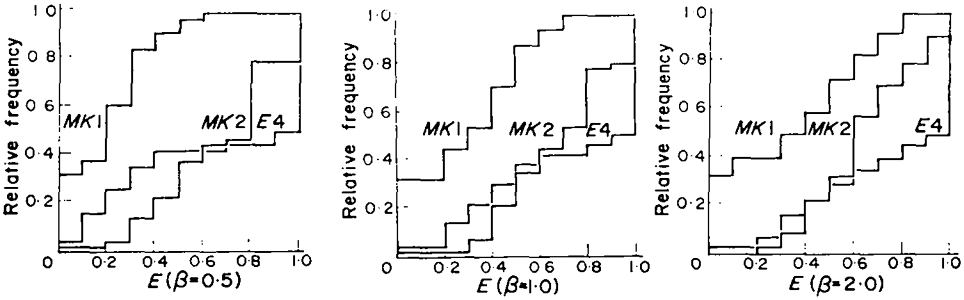

FIG. 14. Comparison of the effectiveness of the global cluster search strategy (*E*4) and the ideal strategies *MK*1 and *MK*2.

In each figure two of the benchmarks described in section (ii) are shown. The benchmark labelled *MK*1 is the ideal effectiveness attainable by retrieval of the optimal cluster in response to each request. The benchmark labelled *MK*2 is the ideal effectiveness attainable by linear associative retrieval with cut-off at an optimal single rank position. To approach the effectiveness of *MK*2 in practice an associative retrieval strategy would have to include some procedure for learning the optimal cut-off position. We have chosen *MK*2 as a benchmark rather than the ideal effectiveness attainable by linear associative retrieval with choice of an optimal cut-off for each request (*MK*3 shown in Fig. 3), because as indicated in section (i) it appears to be a fairer estimate of what is likely to be attainable in practice by a linear associative strategy. In Table 2 we give the mean values of *E* achieved by the various strategies.

The following tentative conclusions can be drawn from the results shown in Figs. 11–14 and Table 2:

1. A simple downward search using criterion C for cluster representation proved to be more effective than the other two simple downward search strategies. This is not surprising because criterion C takes account of more information about each cluster than criterion A or criterion B.

2. Global search using criterion C for cluster representation proved to be scarcely more effective than a simple downward search using the same criterion. The result is, at first sight, surprising, but it is also encouraging. It suggests that the highly efficient simple downward search strategies are appropriate for cluster-based retrieval, and that improvement in effectiveness is best sought by experimenting with more sophisticated cluster representatives

TABLE 2. MEAN VALUES OF E ACHIEVED BY THE IDEAL STRATEGIES $MK1$ AND $MK2$ AND THE EXPERIMENTAL STRATEGIES $E1$–4

	$\beta=0{\cdot}5$	$\beta=1{\cdot}0$	$\beta=2{\cdot}0$
$MK1$	0·16	0·25	0·29
$MK2$	0·63	0·64	0·58
$E1$	0·74	0·79	0·81
$E2$	0·68	0·73	0·77
$E3$	0·64	0·70	0·73
$E4$	0·63	0·68	0·71

and matching functions. This is encouraging because simple downward search strategies are the most efficient possible. Choice of alternative cluster representatives and matching functions can lead only to relatively slight loss of efficiency.

3. All the cluster-based retrieval strategies show greater effectiveness the greater the relative importance attached to precision (i.e. the *smaller* the value of β).

4. None of the cluster-based retrieval strategies approaches the effectiveness of the ideal cluster-based retrieval strategy $MK1$.

5. The global cluster search using criterion C for cluster representation approaches the effectiveness of the ideal linear associative strategy $MK2$ except when more importance is attached to recall than to precision (i.e. when β is greater than 1).

6. The simple downward search using criterion C for cluster representation approaches the effectiveness of the ideal linear associative strategy $MK2$, except when more importance is attached to recall than to precision.

(vi) *Developments of cluster-based retrieval*

We outline briefly some of the ways in which we intend to develop cluster-based retrieval strategies to improve their efficiency and effectiveness.

Extension to large document collections. The DYNALINK algorithm described by VAN RIJSBERGEN [43] cannot be applied to collections of more than about 2000 documents, because computation time shows an order n^2 dependence. It is, however, possible to construct a hierarchic clustering much more efficiently using a modification of this approach. The single-link cluster method is applied to a sample of the document collection to construct a *core* clustering. The remaining documents are then allocated to clusters in the core clustering as follows. A matching function is defined between core clusters and documents. Each document is then assigned to a cluster using a simple downward search. The efficiency of this method is estimated in section (i). The effectiveness of retrieval strategies based on hierarchic clustering of this kind depends on the adequacy with which the core clustering represents the structure of the entire collection.

The time taken by the DYNALINK algorithm to update the single-link clustering on a set of documents when an additional set of documents is incorporated is a measure of the extent to which the clustering has been modified. If many new clusters are created and many existing clusters are amalgamated, the updating is relatively slow; if the new documents fit readily into existing clusters updating is relatively fast. We may use this fact to find out how large a random sample of a document collection need be clustered to provide an adequate core clustering. Suppose that we select n documents at random from a collection, cluster them using DYNALINK, and then incorporate into the resultant system of clusters successive increments of *n* randomly selected documents. If the relative extent of modification falls steadily and at the *i*th increment falls below some chosen low threshold, it is reasonable to infer that the clustering obtained when the *i*th set of documents is incorporated is an adequate core clustering. On the Cranfield collection, for example, we have found that a core clustering of 50 documents closely approximates the cluster structure of the entire collection.

Using this approach it would be possible to generate a hierarchic clustering on at least 50,000 documents provided that a core clustering on a random sample of size 500 proved to be adequate.

Generalization of the cluster search strategy. Particularly when high importance is attached to recall it may be profitable, despite the loss in efficiency, to use a search strategy which inspects more clusters than does the simple downward search [described in section (iv)].

The simple downward search may be generalized by allowing the selection of a cluster at each stage to be determined by the highest matching value which would be achieved at the *n* subsequent stages *if* it were selected. Informally speaking, the search strategy looks several stages ahead before making a decision. The simple stopping rule can be generalized by causing the search to terminate only when the best matching value achieved at each of the *n* preceding stages is greater than that achieved at the $n+1$'th preceding stage. Informally speaking, the stopping rule looks back several stages before making a decision. The fact that on the Cranfield collection a global cluster search did not improve on the effectiveness of a simple downward search casts some doubt on the likely effectiveness of these generalizations.

A different, and perhaps more promising, generalization of the simple downward search strategy allows the search to proceed down more than one branch of the hierarchy whenever at some stage the match between a request and the best matching cluster is only slightly better than its match with another cluster. With this strategy it would be appropriate to retrieve more than one cluster if the highest matching value eventually achieved were approximated by matching values achieved by clusters on other branches of the hierarchy.

Request expansion. Automatic classification of index terms based on statistical measures of concurrence of terms in document representatives has been used to generate thesauri. Several workers, notably STILES [47] and SPÄRCK JONES [26, 1], have attempted to improve recall by request expansion based on an automatically generated thesaurus. Some or all of the index terms which occur in the same categories of the thesaurus as the terms in the request are added to the request.

As pointed out by DOYLE [37], a term classification can be generated indirectly *via* a hierarchic document clustering. Each term cluster consists of terms which occur in more than some specified proportion of the documents in a single document cluster. Expansion of requests using a term clustering generated in this way is an appropriate adjunct to cluster-based retrieval and involves relatively little extra computation at search time.

Improvement of matching functions. The cluster representatives which we have used as a basis for calculation of matching values between requests and clusters are somewhat arbitrary. Intuition is probably a poor guide in seeking a matching function which will achieve a maximum value in a downward search on the optimal document cluster.

It is probable that the efficiency of simple downward search strategies can be improved by using matching functions which allow numerical weighting of the index terms in document cluster representatives. At each stage in a simple downward search a single cluster is selected from a set of clusters with the same parent (see Fig. 8). Let $C_1 \dots C_n$ be a set of clusters with the same parent. If f_{ij} is the frequency of the ith index term in the jth cluster

$$W_{ij} = \frac{f_{ij}}{\sum\limits_{j=1 \cdots n} f_{ij}}$$

is a measure of the *evidence* for selecting C_j from $C_1 \dots C_n$ given by the occurrence of the ith index term in a request. It is difficult to find appropriate ways of combining evidence weights in calculating matching values between requests and clusters, because index terms are not statistically independent.

An alternative way of obtaining index term weights is to use learning strategies which seek an assignment of weights which optimizes the effectiveness of a cluster-based retrieval strategy on test sets of requests.

Summary of results

1. Hierarchic cluster-based document retrieval strategies are substantially more *efficient* than linear associative strategies, and theoretical reasons are given for believing that their *effectiveness* in terms of recall and precision can equal, or improve on, the effectiveness of linear associative strategies.
2. A new approach to the comparison of effectiveness of experimental document retrieval strategies is suggested and is related to existing evaluation methods.
3. It is shown that on the Cranfield data-base the theoretical upper bound to retrieval effectiveness achievable by a hierarchic cluster-based strategy is substantially better than the theoretical upper bound to retrieval effectiveness achievable by a linear associative strategy.
4. When applied to the Cranfield data-base, one of the hierarchic cluster-based strategies tested proved to approach in effectiveness the theoretical upper bound to effectiveness of linear associative retrieval at a single cut-off; but all cluster-based strategies tested fell far short of the theoretical upper bound to hierarchic cluster-based retrieval.
5. The DYNALINK algorithm which constructs hierarchic clusterings for use in cluster-based retrieval can be applied to collections of up to about 2000 documents. It can handle document representatives sequentially and hence can be applied to growing document collections. We describe a modification of this approach which could be used to cluster much larger document collections.
6. We outline ways in which the relatively crude cluster-based retrieval strategies which we have tested may be improved in the quest for greater effectiveness without substantial loss of efficiency.

Finally, we conclude that the results reported in this paper should be regarded with caution until they have been substantiated on other data-bases. It is difficult to decide whether experimental evaluations of effectiveness and efficiency of retrieval strategies on a particular data-base reflect properties of the retrieval strategies or peculiarities of the data-base.

Acknowledgements—The authors thank Dr KAREN SPÄRCK JONES, Dr J. K. M. MOODY and Dr P. L. HOLMES for many stimulating discussions. The Cranfield data-base was generously made available by Dr SPÄRCK JONES. The work was supported by an OSTI Research Grant and was carried out during the tenure of a Royal Society Scientific Information Research Fellowship by N. JARDINE. Facilities were kindly provided by the King's College Research Centre, Cambridge and the Cambridge University Computer Laboratory. We thank Mrs NORMA SOBER who drew the figures.

REFERENCES

[1] K. SPÄRCK JONES and D. M. JACKSON: *Inform. Stor. Retr.* 1970, **5**, 175–207.
[2] C. CLEVERDON, J. MILLS and M. KEEN: *Factors Determining the Performance of Indexing Systems*, 2 Vols., 1966, College of Aeronautics, Cranfield.
[3] E. M. KEEN: Evaluation parameters, Report ISR 13 to the National Science Foundation, 1967, Cornell University, Department of Computer Science, Chap. II.
[4] J. A. SWETS: Effectiveness of information retrieval methods, Bolt Beranek and Newman, Rept. 47CAFCRL-67-0412, 1967, Cambridge, Mass.
[5] S. E. ROBERTSON: *J. Docum.* 1969, **25**, 1–27.
[6] F. W. LANCASTER: *Information Retrieval Systems: Characteristics, Testing and Evaluation*, John Wiley, New York (1968).
[7] C. CLEVERDON: *J. Docum.* 1970, **25**, 55–67.
[8] S. SIEGEL: *Nonparametric Statistics for the Behavioral Sciences*, McGraw-Hill, New York (1956).
[9] G. SALTON: *Automatic Information Organisation and Retrieval*, McGraw-Hill, New York (1968).
[10] P. F. WINDLEY: *Comput. J.* 1960, **3**, 84–88.
[11] L. GOODMAN and W. KRUSKAL: *J. Am. statist. Ass.* 1954, **49**, 732–764.
[12] L. GOODMAN and W. KRUSKAL: *J. Am. statist. Ass.* 1959, **54**, 123–163.
[13] J. L. KUHNS: In: *Statistical Association Methods for Mechanised Documentation* (Eds. M. E. STEVENS, V. E. GUILIANO and L. B. HEILPRIN), pp. 33–39, U.S. Department of Commerce, Washington D.C. (1965).
[14] R. R. SOKAL and P. H. A. SNEATH: *Principles of Numerical Taxonomy*, W. H. Freeman, San Francisco, (1963).
[15] R. M. CORMACK: *Jl R. statist. Soc. B*, (in press).
[16] P. SWITZER: In: *Statistical Association Methods for Mechanised Documentation* (Eds. M. E. STEVENS, V. E. GUILIANO and L. B. HEILPRIN), pp. 163–171, U.S. Department of Commerce, Washington D.C. (1965).
[17] V. E. GUILIANO: In: *Statistical Association Methods for Mechanised Documentation* (Eds. M. E. STEVENS, V. E. GUILIANO and L. B. HEILPRIN), pp. 25–32, U.S. Department of Commerce, Washington D.C. (1965).
[18] M. E. MARON and J. L. KUHNS: *J. ACM*, 1960, **7**, 216–244.
[19] G. H. BALL: *Proc. Fall jt. comput. Conf.* 1966, **27**, 533–559.
[20] I. HARRISON: *Metra* 1968, **7**, 513–528.
[21] N. JARDINE and R. SIBSON: *Mathematical Taxonomy*, John Wiley, New York (1971).
[22] K. SPÄRCK JONES: *J. Docum.* 1970, **26**, 89–107.
[23] R. A. FAIRTHORNE: *Proc. Brit. Soc. Int. Bibl.* 1947, **9**, 35.
[24] I. J. GOOD: Speculations concerning information retrieval, Research Report RC-78, 1958, IBM Research Centre, Yorktown Heights, New York.
[25] J. D. BROFFIT, H. L. MORGAN and J. V. SODEN: On some clustering techniques for information retrieval, Report ISR 11 to the National Science Foundation, Sect. IX, 1966, Cornell University, Department of Computer Science.
[26] K. SPÄRCK JONES: *Proc. IFIP Congress* 1968, Booklet G, 5–9.
[27] R. M. NEEDHAM and K. SPÄRCK JONES: *J. Docum.* 1964, **20**, 3–15.
[28] J. J. ROCCHIO: *Document Retrieval Systems—Optimisation and Evaluation*, Report ISR 10 to the National Science Foundation, 1966, Harvard University Computer Laboratory, Chap 4.
[29] R. M. NEEDHAM: The theory of clumps. II. Rept. ML 139, 1961, Cambridge Language Research Unit.
[30] R. T. DATTOLA: A test algorithm for automatic classification, Report ISR 14 to the National Science Foundation, 1968, Cornell University, Department of Computer Science.
[31] C. C. GOTLIEB and S. KUMAR: *J. ACM* 1968, **15**, 493–513.
[32] J. G. AUGUSTSON and J. MINKER: *J. ACM* 1970, **17**, 571–588.
[33] J. G. AUGUSTSON and J. MINKER: *Jl Am. Soc. inf. Sci.* 1970, **21**, 101–111.
[34] P. K. T. VASWANI and J. B. CAMERON: The National Physical Laboratory experiments in statistical word associations and their use in document indexing and retrieval, National Physical Laboratory, Division of Computer Science, 1970, Publ. 42.
[35] R. E. BONNER: *IBM Jl Res. Dev.* 1964, **8**, 22–32.
[36] H. BORKO and M. BERNICK: *J. ACM* 1963, **10**, 151–162.
[37] L. B. DOYLE: In: *Statistical Association Methods for Mechanised Documentation* (Eds. M. E. STEVENS, V. E. GUILIANO and L. B. HEILPRIN), pp. 15–24, U.S. Department of Commerce, Washington D.C. (1965).
[38] H. LITOFSKY: *Utility of Automatic Classification Systems for Information Storage and Retrieval*, Doctoral Dissertation, 1969, University of Pennsylvania.
[39] N. S. PRYWES and H. LITOFSKY: *Proc. Spring jt. comput. Conf.* 1970, 323–331.

[40] J. H. WARD: *J. Am. statist. Ass.* 1963, **58**, 236–244.
[41] N. JARDINE and R. SIBSON: *Comput. J.* 1968, **11**, 177–184.
[42] D. M. JACKSON: *Automatic Classification and Information Retrieval*, Doctoral Dissertation, 1969, University of Cambridge, England.
[43] C. J. VAN RIJSBERGEN: *Comput. J.* 1971 (in press).
[44] L. HYVARINEN: *Nord. Tidskr. Inf.-Behandl.* 1968, **2**, 83–89.
[45] L. SILVESTRI, M. TURRI, L. R. HILL and E. GILARDI: In: *Microbial Classification* (Eds. G. C. AINSWORTH and P. H. A. SNEATH), pp. 333–360, Cambridge University Press (1967).
[46] J. RUBIN: *J. theor. Biol.* 1967, **15**, 103–144.
[47] H. E. STILES: *J. ACM* 1961, **8**, 271–279.

APPENDIX

A data-structure for experimental cluster-based document retrieval

The data-structure used in our experiments on hierarchic cluster-based retrieval is shown diagrammatically in Fig. 15. The tree represents a hierarchic clustering of the set of objects $\{A, B, C, D, E, F\}$. Each node represents a cluster at its splitting-level in the hierarchy and corresponds to a cell in the data-structure. Each cell is divided into four fields. The fiist field contains the level of the corresponding node. The third and fourth fields contain pointers which indicate the parent and filial set of each node. These pointers make it possible to move around the data-structure systematically. Only these three fields are used in generating the clustering (see VAN RIJSBERGEN [43]). Note that the level of a terminal node is set to -1 to allow clustering at level 0.

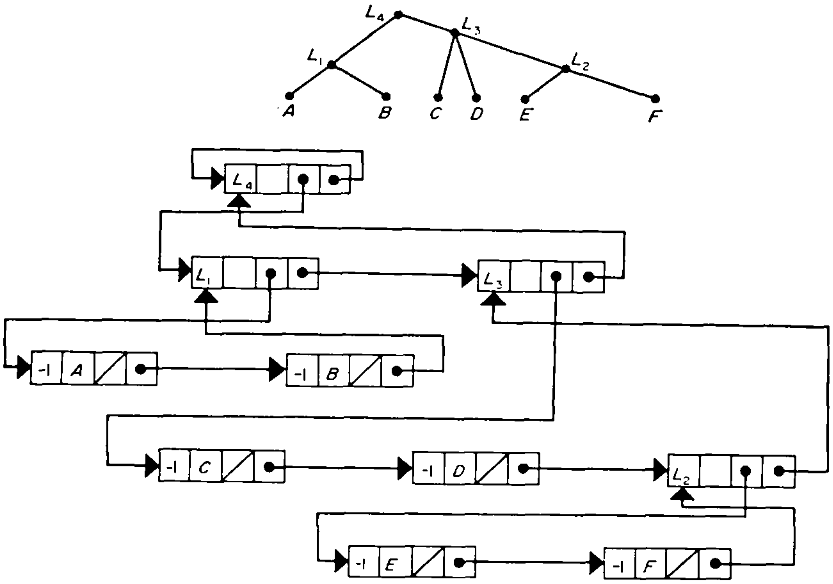

FIG. 15. A data-structure for experimental cluster-based document retrieval.

The second field is used during retrieval experiments. It contains a pointer to the information required to compute a matching value between a request and the node. This information varies in complexity. In the simplest case the information is the binary string representing a document chosen to typify the cluster. More complicated cases arise when the information consists of a cluster representative which is a function both of the representatives of the documents in the cluster and of such parameters of clusters as their size. In such cases rather than compute a chosen representative before an experiment, it is computed at search time. This entails storing the representatives of the documents which belong to each cluster. The pointer in field two now points to a list of cluster members and to any cluster parameters needed to compute the matching function.

We emphasize that this data-structure is designed for experimental purposes. Flexibility in the system is obtained at the cost of extra storage and loss in efficiency. In an operational situation the chosen cluster representatives would be computed in advance rather than at search time. Field two of a cell in the data-structure would contain a pointer to the cluster representative and both the list of cluster members and the dictionary of document representatives would be eliminated.