Information Processing & Management Vol. 24, No. 5, pp. 577–597, 1988
Printed in Great Britain.

0306-4573/88 $3.00 + .00

TRENDS IN . . . A CRITICAL REVIEW

RECENT TRENDS IN HIERARCHIC DOCUMENT CLUSTERING: A CRITICAL REVIEW

PETER WILLETT
Department of Information Studies, University of Sheffield
Western Bank, Sheffield, S10 2TN, U.K.

(*Received* 29 *October*; *accepted in final form* 26 *January* 1988)

Abstract—This article reviews recent research into the use of hierarchic agglomerative clustering methods for document retrieval. After an introduction to the calculation of interdocument similarities and to clustering methods that are appropriate for document clustering, the article discusses algorithms that can be used to allow the implementation of these methods on databases of nontrivial size. The validation of document hierarchies is described using tests based on the theory of random graphs and on empirical characteristics of document collections that are to be clustered. A range of search strategies is available for retrieval from document hierarchies and the results are presented of a series of research projects that have used these strategies to search the clusters resulting from several different types of hierarchic agglomerative clustering method. It is suggested that the complete linkage method is probably the most effective method in terms of retrieval performance; however, it is also difficult to implement in an efficient manner. Other applications of document clustering techniques are discussed briefly; experimental evidence suggests that nearest neighbor clusters, possibly represented as a network model, provide a reasonably efficient and effective means of including interdocument similarity information in document retrieval systems.

1. INTRODUCTION

Clustering involves the grouping of similar objects, and has been practiced, consciously or unconsciously, for many thousands of years. The development of computer technology has resulted in *cluster analysis*, a multivariate statistical technique that allows the production of classifications by purely automatic means, the grouping generally being achieved by means of the calculation of all of the interobject similarities. Cluster analysis is often referred to as *classification* but this is potentially misleading. Classification normally refers to the *assignment* of objects to predefined classes whereas cluster analysis requires the *identification* of these classes; thus, clustering must precede classification in the analysis of a dataset. Cluster analysis was first studied intensively in the biological sciences, but is now used in a wide range of disciplines including archaeology, astronomy, computer science, geology, market research, marketing, medicine and psychiatry [1–10].

Two types of clustering have been studied in the context of document retrieval systems: the clustering of documents, usually on the basis of the terms that they have in common, and the clustering of terms on the basis of the documents in which they cooccur [11,12]. The availability of a term clustering allows each term in a document and/or query to be replaced by the identifier of the cluster containing that term. Alternatively, a query can be expanded by adding terms from each of the groupings which contain one of the original query terms. In both cases, the clusters provide a mechanism for the identification of additional matches between the sets of document and query terms, and thus for increasing the recall of a document retrieval system. Early enthusiasm for such techniques [13] was followed by a realization that the empirical clustering procedures which were used did not always lead to increases in retrieval effectiveness and could, indeed, reduce search performance in some cases [14]. More recently, probabilistic techniques have been used to

investigate the statistical relationships which exist between terms in document collections. This work has provided a firm theoretical basis for the use of term cooccurrence information, but experimental tests show that it does not seem possible to use such information to increase system performance in practical retrieval environments [15,16]. Accordingly, this review focuses on the clustering of documents.

Two main justifications have been put forward for the use of document clustering techniques. Early studies by Salton and his co-workers [17] used clustering as a way of improving the *efficiency* of best match searching in information retrieval systems; at the time when the work was carried out, best match searching was implemented by a serial scan of a database and thus the number of query–document similarity calculations could be reduced if documents were grouped into clusters. Alternatively, it has been argued that clustering methods can increase the *effectiveness* of document retrieval [18,19]. Conventional document retrieval systems involve the matching of a query against individual documents; a clustered search, in which a query is compared with clusters of documents, may achieve better levels of retrieval effectiveness since the file organization and the search strategy take some account of the relationships in content which exist between the documents in the database.

The article is organized as follows. Section 2 discusses the measurement of interdocument similarity and some of the more important methods that have been used for document clustering, with particular attention being paid to hierarchic agglomerative clustering methods and to algorithms for their efficient implementation. Section 3 discusses the application of work on cluster validity to document clustering while Section 4 describes the search strategies that are available for retrieval from clustered document files. Section 5 reviews the main studies that have been carried out of the effectiveness of cluster searching and Section 6 closes with related studies, some conclusions, and suggestions for future work.

2. CLUSTER ANALYSIS IN INFORMATION RETRIEVAL

2.1 *The measurement of interdocument similarity*

Cluster analysis methods are all based in some way on measurements of the similarity between a pair of objects, these objects being either individual documents or clusters of documents. In general, the determination of similarity between a pair of objects involves three major steps, these being the selection of the variables that are to be used to characterize the objects, the selection of a weighting scheme for these variables, and the selection of a similarity coefficient to determine the degree of resemblance between the two attribute vectors. An extremely detailed account of the measurement of similarity is given by Sneath and Sokal [2] while van Rijsbergen discusses the topic in the particular context of document clustering [12].

Two types of variable have been used for the characterisation of documents in document clustering research. Citation clustering, which is used in studies of the sociology of science, involves measuring the degree of similarity between a pair of documents by the citations which they share in common; an example of work in this area is described by Small and Sweeney [20]. However, much more interest has been shown in the use of clustering techniques that assume that the documents are represented by lists of manually or automatically assigned lists of index terms, keywords, or thesaural terms that describe the content of the documents; the latter type of document representation forms the basis of the present review.

It may seem intuitively reasonable that some characteristics are of more importance than others in determining the similarity relationships that are present within a dataset. Such ideas are commonly used when queries need to be matched against a file of documents (e.g., the well-known inverse document frequency weighting scheme [21,22]). Sneath and Sokal argue strongly against the use of weighting schemes in cluster analysis and present no less than seven reasons in favor of the equal weighting of all of the variables in a study; some small-scale experiments by Willett showed that the use of term weighing made

little difference to the clusterings produced by the single linkage method when it was used with three document test collections [23].

There are very many types of similarity coefficient available for determining the degree of similarity between a pair of objects. Thus, Sneath and Sokal describe four main classes of coefficient: *distance coefficients*, *association coefficients*, *probabilistic coefficients*, and *correlation coefficients* [2].

Distance coefficients, such as the Euclidean distance, have been used very extensively in cluster analysis, owing to their simple geometric interpretation. However, a major limitation of the Euclidean distance in the information retrieval context is that it can lead to two documents being regarded as highly similar to each other, despite the fact that they share no terms at all in common. The Euclidean distance is thus not widely used for document clustering, the only notable exception being its use in Ward's clustering method. Association coefficients, conversely, have been very widely used for document clustering. The simplest association coefficient, the *simple matching coefficient* is just c, the number of terms common to a pair of documents having a and b terms, respectively. This has the disadvantage that the similarity is not *normalized* (i.e., it takes no account of the numbers of terms in each of the documents). This is a severe limitation [12], and thus most coefficients that have been used try to normalize c in some way. Two commonly used examples of normalized association coefficients are the Dice coefficient

$$2c/(a + b)$$

and the Jaccard coefficient

$$c/(a + b - c)$$

Other such coefficients are described by Salton and McGill [22]. The coefficients may readily be generalized to nonbinary data, as would be required if the documents had been automatically indexed so that they were characterized by the frequency of occurrence of each keyword, rather than by their mere presence or absence. An analogous situation arises if the similarity needs to be calculated between two clusters, each of which may consist of some or many individual documents, during the course of a clustering procedure. Probabilistic coefficents have been used in recent work by El-Hamdouchi on clustering methods in which the main criterion for the formation of a cluster is that the documents in it have a maximal probability of being jointly corelevant to a query [24]. There do not seem to have been any reports of the use of correlation coefficients for document clustering.

Experimental comparisons of different similarity coefficients [23–27] suggest that the similarity coefficient may affect the clustering that is obtained, but there seems to be little consensus as to which types of coefficient are most generally applicable. Sneath and Sokal suggest that one should generally use the simplest type of coefficient that seems appropriate, and point out that such coefficients are often monotonic with more complex measures [2].

2.2 *Document clustering methods*

Two broad classes of clustering method have been used for the grouping of documents: nonhierarchic and hierarchic.

Nonhierarchic clustering methods divide a dataset into a series of subsets, with similar objects in the same cluster being separated from nonsimilar objects placed in different clusters. Such a *partition* describes a classification in which there are no hierarchic relationships between the various clusters that have been identified by the clustering procedure. An optimal solution to the problem of finding a partition may be obtained by the simple expedient of defining some criterion of the "goodness" of a classification, and then systematically generating and evaluating all possible partitions to see which of them best satisfies this criterion. The problem here is that a complete enumeration is quite infeasible for datasets of anything but trivial size [28]; because of this, a range of heuristics has been

described which permit suboptimal partitions to be obtained at a much lower computational cost.

The most common approach tries to partition the set of N objects in a dataset into C clusters so as to minimize the total within-cluster sum of squares about the C cluster centroids. This simple and intuitively appealing idea forms the basis for a large number of nonhierarchic methods that proceed directly from the object descriptions and that differ only slightly in the exact manner in which they are implemented [5,6,29]. Several nonhierarchic procedures have been used for document clustering, the main emphasis being on the use of heuristic clustering procedures that can achieve a grouping of the documents at a low computational cost. The methods have generally involved a number of input parameters that are used to control the clustering (e.g., the number of clusters required, minimum or maximum cluster sizes, and threshold document-cluster similarity levels [17]). These methods are attractive from a computational viewpoint since their time requirements are typically of order $O(N)$ to $O(N \log N)$ for the clustering of N documents. However, the classifications are rather arbitrary in operation since the final clusters may depend on the order in which the document file is processed, the random selection of documents as initial cluster centers, or the exact parameter values that are used. It is possible to overcome some of these limitations [30,31], but retrieval tests suggest that the use of the methods leads to substantial reductions in effectiveness when compared to searches of the unclustered file [17,32]. As noted previously, the main aim of this early work on document clustering was to improve the efficiency of best match searching. There has, however, been a substantial body of work that has resulted in efficient algorithms that allow best match searches to be executed almost as efficiently as can Boolean searches [33]; accordingly, there seems little reason to consider nonhierarchic methods further, and the remainder of this review will concentrate on the use of hierarchic clustering methods for document retrieval.

Hierarchic clustering methods result in binary treelike classifications in which small clusters of documents that are judged to be strongly similar to each other are nested within larger and larger clusters containing documents that are less similar. The single cluster containing the entire collection is represented by the root of the tree while the individual documents reside in the leaves; nodes in the body of the tree correspond to clusters that are formed during the operation of the clustering procedure.

Two main strategies are available for the construction of hierarchic classifications [2,3,34]. An *agglomerative* clustering strategy proceeds through a total of $N - 1$ fusions for a collection of N documents and results in the classification being built upwards from the leaves, with the smallest clusters being generated first and with the final fusion resulting in the root of the tree. Alternatively, a *divisive* clustering strategy may be adopted in which a single initial cluster is subdivided into smaller and smaller groups of documents. Divisive clustering methods generally result in *monothetic* classifications in which all of the documents in a given cluster must contain certain terms if they are to belong to it; such clustering methods are of less use for applications in document retrieval than *polythetic* classifications, where each document in a cluster has some, or many, terms in common with each of the other documents in that cluster but where there are no specific terms required for cluster membership [12]. Polythetic hierarchic classifications are normally generated using the agglomerative methods, which are by far the most popular type of clustering procedure. They were developed primarily for applications in the life sciences, where the hierarchic classifications could be compared with traditional biological taxonomies in which specimens are grouped into species and species grouped into genera. The production of such taxonomies was the first major application of cluster analytic techniques, and it is thus not surprising that the hierarchic methods established a supremacy over the nonhierarchic methods that has remained even when quite different sorts of application are considered. Extensive discussions of taxonomic applications are given by Sneath and Sokal [2] and by Clifford and Stephenson [34] while good general descriptions of hierarchic clustering are provided by Cormack [1], Everitt [3] and by Dubes and Jain [5].

The most famous of the agglomerative clustering methods is *single linkage*, or *nearest neighbor*, in which clusters are formed on the basis of the similarity between the most sim-

ilar pair of documents, one of which is in each of a pair of clusters. The clusters formed by this method have the property that any cluster member is more similar to at least one member of that cluster than it is to any member of any other cluster; hence the name nearest neighbor. In graph theoretic terms, the single linkage clusters at some similarity level are the connected components of a graph. A characteristic of this method is its tendency to form loosely bound clusters with little internal cohesion, a phenomenon referred to as *chaining* and that has resulted in several attempts to produce related methods that do not suffer from this defect [35,36].

An alternative formulation [8] of the single linkage method assumes that an $N \times N$ interdocument similarity matrix is available where the element SIM[K,L] contains the similarities between two documents K and L. Then K and L are defined as belonging to the same cluster at some similarity threshold T if there exists a chain of intermediate documents

$$\text{SIM}[\text{K},\text{X}_1],\text{SIM}[\text{X}_1,\text{X}_2]\ldots\text{SIM}[\text{X}_P,\text{L}] \qquad (1 \leq P \leq N)$$

linking them together where each similarity is greater than T. Thus, the number of documents in the cluster will increase as T is decreased, with the documents in the cluster at some threshold being a subset of those in the cluster at some lower similarity threshold.

Closely related to single linkage clustering is the concept of a *minimal spanning tree*, or MST. Given the set of $N(N-1)/2$ interdocument similarities, a spanning tree is a set of $N-1$ similarities that links all of the N objects in the dataset together into a connected graph without any circuits; the MST is that spanning tree for which the sum of the $N-1$ similarities is a maximum [7]. Gower and Ross showed that the single linkage clusters at some threshold T can be obtained by deleting all coefficients from the MST for which the similarity was less than T [37]. MSTs have been used for the clustering of index terms in studies of query expansion [38,39]; however, they do not seem to have been used for the clustering of documents.

The single linkage method has been described in terms of the characteristics of the clusters which arise from its use. An alternative and popular approach to the description of clustering methods is to use the algorithms that are used for the implementation of the methods. Note that there are often very many different algorithms that can be used for the implementation of the same method: this point is discussed further in Section 2.3.

Single linkage is just one of a group of hierarchic agglomerative clustering methods that have been used extensively over the years and that includes the *complete linkage, group average,* and *Ward* methods *inter alia*. All of these methods may be described by the following general algorithm:

```
FOR I := 1 TO N − 1 DO
  FOR J := I + 1 TO N DO calculate SIM[I,J];
REPEAT
  search SIM to identify the most similar remaining pair of clusters;
  fuse this pair, K and L, to form a new cluster KL;
  update SIM by calculating the similarity between KL and
    each of the remaining clusters
UNTIL there is only a single cluster.
```

The various hierarchic agglomerative methods differ in the definition of similarity that is used for the selection of the most similar pairs of clusters and for the updating of SIM in this algorithm.

The complete linkage, or furthest neighbor, method is the converse of single linkage, since the least similar pair of documents forms the basis for the definition of intercluster similarity; thus, each cluster member is more similar to the least similar document of that cluster than to the least similar document in any other cluster. This definition of cluster membership is very much stricter than that for single linkage, and the large straggly clusters in the latter case are here replaced by large numbers of small tightly bound groupings, a

form of classification that may be just as inappropriate as the extended single linkage clusters in some applications. In graph theoretical terms, complete linkage clustering corresponds to the identification of the maximally complete subgraphs at some threshold similarity.

The group average method results in clusters such that each cluster member has a greater average similarity to the remaining members of that cluster than it does to all members of any other cluster. It thus represents a midpoint between the two extremes represented by single linkage and complete linkage but has been criticized for tending to produce small "nonconformist" clusters of outliers that are unlike every other cluster [41]. There are several other average linkage methods, such as the median and centroid methods, but these suffer from the fact that inversions may arise in the dendrogram representing a hierarchy [2,9,10].

Ward's method joins together those two clusters whose fusion results in the least increase in the sum of the distances from each document to the centroid of its cluster. Ward's method has proved to be an extremely powerful grouping mechanism, and Wishart suggests that it is probably the most generally useful hierarchic procedure [42]. It has, however, been criticized for tending to produce spherical clusters which may not accurately reflect the true shape of the clusters present in the data set. Moreover, it is only defined explicitly when the Euclidean distance is used for the calculation of the interdocument similarities; the use of an association coefficient (e.g., the Dice coefficient) will not result in an exact Ward classification.

2.3 *Document clustering algorithms*

Early applications of cluster analysis generally involved datasets containing only a few tens, or hundreds at most, of objects and thus relatively little attention needed to be given to the development of highly efficient algorithms. Most current hierarchic agglomerative clustering programs are based on the simple, stored matrix algorithm shown previously. An inspection of this shows that SIM contains $O(N^2)$ elements and is searched $N - 1$ times, once for each fusion. The algorithm accordingly has overall time and space requirements of $O(N^3)$ and $O(N^2)$ respectively; Hartigan [4] suggests that the time complexity can be reduced by a "spiral search" procedure, which involves sorting the elements of SIM, but there do not seem to have been any reports of the use of such a technique for document clustering.

The stored matrix algorithm has two useful features. Firstly, the entries in the similarity matrix corresponding to the new cluster KL can overwrite those corresponding to the predecessor clusters K and L; thus, once the initial similarities have been calculated, no additional storage is required during the classification. The second point is that the new entries in the matrix corresponding to the similarities between KL and each of the other clusters may be calculated from the entries for K and L without recourse to the original data matrix using a recurrence formula due to Lance and Williams [40]. However, the time and space complexity means that the algorithm is not suited to document clustering, where datasets may contain thousands, or tens of thousands, of documents.

If large document collections are to be clustered, use needs to be made of the extensive research that has been carried out over the last few years to devise efficient new algorithms for the main hierarchic agglomerative methods. At the same time, the need to calculate large numbers of interdocument similarity coefficients during the generation of a classification has stimulated studies of efficient nearest neighbor searching algorithms. The close relationship between searching and clustering algorithms has been noted by Murtagh [9,10].

The first report of an hierarchic agglomerative clustering method being applied to a substantial document collection is that of Croft [43], who used the single linkage method to cluster 11,613 titles taken from the UKCIS document test collection. The similarities required by the clustering algorithm were generate using an inverted file to the document collection, thus avoiding the need to calculate the many zero-valued interdocument similarities. However, Harding and Willett showed that although this procedure is very efficient when short document descriptions are used (e.g., the titles employed in Croft's experi-

ments), it can result in large numbers of non-zero-valued coefficients being calculated several times over when more extended document descriptions are used (e.g., document abstracts), with a consequent substantial increase in the running time [44]. Willett subsequently described an improved inverted file algorithm for the calculation of interdocument similarity coefficients; this not only avoids the need to make redundant similarity calculations but also requires that each document description be processed once only for the calculation of all of the similarities which involve it [45]. The algorithm, which can also be used for the calculation of interterm similarities [45,46], is derived from the inverted file nearest neighbor searching algorithm described by Noreault *et al.* for calculating query-document similarities in the SIRE information retrieval system [47]. The algorithm has been used extensively in the SMART system for both searching [48] and clustering [49] and is probably the most efficient way of calculating similarities that is currently available [33].

There are very many single linkage algorithms available [50], and this method has proved relatively easy to implement for the clustering of large document classifications. The algorithm used by Croft [43] had originally been devised by van Rijsbergen [51]; it has the advantages that the hierarchy is progressively updated as new similarities become available, regardless of the order in which they are calculated, and that there is no need to store an interdocument similarity matrix. This is an important point since although computational speeds have increased sufficiently to make $O(N^2)$ time algorithms at least feasible, $O(N^2)$ storage requirements are still very difficult to accommodate. Another single linkage algorithm that operates by progressively updating a hierarchy is the SLINK algorithm of Sibson [52]; this has time and storage requirements of $O(N^2)$ and $O(N)$ respectively and has been shown to be optimally efficient. It has been used for large-scale document clustering research by El-Hamdouchi [24]. An important feature of the SLINK algorithm is that it makes use of what Sibson refers to as a *packed representation*; here, each document is described in terms of its position relative to the other documents in the hierarchy and of the similarity level at which it is not the last document in its cluster. Croft has described how this representation can be used to store a complete document hierarchy in a storage-efficient manner for retrieval [53]. An alternative algorithm has been described by Voorhees, who has made use of the fact, mentioned previously in Section 2.2, that single linkage clusters can be identified by applying a similarity threshold to a minimal spanning tree; the generation of minimal spanning trees has been extensively investigated [9,10] and this thus provides an efficient way of generating single linkage classifications. Voorhees' algorithm operates by successively identifying that document not currently linked into the hierarchy which has the greatest similarity to a document that is so linked [49].

As described in Section 5, there has been considerable interest recently in the use of other hierarchic agglomerative methods for document clustering: this has necessitated the development of efficient (i.e., $O(N^2)$ time and $O(N)$ space) algorithms for these methods as well. An algorithm with these complexities for the complete linkage method has been reported by Defays [54], his CLINK algorithm being a modification of Sibson's SLINK algorithm for the single linkage method. El-Hamdouchi has used the CLINK algorithm for generating complete linkage classifications for several large document test collections but has found that the resulting hierarchies give very poor levels of retrieval effectiveness [24]. The reason for this is that the CLINK algorithm does not seem to generate an exact complete linkage hierarchy; Defays hints at this fact in his original paper [54] and Wishart has also noted that the algorithm can give unsatisfactory results in some cases [42]. This would certainly appear to be so in the context of document clustering since El-Hamdouchi also reports experiments using Voorhees' complete linkage algorithm; he states that the classifications resulting from the latter algorithm yield much more satisfactory levels of retrieval effectiveness in cluster searches [24]. The computational requirements of Voorhees' complete linkage algorithm are analogous to those of the simple stored matrix algorithm; however, the large number of zero-valued similarities means that the storage requirement is generally much less than the worst case of $O(N^2)$. Nevertheless, the algorithm is very demanding of storage and involves extended execution times when used with large document collections [24,49]. The algorithm is based on the fact that two complete linkage clusters merge at the level of minimum similarity between any pair of documents, one of

which is in one cluster and one in the other. If the similarities between all pairs of documents are processed in descending similarity order, then two clusters of sizes m and n documents can be merged as soon as all of these $m * n$ similarities have been processed [49].

The heart of the stored matrix algorithm is the repeated searching and updating of the similarity matrix to identify the most similar pair of current clusters. This fact has led to the development of a novel type of clustering algorithm which is based on repeated nearest neighbor searching [9,10,24,55]. Specifically, the algorithm involves the generation of chains of nearest neighbors until a pair of points (i.e., individual documents or clusters of documents) is identified such that one point is the nearest neighbor of the other and *vice versa*. These *reciprocal nearest neighbors* correspond to the clusters in an hierarchic agglomerative classification, and the algorithm thus provides an efficient means of generating such classifications given an appropriate nearest neighbor procedure (such as the one described earlier in this section). El-Hamdouchi [24] and El-Hamdouchi and Willett [55] describe the use of this algorithm for generating Ward classifications of several document test collections. The algorithm requires that the documents comprising a cluster can be represented by a single point (i.e., by a cluster centroid of some sort) and that the clustering method that is to be implemented is based on the similarities between such centroids; it can thus be used for the implementation of Ward's method (and also for the less widely used median and centroid methods). Where this is not the case, other approaches may be more suitable (e.g., El-Hamdouchi reports that the SLINK algorithm is typically about three times as fast at generating single linkage document classifications as the reciprocal nearest neighbors algorithm is at generating Ward classifications of the same document collections [24]).

No $O(N^2)$ time, $O(N)$ space algorithm is known for the group average method since the interpoint similarities here are based on the calculation of all pairwise interdocument similarities such that one document is in one cluster and one document in the other cluster. However, Voorhees has noted that the sum of these interdocument similarities is equal to the intercentroid similarity if a suitable weighting of the centroid elements is used. She makes use of this fact in her group average algorithm, which is a modification of the stored matrix approach that involves the repeated identification of the most similar pair of current clusters but without the storage and updating of the full similarity matrix [49]. The availability of an intercentroid similarity has enabled El-Hamdouchi to adopt Voorhees' weighting scheme to allow the group average method to be implemented using the reciprocal nearest neighbors algorithm [24].

It is thus now at least feasible to implement the main hierarchic agglomerative clustering methods on document collections containing up to a few tens of thousands of documents, given a sufficiently powerful processor. It should, however, be noted that the algorithms described here generally require a very large amount of memory. For example, the reciprocal nearest neighbors algorithm requires that not only the entire document collection that is to be clustered but also the inverted file to it (which is continually updated as new clusters are identified) are held in the main store [55]. The overheads are least for the SLINK algorithm, and this is probably the only one of the algorithms described here that could be implemented on document collections of really substantial size.

Having established that it is possible to generate a range of different types of document classification, the question arises as to whether these have any meaning—a question addressed in the next section.

3. THE VALIDITY OF DOCUMENT CLASSIFICATIONS

Clustering methods are known to produce well-marked classifications of a dataset, even if the data has been generated randomly, and there is thus considerable interest in the development of *cluster validity* techniques. Dubes and Jain identify three main types of cluster validity study in their excellent review of the subject [56]. The first of these seeks to determine whether the clusters in a dataset are significantly different from random, since it is clearly inappropriate to consider the application of a clustering procedure if this is not the case. Once it has been shown that a dataset does indeed exhibit a nonrandom cluster-

ing tendency, tests may be carried out to determine the extent to which the output from the classification procedure recovers the structure of the dataset. The use of these two types of analysis in document clustering research are discussed further in Sections 3.1 and 3.2. Finally, once the overall structure has been confirmed, the validity of individual clusters within a hierarchy or a partition may be of interest; this last type of study is of less interest in document clustering applications where the characteristics of an individual cluster are of less importance than the whole set of clusters (or some subset of them, such as those at the base of a cluster hierarchy or those containing relevant documents).

3.1 *Use of random document graphs*

Ling has described a test that can be used to determine whether document clusterings are nonrandom in character: this test is based on the *random graph hypothesis*, a null hypothesis that is invoked to determine whether any significant predisposition to cluster is present within a dataset [57]. Given a dataset containing N documents, it is assumed that an $N * N$ matrix, SIM, is available in which, as before, the elements SIM[I,J] contain the similarities between each pair of objects I and J. The rank matrix RANK is another $N * N$ symmetric matrix, the elements of which, RANK[I,J], contain the rank orders of the corresponding elements in SIM after they have been sorted into order of decreasing similarity; the upper portion of RANK will accordingly contain all integers in the range 1 to $N(N-1)/2$. The random graph hypothesis is that all $[N(N-1)/2]!$ such matrices are equally likely, and this null hypothesis may be used to study various characteristics of clusterings. One such characteristic is the minimum number of edges, VMIN, at which a random graph on N nodes becomes connected (i.e., when all of the nodes are contained within a single cluster). Ling [57] found an accurate method for the enumeration of all connected graphs on N nodes and V edges, and Ling and Killough [58] used this to calculate tables for the probability of observing specific values of VMIN under the random graph hypothesis given a graph containing N nodes.

The connected subgraphs of a graph may be obtained by setting some threshold similarity T and then linking together all pairs of documents for which SIM[I,J] $\geq T$. As noted previously, the resulting subgraphs correspond to the clusters formed at that threshold by the single linkage clustering method, and Ling and Killough's tables may hence be used by studying single linkage cluster hierarchies. Specifically, a note is made of the rank in RANK at which all of the objects in a dataset become contained within a single cluster; this rank may then be compared with the tables that give values for the probability of observing such a value of VMIN on the assumption that the graph is not a random one. If this probability is greater than some arbitrary threshold value, for which Dubes and Jain suggest 0.99 [56], then evidence exists for the conclusion that RANK is significantly different from a random graph, and that some nonrandom clustering tendency is present in the dataset. Thus, the degree of clustering tendency may be tested for simply by determining whether the observed VMIN is large when compared to the distribution of VMIN under the assumption of a random graph.

Griffiths and Willett [59] and Rowlands [60] used subsets of the Keen and Cranfield document test collections; these subsets each contained 100 randomly selected documents (since the tables of Ling and Killough give critical values for datasets of this size or less). Twenty different subsets were generated and single linkage classifications produced using the Dice coefficient as the interdocument similarity measure; 20 proximity matrices were produced using a random number generator for comparative purposes. All of the 20 matrices for both collections were found to produce VMIN values that were sufficiently large to suggest that the datasets exhibited a nonrandom clustering tendency; conversely, none of the randomly generated matrices exhibited such a tendency. It would thus not seem inappropriate to use cluster analysis methods for the processing of these datasets.

The random graph hypothesis has also been used by Shaw in a recent study that related the statistical significance of partitions of single linkage hierarchies to retrieval effectiveness in cluster searches [61]. The particular graph characteristic studied here was the *component order distribution*, (i.e., the distribution of the sizes of the components in a graph; these components are obtained by applying a similarity threshold to a single link-

age hierarchy so as to obtain a partition). Each and every cluster (i.e. graph component,) in a single linkage hierarchy for 250 master's degree papers in library science was considered as the retrieved set of documents for each of 22 queries, and that cluster chosen which gave the best retrieval using the chosen measure of retrieval effectiveness, which was the van Rijsbergen E measure [12]. Different partitions, and hence sets of components, were obtained from the single linkage hierarchy by applying different similarity thresholds; different single linkage hierarchies were generated by altering the exhaustivity and specificity of the document descriptions. The E measure was then used to identify the best combination of similarity threshold and indexing strategy for cluster-based retrieval; these were then compared with a statistic representing the component order distribution. The results showed that variations in optimal retrieval effectiveness were mirrored by those in the test statistic, this suggesting that it might be possible to specify the best partition for retrieval purposes solely on the basis of characteristics of the document hierarchy. The work represents a contribution to the literature on partitioning hierarchic classifications [62], which is, in turn, related to the more general problem of specifying the number of clusters present in a dataset; the importance of studies in this are has been noted by Everitt [63].

3.2 *Use of distortion measures*

The second type of cluster validity study considers the extent to which the hierarchic structure resulting from a clustering method reflects the similarity relationships in the original dataset. Such studies usually involve a *distortion measure* (i.e., a quantitative measure of the extent to which the operation of the clustering method results in a modification of the interobject similarities in the original similarity matrix). Distortion measures have been suggested as a basis for the comparison of different clusterings of the same dataset since a clustering method that imposes a small degree of distortion on the similarity matrix may be thought of as identifying more natural clusters than one that results in a cluster structure that is not actually present in the matrix.

Quantitative measures of distortion may be obtained by comparing the set of interdocument similarities in an input similarity matrix with the corresponding set of similarities as defined by the output dendrogram. Reviews of such measures are given by Cormack [1] and by Rohlf [64]; the most well-known examples are the cophenetic correlation coefficient [65] and the family of distortion measures described by Jardine and Sibson [66]. Both of these measures were used by Griffiths and Willett [59] in their studies of the Keen, Cranfield, and Evans document collections; in common with other, nonbibliographic studies, they concluded that the group average and single linkage methods resulted in the lowest degrees of distortion. However, it should not be assumed that these methods will accordingly result in the most effective classifications; indeed, Clifford and Stephenson [34] and Williams and Clifford [67] have suggested that distortion of the similarity matrix is not necessarily to be avoided and that a clustering method should attempt to identify groupings that are more intense than those present in the similarity matrix. These remarks seem to be particularly apposite in the information retrieval context, where small, tightly bound clusters of documents seem to give the best retrieval results (discussed as follows).

3.3 *Tests of clustering tendency*

Although it does not seem to fit readily into Dubes and Jain's three-part typology, there is a further type of validity study that is of particular importance in the context of document clustering. It is known that different document test collections react in differing ways to information retrieval techniques [68] and it is thus increasingly common to use a range of test collections in experimental information retrieval research. Document clustering is carried out to improve retrieval effectiveness and is a computationally demanding task. It would thus be exceedingly useful if there was some simple measure of clustering tendency available that could determine whether a particular document collection was likely to respond well to the application of a clustering method (i.e., if the resulting classification was likely to give search performance superior to that of a conventional, unclustered database).

Van Rijsbergen and his co-workers [12,18,69] have made extensive use of the *cluster hypothesis test*. This states that documents which are similar to each other may be expected to be relevant to the same requests; dissimilar documents, conversely, are unlikely to be relevant to the same requests. The hypothesis may be tested by calculating all of the relevant-relevant (RR) and relevant-nonrelevant (RNR) interdocument similarities for a given query; if the hypothesis is correct, the average RR coefficient will be larger than the average RNR coefficient. Alternatively, the coefficients may be summed over a set of queries and the results displayed by a pair of relative frequency histograms; a quantitative measure of the extent to which the RR and RNR distributions overlap has been described by Griffiths *et al.* [70]. A second, *nearest neighbor* test has been described by Voorhees [71]. This takes each of the relevant documents for some query in turn and then identifies how many of its nearest neighbors are also relevant to the query; in her experiments, Voorhees calculated the percentage of the relevant documents in a collection that had 0, 1, 2, 3, 4, or 5 relevant documents in their sets of 5 nearest neighbors. El-Hamdouchi and Willett have recently proposed a *term density test* [72]; here, the degree of clustering tendency in a collection is estimated by the total number of postings in the database divided by the product of the number of documents in the collection and the number of terms that have been used for indexing these documents. The basis for this idea is that a database where each document has only a few terms selected from a large indexing vocabulary will be one in which the great bulk of the pairs of documents will have no terms in common when the interdocument similarities are calculated in the course of some clustering procedure. When the data matrix is more densely populated, conversely, documents can share many terms in common and it will be possible to differentiate between documents that are very similar to each other and those that are less closely related; the resulting classification can thus display more accurately the interdocument relationships than can one where the range of possible interdocument similarities is very limited. These three measures of clustering tendency were used to rank seven document collections in order of decreasing clustering tendency, and it was found that the methods produced rather different rankings; of these, that resulting from the term density test was found to correlate best with the effectiveness of cluster searching. Thus, a high term density for a collection would seem to suggest that a clustering method might usefully be applied to the collection for retrieval purposes; a low density, on the other hand, suggests that clustering is unlikely to produce useful results [72].

The mention of retrieval effectiveness implies the availability of strategies for searching clustered document files; such strategies are discussed in the following section.

4. SEARCHING HIERARCHIC DOCUMENT CLASSIFICATIONS

Two primary sorts of search strategy have been described for the matching of a query (i.e., a set of index terms), against a file of documents: Boolean searching, where the query terms are linked by the Boolean logical operators of AND, OR and NOT and where those documents are retrieved that satisfy the logical constraints in the query, and best match searching, where the documents are retrieved in order of decreasing similarity with the query. Although Boolean search strategies have been described for hierarchical cluster searching [73,74], most of the research work in this area has considered best match retrieval algorithms; however, there are many ways in which such a search can be carried out. Specifically, decisions must be made as to how the hierarchy is to be used to guide the progress of the search, how the documents comprising a cluster are to be represented for matching against queries, and what sort of retrieval criterion is used to control the output produced in response to a query.

4.1 *Search strategies*

Given a hierarchy, there are three main ways in which a query can be matched against the clusters in the tree structure. A *top-down* search is one in which the query enters the tree via the root; it is then matched against the two child clusters and that downward path in the tree chosen for which the query-cluster similarity is the greater. The search then

moves down the tree until some retrieval criterion is satisfied. Two obvious criteria present themselves. The first of these involves the user specifying some minimal number of documents that must be produced by the search, in which case it continues until a sufficiently small cluster of documents is identified. Alternatively, the query-cluster similarities are noted at each stage of the downward scan of the tree, and the search terminated when these similarity values start to fall. In both cases, the output from the search is a single cluster of documents.

Such a search is intuitively appealing, and is superficially similar to conventional tree searching algorithms [75,76] in that it requires $O(\log N)$ matching operations to produce the search output. The main difficulty associated with such a search is that the first few query-cluster matches, which are carried out near to the root of the tree, involve clusters representing a significant fraction of the entire database. When many hundreds or thousands of documents are contained in a cluster, it is not possible to obtain a satisfactory representation of the documents in that cluster; thus, the first few choices in a top-down search are almost arbitrary in character if the database is of nontrivial size. It is, accordingly, very easy to miss those parts of the hierarchy that contain the relevant material for the query (if these documents have, in fact, been grouped together by the clustering method). Top-down searching is thus likely to be practicable only if some thresholding procedure is applied to the hierarchy to obtain clusters no larger than some threshold size which, it is felt, can be represented adequately. Once such a partition has been obtained, the search commences with a best match scan of the clusters in the partition; the best matching cluster is then used as the starting point for the search.

Many of the limitations of a top-down search are eliminated by the use of a *bottom-up* search strategy which is, as its name suggests, the inverse of the previous retrieval strategy since the search here commences at a document or cluster at the base of the tree and then moves up it towards the root until the retrieval criterion is satisfied. The main problem associated with bottom-up searching is choosing the starting point for the search. The simplest approach, which was used by van Rijsbergen and Croft in their initial work on bottom-up searching [77], is to assume that a single relevant document is already available; this is often the case in a practical retrieval environment. Alternatively, if such a document is not available, a conventional, nonclustered best match search can be carried out to identify that document that is most similar to the query; this document is then chosen as the starting point for the bottom-up search [19]. Rather than using an individual document, a third approach involves a best match scan of the *bottom level clusters* to identify that one that is most similar to the query; this cluster, rather than an individual document, then acts as the point of departure for the bottom-up search. Each of the documents in a collection is associated with one bottom-level cluster, this being the *smallest* cluster that contains it (i.e., the initial cluster for that document when it first becomes linked into the hierarchy). It will be realized that there can be a considerable degree of overlap in the constitution of the bottom level clusters; thus, a pair of documents, A and B, which fuse together have identical bottom-level clusters, and this will be a subset of the bottom-level cluster for some document, C, which joins the hierarchy by fusing with the cluster containing A and B. Given a starting point, the bottom-up search strategy identifies the relevant document, best matching document, or best matching bottom-level cluster at the base of the tree. The search then moves upwards until the retrieval criterion is achieved.

Van Rijsbergen and Croft [77] and Croft [19] have carried out comparative studies of top-down and bottom-up searches of the Cranfield-1400 collection and concluded that the latter strategy gave better levels of retrieval effectiveness. In some of his experiments, Croft used the bottom-level clusters by themselves for the bottom-up search, without consideration of the rest of the cluster hierarchy, so that the search involves just a scan of the bottom-level clusters to identify that which is most similar to the query. His results suggest that this strategy, which avoids the need to inspect any of the hierarchy above the bottom-level clusters, seems to give the best results of all of the top-down and bottom-up strategies which were tested [19]. This finding has been confirmed in more extensive studies by El-Hamdouchi and Willett [78].

Top-down and bottom-up searches usually result in the retrieval of a single cluster:

Jardine and van Rijsbergen refer to this as *cluster-based retrieval* [18]. Cluster-based retrieval has the limitation that all of the documents in the cluster which is retrieved are considered to be of equal relevance to the query and the output is thus not ranked in order of decreasing probability of relevance to the query. This situation, which is analogous to that encountered in Boolean retrieval systems, means that the searcher may not have control over the precise volume of output produced unless further processing of the output is undertaken. Thus, van Rijsbergen and Croft have suggested that if a very large cluster results from a bottom-up search, it should act as the starting point for a top-down search. This problem is avoided, of course, if a threshold cluster size is used as the retrieval criterion.

Rather than retrieving too much output, it should be noted that most early studies of cluster-based retrieval [18,19,77,80,81] involved the retrieval of just a single cluster; if this happens to be a small one, near the bottom of the hierarchy, then only two or three documents will result from a search and this is clearly likely to be insufficient for many practical purposes. More recent studies have, accordingly, retrieved more clusters; Griffiths *et al.* [70] report experiments, using just the bottom-level clusters, in which either the 5 top-ranking clusters are retrieved or sufficient top-ranking clusters to retrieve 10 distinct documents. A further modification, which Voorhees has suggested gives better results, is to take some number of clusters and then to individually rank the documents in these clusters against the query [79].

4.2 *Matching of queries and clusters*

Several of the search strategies described in the previous section have involved the matching of a query against the clusters to identify those that are most similar to it. The similarity between a query and an individual document is calculated by means of some matching function based on the terms in common between them and thus, if analogous query-cluster similarities are needed, some sort of document-like *representative* is required summarising the index term characteristics of the documents in a cluster. There are many possible ways in which this can be achieved. Thus, van Rijsbergen and Croft add together the term vectors describing each of the documents in a cluster, rank the terms in decreasing order of frequency of occurrence in the cluster, and assign the resulting ranks as the weight for each term in the representative; low weight terms are then deleted by a thresholding procedure [77]. Alternatively, the actual cluster frequencies can be used as the weights, rather than the ranks derived from them [19], or the deletion strategy can be omitted so that all of the terms in a cluster can be included in the representative [70]. Both of these types of representative result in weighted representatives (i.e., ones in which each of the terms is accompanied by a measure of its relative importance in the cluster). Alternatively, a binary representative can be used, this generally being obtained by restricting membership of it to those terms that occur in some threshold number of documents in a cluster (e.g., Jardine and van Rijsbergen describe representatives in which a term is included in a representative containing n documents if that term occurs in more than either 1 of them or $log_2 n$ of them [18]). Detailed studies of cluster representatives are described by Voorhees [79], Croft [80], and Murray [82] in their doctoral theses.

Although a cluster representative can give a reasonable description of the individual documents in a cluster when that cluster is small, the whole idea of a representative becomes less appropriate when large clusters are considered. These inevitably correspond to more heterogeneous bodies of documents than do small clusters and thus the size of the representative (i.e., the number of terms contained within it will become comparable with the size of the indexing vocabulary that has been used for the characterisation of the complete document collection. If this is not to happen, rather drastic term deletion strategies must be adopted to constrain the size of the representative; however, this in turn will then mean that the representative does not describe accurately the terms present in the cluster. Apart from the very substantial storage overheads required for the representatives of large clusters, there is also the problem that individual documents within a large cluster are less like their representative than in a small cluster, and the representative is accordingly less likely to be able to direct a query towards those few documents that are relevant to that

query. This is particularly important in the context of top-down searches where, as has been noted above, the initial query-cluster matches involve clusters containing significant fractions of the entire database that has been clustered. Croft presents a probabilistic analysis of top-down and bottom-up searches and shows that the latter will be expected to give more effective retrieval, a conclusion supported by his experimental results [19]. Confirmatory evidence to support the view that small clusters are the best for retrieval is provided by Griffiths *et al.* who show that clustering methods that result in large numbers of very small bottom-level clusters tend to give better results than methods that can give rise to large bottom-level clusters [70,81].

Having suggested that bottom-up searching will generally give better results than top-down searching especially when the very small bottom-level clusters are used, it is rather surprising to find that Voorhees has concluded that top-down searches are to be preferred [79]. However, this conclusion is based on experiments with the complete linkage method, which results in very small clusters, with the great bulk of the intercluster fusions taking place at the level of zero similarity, so that even the largest complete linkage clusters, which are those inspected first in a top-down search, are likely to be quite small and to contain only a few documents.

Once a set of cluster representatives has been obtained, a matching function is calculated between the query and each of them to calculate the query-cluster similarities. Most experiments have used similarity measures like the cosine coefficient, but there has been some interest in the use of probabilistic matching functions. Thus, Yu *et al.* have described a model that involves estimating the number of documents in a cluster that contain a user-defined number of terms in common with the query [83,84]. Salton and Wong used this model to select those clusters which were expected to have the largest numbers of documents with many terms in common with the query [85]. Kar and White [86] and Hamill and Zamora [87] have described probabilistic methods for classifying a document into a fixed set of clusters; Croft has developed their methods so that they can be used for the related problem of classifying a query into one or more of a set of clusters to identify those that should be retrieved [19]. Croft suggests that his model and that of Salton and Wong both tend to retrieve the same clusters, despite their very different theoretical bases.

5. THE EFFECTIVENESS OF CLUSTER SEARCHES

As has been noted in the Introduction, the main rationale for the use of hierarchic document clustering methods is that they may be able to increase the effectiveness of document retrieval systems; this section reviews the experiments that have been carried out to test this rationale.

5.1 *Use of the single linkage method*

Jardine and van Rijsbergen were the first to suggest that cluster-based retrieval could give improved performance when compared with a conventional best match search in which the individual documents are ranked in order of decreasing similarity with a query [18]. They based this suggestion, in large part, on *optimal cluster searches*, where the optimal cluster for a particular query is that cluster which, if retrieved, would give the maximum possible value of the chosen effectiveness criterion, the E value in the case of Jardine and van Rijsbergen's experiments. Optimal cluster searching thus represents an upperbound to the performance of a retrieval system based on clusters: whether this upperbound can be achieved in practice will depend on the effectiveness of the search strategy that is used. Jardine and van Rijsbergen's experiments used the Cranfield-200 collection; their results suggested that cluster-based retrieval had the potential to greatly exceed nonclustered best match searching and similar results were obtained in later tests using the full Cranfield collection [77]. Actual cluster searches using a range of top-down and bottom-up strategies were noticeably inferior to the optimal searches but were sufficiently effective to encourage the belief that cluster based retrieval could yield useful results in practice [18,77,88]. This belief was further supported by Croft's later work with the Cranfield-1400 collection;

this showed cluster-based retrieval, using a bottom-up search commencing at the best-matching bottom-level clusters in the hierarchy, outperforming noncluster searches [19,80].

In retrospect, there are three possible problems with these studies. First, the concentration on the use of the Cranfield collection, which seems to respond very much better to the use of clustering techniques than other document test collections [68–70,72]. Second, many of the tests involved the retrieval of only a single cluster that typically contained only two or three documents, a situation far removed from that of a conventional document retrieval system. Last, the work involved only a single clustering method: single linkage.

5.2 *Use of other methods*

The use of the single-linkage method was appropriate in the light of Jardine and Sibson's theoretical analyses of clustering methods [66], which demonstrated several theoretical characteristics of this method that make it uniquely well suited to cluster analysis. However, many comparative studies of hierarchic agglomerative methods, using both simulated and real datasets, have shown that the single-linkage method generally gives results that are far inferior to those obtainable when the other hierarchic agglomerative methods are used [62,63,89–97]. Accordingly, Griffiths *et al.* suggested that further increases in performance might be obtained if these other methods were used for document clustering [81]; to test this, they carried out both optimal and actual cluster-based retrieval searches on single linkage, complete linkage, group average and Ward classifications of the Keen and Cranfield collections (this being subdivided into two halves to allow the use of the clustering routines in the CLUSTAN package [42,59]). Their results showed that optimal searches of the single linkage hierarchies were significantly poorer than those of the other three methods, although the difference was noticeably less when actual searches were carried out; of the four methods tested, group average gave the best results [81]. Griffiths *et al.* also analyzed the structures of the dendrograms resulting from the fourth method, using two coefficients of hierarchic structure due to Murtagh [98,99], and showed that while the single linkage method resulted in the well-known, highly chained clusters, the other three methods all produced a much greater number of small, tightly defined clusters. Further cluster searches, using real retrieval strategies, on the Keen and Cranfield datasets and on a subset of the Evans collection are reported by Griffiths *et al.* [70]. In these experiments, several clusters were retrieved, rather than just one as in previous cluster-based retrieval experiments; Ward's method was found to give the best overall results, then complete linkage and group average, and with single linkage being consistently the worst. These experiments, as well as most of the previous ones [81], involved scanning just the bottom-level clusters and Griffiths *et al.* reported a detailed analysis of the constitution and the size of these clusters in the four types of hierarchy [70]; this analysis further emphasized the great similarities between the complete linkage, group average, and Ward classifications and the very disparate nature of the single linkage classifications.

The work of Griffiths *et al.* [70,81] was restricted to collections containing 800 documents or less and they emphasized the need to extend their comparative studies to larger datasets. The development of the fast clustering and nearest neighbor searching procedures described in Section 2.3 has now allowed such large-scale tests of effectiveness to be carried out. Voorhees used a range of search strategies to compare single linkage, complete linkage, and group average classifications of collections containing between 1033 and 12684 documents [79]. She concluded that the complete linkage method was to be preferred to group average, with single linkage again performing least well. In addition to comparing cluster searches in the three types of hierarchy, Voorhees also compared the results with those obtained in noncluster searches, these being based on the p-norm Boolean model [100]. She concluded that the top-down complete linkage searches were marginally better than the noncluster searches if the cluster search first ranked the clusters and then ranked the documents within the highest-ranking clusters. She also compared the efficiencies of cluster and noncluster searching in terms of the number of similarity calculations and disc accesses required and found that the cluster searches were much more demanding of com-

puter resources unless an extremely complex query-document matching function was used in the noncluster search [101].

More recently, El-Hamdouchi [24] and El-Hamdouchi and Willett [78] have used the same four methods as Griffiths *et al.* to cluster seven document collections containing between 800 and 27,361 documents (the UKCIS test collection). Four different bottom-up search strategies were used; for all of the collections, the best results were obtained from ranking the bottom-level clusters in order of decreasing similarity and then applying a cut-off to retrieve the required number of documents (as originally advocated by Croft [19]). Despite the substantial differences in the characteristics of the collections (e.g., indexing exhaustivity, term density, and number of relevant documents), the order of decreasing performance was almost always found to be group average, Ward's method, single linkage, and complete linkage. They also compared cluster searches with noncluster searches; with the exception of the Cranfield collection, the latter were generally to be preferred. An explanation of this behavior is presented by El-Hamdouchi [24].

While agreeing on the poor performance of single linkage, these comparative studies of hierarchic document clustering seem to give extremely divergent results; however, it is felt that this difference is more apparent than real for the following reasons. First, Ward's method (suggested by Griffiths *et al.* as the best method in their later experiments) is explicitly defined only when a distance measure, rather than a similarity coefficient, is used for the calculation of the interdocument similarities. Unfortunately, most of the experiments carried out by Griffiths *et al.* [70,81] were based on the use of the Dice coefficient and the method described by them as Ward's method was not; in fact, some of the characteristics of this (unnamed) method are discussed by El-Hamdouchi [24]. Second, the complete linkage clusters used in El-Hamdouchi and Willett's experiments were those produced by the CLINK algorithm of Defays [54]; as has been noted in Section 2.3, this has been shown to be suboptimal in operation [24]. When Voorhees' exact, but extremely demanding, complete linkage algorithm was used on the three smallest collections available to them, El-Hamdouchi and Willett found that this method did indeed perform best (in accordance with Voorhees' results). Accordingly, this reviewer believes that the complete linkage method is probably the most effective of the common hierarchic agglomerative methods when used for document clustering. Of the various search strategies that have been used, the top-down strategy works well with complete linkage owing to the large number of very small clusters even at the topmost level of the hierarchy; with the other methods, ranking the bottom-level clusters seems to give the best results. In view of the fact that the complete linkage method is the most effective of the methods, it is unfortunate that it also requires the greatest computational resources to generate the clusters.

6. RELATED WORK AND CONCLUSIONS

The previous sections have described much of the recent work that has been carried out into hierarchic document clustering. This closing section briefly outlines some related areas of study and makes suggestions for further work.

As noted in Section 1, the main use of classification, as against cluster analysis, in library and information systems is for the organization of library collections using the standard bibliographical classification schemes. These schemes are used to assign classification codes manually, using human assessments of the subject area of a bibliographic item and using the classification schedules of the chosen scheme. There has recently been some interest in carrying out book classification automatically using keyword descriptors as the basis for the classification. Garland clustered a set of 416 monographs represented by Library of Congress subject headings and title keywords. She used the single linkage method and compared the resulting clusters with the Library of Congress classes [102]. A much more extensive study is reported by Enser, who used monograph titles, tables of contents and back-of-the-book indexes as the basis for keyword and n-gram substring characterizations of 250 books classified using the Dewey Decimal Classification [103]. The various representations of content were used as the basis for clustering experiments with

the group average and Ward hierarchic methods and with a nonhierarchic, relocation method [30]. The resulting clusters were then evaluated by simulated searches and by comparison with the Dewey classmarks. Enser concluded that at least some of the automatic classifications seemed to be superior to the conventional library classifications [103].

The implementation of clustering on a large scale has been discussed in Section 2.3 above; one way in which the computational requirements can be reduced is by applying the clustering operations only to a subset of the database that has been identified by some previous search operation. Examples of the use of such an approach with the single-linkage method have been discussed by Becker [104], who reported a simulation of clustering the output from Boolean searches, and by Willett [105], who clustered highly ranked documents from coordination level searches of three standard test collections; the latter's results suggest that the effectiveness of cluster searches of the resulting classifications are less effective than comparable searches when the entire collection is clustered. Despite this finding, clustering the output of a recall-oriented search does provide a simple means for the searcher to gain a feeling for the range and classes of documents in the retrieved set; a successful application of this approach has been reported in the context of chemical substructure retrieval [106].

In their comparative studies of the characteristics of bottom-level clusters, Griffiths *et al.* noted that the methods which gave good retrieval results all yielded very small bottom-level clusters, often containing just a pair of highly similar documents. They accordingly suggested that acceptable levels of retrieval effectiveness could be obtained by using a very simple, overlapping classification containing just the *nearest neighbor clusters* for a collection, where a nearest neighbor cluster consists of a document and that document that is most similar to it. Such a classification can be generated at a reasonable computational cost using the fast similarity algorithms described in Section 2.3, and the resulting clusters can also be searched very efficiently using a conventional inverted file [70]. Griffiths *et al.* demonstrated that these clusters gave a search performance comparable to that of noncluster searches across a wide range of test collections; in some cases, the cluster search was shown to be significantly superior to the noncluster search [70]. Further studies are reported by El-Hamdouchi [24] and El-Hamdouchi and Willett [78] while Wade and Willett describe a practical implementation of retrieval based on nearest neighbor clusters [107].

The utility of nearest neighbor clusters has also been advocated by Croft as a result of his extensive studies of single linkage clustering [19,77,80]. The detailed results reported in his doctoral thesis provide both a theoretical and a practical basis for using the smallest clusters in a hierarchy; these are typically the bottom-level clusters (with the exception of those documents that only join the hierarchy at a very low similarity level). Taking this as a starting point, Croft has advocated the use of a network model for document retrieval [108,109]. In this, the nodes are documents and terms, and the linkages are either between documents and terms, corresponding to the linkages implicit in conventional serial and inverted files, or between pairs of documents (or pairs of terms), corresponding to documents (or terms) and their nearest neighbors. Such an organization allows the implementation of clustered, Boolean, and nearest neighbor search strategies and is also hospitable to the inclusion of additional information if this available (e.g., citation data or semantic, rather than statistical, linkages). The network organization also allows relatively efficient updating as the file changes (as does the nearest neighbor cluster model); with one or two exceptions [80,110], this important characteristic of an information retrieval system has been little studied in the context of document classifications.

The network organization seems an attractive model for future work on exploiting the similarities between documents and has clear affinities with the associative retrieval strategies that have been advocated by several workers (e.g., Goffman's indirect retrieval method [111,112], the tree search procedure of Mansur [113], and the THOMAS and OAQS systems of Oddy [114] and Preece [115] (which foreshadow many of the features of the network organization)). The availability of several different retrieval strategies in this file organization raises the question as to which strategy should be used for an individual query, or whether some combination needs to be used to obtain the best results. Initial

work on the automatic selection of strategies, including the choice between cluster and noncluster searches, has been inconclusive [116,117] and further studies are clearly needed. A further way of improving performance might be to take the similarities between terms into account in deciding which clusters to retrieve. Probabilistic models of cluster searching [19,85] make explicit assumptions of term independence and although such (unrealistic) assumptions have been shown to give reasonable results when queries are matched against individual documents [21,118], it seems most inappropriate to use them for the searching of clusters that have been generated on the basis of terms common to sets of documents. Some initial attempts to take term dependencies into account during cluster searching have been made by Wong [119] and this work should be extended to see whether substantive performance improvements can be achieved.

It is time now to summarize the main conclusions of this review, which has focused particularly on the efficiency with which various sorts of classification can be generated and on the effectiveness of searches of the resulting clusters. Early work used the single linkage method, which has several attractive theoretical characteristics that mark it out from the other hierarchic agglomerative methods. These experiments showed that hierarchic document clustering could increase the effectiveness of retrieval, in comparison with noncluster search strategies, but that cluster generation can be extremely demanding of computational resources. Later work showed that it was possible to scale up the single-linkage method to handle databases of nontrivial size. More recently, it has been shown that other hierarchic agglomerative methods are superior to the single-linkage method in terms of retrieval effectiveness and that these methods also can be implemented in a relatively efficient manner; unfortunately, the most effective of these methods, the complete-linkage method, seems to be the most demanding of computational resources. The most cost-effective way currently available for the incorporation of interdocument similarity information in a document retrieval system seems to involve considering just pairs of documents that are nearest neighbors; these pairs can usefully be represented in a network.

Acknowledgments—I would like to thank Abdelmoula El-Hamdouchi, Alan Griffiths, Clair Luckhurst, Fiona McCall, Edie Rasmussen, Ley Robinson, and Ian Rowlands for their contributions to the research into document clustering, which has been carried out in Sheffield over the last eight years. Funding for this work was provided by the British Council, the British Library Research and Development Department, the Department of Education and Science, and the Manpower Services Commission.

REFERENCES

1. Cormack, R.M. A review of classification. Journal of the Royal Statistical Society 134:321–367; 1971.
2. Sneath, P.H.A.; Sokal, R.R. Numerical Taxonomy. San Francisco: Freeman; 1973.
3. Everitt, B.S. Cluster Analysis. London: Heinemann; 1974.
4. Hartigan, J.A. Clustering Algorithms. New York: Wiley; 1975.
5. Dubes, R.; and Jain, A.K. Clustering methodologies in exploratory data analysis. Advances in Computers 19:113–228; 1980.
6. Spath, H. Cluster Analysis Algorithms. Chichester: Ellis Horwood; 1980.
7. Lee, R.C.T. Clustering analysis and its applications. Advances in Information Systems Science 8:169–292; 1981.
8. Gordon, A.D. Classification. London: Chapman and Hall; 1981.
9. Murtagh, F. A survey of recent advances in hierarchical clustering algorithms. Computer Journal 26:354–359; 1983.
10. Murtagh, F. Complexities of hierarchic clustering algorithms: state of the art. Computational Statistics Quarterly 1:101–114; 1984.
11. van Rijsbergen, C.J. Automatic classification in information retrieval. Drexel Library Quarterly 14:75–89; 1978.
12. van Rijsbergen, C.J. Information Retrieval. London: Butterworths; 1979.
13. Sparck Jones, K. Automatic Keyword Classification. London: Butterworths; 1971.
14. Minker, J.; Wilson, G.A.; Zimmerman, B.H. An evaluation of query expansion by the addition of clustered terms for a document retrieval system. Information Storage and Retrieval 8:329–348; 1972.
15. Smeaton, A.F.; van Rijsbergen, C.J. The retrieval effects of query expansion on a feedback document retrieval system. Computer Journal 26:239–246; 1983.
16. Yu, C.T.; Buckley, D.; Salton, G. A generalized term dependency model in information retrieval. Information Technology: Research and Development 2:129–154; 1983.
17. Salton, G. The SMART Retrieval System. Englewood Cliffs, NJ: Prentice-Hall; 1971.
18. Jardine, N.; van Rijsbergen, C.J. The use of hierarchical clustering in information retrieval. Information Storage and Retrieval 7:217–240; 1971.

19. Croft, W.B. A model of cluster searching based on classification. Information Systems 5:189–195; 1980.
20. Small, H.; Sweeney, E. Clustering the Science Citation Index using cocitations. Scientometrics 7:391–409; 1985.
21. Croft, W.B.; Harper, D.J. Using probabilistic models of document retrieval without relevance information. Journal of Documentation 35:285–295; 1979.
22. Salton, G.; McGill, M.J. Introduction To Modern Information Retrieval. New York: McGraw-Hill; 1983.
23. Willett, P. Similarity coefficients and weighting functions for automatic document classification: an empirical comparison. International Classification 10:138–142; 1983.
24. El-Hamdouchi, A. The Use of Inter-Document Relationships in Information Retrieval. PhD thesis, University of Sheffield, England; 1987.
25. Cheetham, A.H.; Hazel, J.E. Binary (presence-absence) similarity coefficients. Journal of Paleontology 43:1130–1136; 1969.
26. Green, P.E.; Rao, V.R. A note on proximity measures and cluster analysis. Journal of Marketing Research 6:359–364; 1969.
27. Adamson, G.W.; Bush, J.A. A comparison of the performance of some similarity and dissimilarity measures in the automatic classification of chemical structures. Journal of Chemical Information and Computer Sciences 15:55–58; 1975.
28. Duran, B.S; Odell, P.L. Cluster Analysis: A Survey. Berlin: Springer-Verlag; 1974.
29. Aldenderfer, M.S. A Consumer Report On Cluster Analysis Software. Philadelphia: Pennsylvania State University; 1977.
30. Willett, P. Document clustering using an inverted file approach. Journal of Information Science 2:223–231; 1980.
31. Can, F.; Ozkarahan, E.A. Two partitioning-type clustering algorithms. Journal of the American Society for Information Science 35:268–276; 1984.
32. Fritsche, M. Automatic Clustering Techniques In Information Retrieval. Luxembourg: Commission of the European Communities; 1974.
33. Perry, S.A.; Willett, P. A review of the use of inverted files for best match searching in information retrieval systems. Journal of Information Science 6:59–66; 1983.
34. Clifford, H.T.; Stephenson, W. An Introduction To Numerical Classification. New York: Academic Press; 1975.
35. Wishart, D. Mode analysis, a generalization of nearest neighbour which reduces chaining, in Numerical Taxonomy. (Edited by A.J. Cole.) London: Academic Press; 1969.
36. Jarvis, R.A.; Patrick, E.A. Clustering using a similarity measure based on shared nearest neighbours. IEEE Transactions on Computers C-22:1025–1034; 1973.
37. Gower, J.C.; Ross, G.J.S. Minimum spanning trees and single linkage cluster analysis. Applied Statistics 18:54–64; 1969.
38. van Rijsbergen, C.J. A theoretical basis for the use of co-occurrence data in information retrieval. Journal of Documentation 33:106–119; 1977.
39. van Rijsbergen, C.J.; Harper, D.J.; Porter, M.F. The selection of good search terms. Information Processing & Management 17:77–91; 1981.
40. Lance, G.N.; Williams, W.T. A general theory of classificatory sorting strategies. I. Hierarchical systems. Computer Journal 9:373–380; 1967.
41. Edelbrock, C. Mixture model tests of hierarchical clustering algorithms: the problem of classifying everybody. Multivariate Behavioural Research 14:367–384; 1979.
42. Wishart, D. CLUSTAN 1C User Manual. Edinburgh: Edinburgh University Program Library Unit; 1978.
43. Croft, W.B. Clustering large files of documents using the single-link method. Journal of the American Society for Information Science 28:341–344; 1977.
44. Harding, A.F.; Willett, P. Indexing exhaustivity and the computation of similarity matrices. Journal of the American Society for Information Science 31:298–300; 1980.
45. Willett, P. A fast procedure for the calculation of similarity coefficients in automatic classification. Information Processing & Management 17:53–60; 1981.
46. Noreault, T.; Chatham, R. A procedure for the estimation of term similarity coefficients. Information Technology: Research and Development 1:189–196; 1982.
47. Noreault, T.; Koll, M.; McGill, M.J. Automatic ranked output from Boolean searches in SIRE. Journal of the American Society for Information Science 28:333–339; 1977.
48. Buckley, C.; Lewit, A.F. Optimization of inverted vector searches. Proceedings of the Eighth International Conference on Research and Development in Information Retrieval. 97–110; 1985.
49. Voorhees, E.M. Implementing agglomerative hierarchic clustering algorithms for use in document retrieval. Information Processing & Management 22:465–476; 1986.
50. Rohlf, F.J. Single-link clustering algorithms. Handbook of Statistics 2:267–284; 1982.
51. van Rijsbergen, C.J. An algorithm for information structuring and retrieval. Computer Journal 14:407–412; 1971.
52. Sibson, R. SLINK: an optimally efficient algorithm for the single link cluster method. Computer Journal 16:30–34; 1973.
53. Croft, W.B. A file organization for cluster-based retrieval. Proceedings of the First International Conference on Research and Development in Information Retrieval. 65–82; 1978.
54. Defays, D. An efficient algorithm for a complete link method. Computer Journal 20:93–95; 1977.
55. El-Hamdouchi, A.; Willett, P. Hierarchic document clustering using Ward's method. Proceedings of the Ninth International Conference on Research and Development in Information Retrieval. 149–156; 1986.
56. Dubes, R.; Jain, A.K. Validity studies in clustering methodologies. Pattern Recognition 11:235–254; 1979.
57. Ling, R.F. An exact probability distribution on the connectivity of random graphs. Journal of Mathematical Psychology 12:90–98; 1975.
58. Ling, R.F.; Killough, G.G. Probability tables for cluster analysis based on a theory of random graphs. Journal of the American Statistical Association 71:293–300; 1976.

59. Griffiths, A.; Willett, P. Evaluation of Clustering Methods for Automatic Document Classification. London: British Library Research and Development Department; 1984.
60. Rowlands, I. Clustering Tendency in Document Test Collections: A Preliminary to Cluster Validation in Automatic Document Classification. MSc dissertation, Sheffield University, England; 1983.
61. Shaw, W.M. An investigation of document partitions. Information Processing & Management 22:19-28; 1986.
62. Mojena, R. Hierarchical grouping methods and stopping rules: an evaluation. Computer Journal 20:359-363; 1977.
63. Everitt, B.S. Some unresolved problems in cluster analysis. Biometrics 35:169-181; 1979.
64. Rohlf, F.J. Methods of comparing classifications. Annual Review of Ecology and Systematics 5:101-113; 1974.
65. Sokal, R.R.; Rohlf, F.J. The comparison of dendrograms by objective means. Taxon 11:33-40; 1962.
66. Jardine, N.; Sibson, R. Mathematical Taxonomy. New York: Wiley; 1971.
67. Williams, W.T.; Clifford, H.T. On the comparison of two classifications of the same set of elements. Taxon 20:519-522; 1971.
68. Sparck Jones, K. Collection properties influencing automatic term classification performance. Information Storage and Retrieval 9:499-513; 1973.
69. van Rijsbergen, C.J.; Sparck Jones, K. A test for the separation of relevant and non-relevant documents in experimental retrieval collections. Journal of Documentation 29:251-257; 1973.
70. Griffiths, A.; Luckhurst, H.C.; Willett, P. Using inter-document similarity information in document retrieval systems. Journal of the American Society for Information Science 37:3-11; 1986.
71. Voorhees, E.M. The Cluster Hypothesis Revisited. Technical report TR 85-658: Cornell University, Ithaca, NY; 1985.
72. El-Hamdouchi, A.; Willett, P. Techniques for the measurement of clustering tendency in document retrieval systems. Journal of Information Science 13:361-365; 1987.
73. Litofsky, B. Utility of Automatic Classification Systems for Information Storage and Retrieval. PhD thesis, University of Pennsylvania, Philadelphia, PA; 1969.
74. Croft, W.B. Using Boolean queries with a clustered file organization. Journal of the American Society for Information Science 30:358-360; 1979.
75. Knuth, D.E. Optimum binary search trees. Acta Informatica 1:14-25; 1971.
76. Nievergelt, J. Binary search trees and file organization. Computing Surveys 6:195-207; 1974.
77. van Rijsbergen, C.J.; Croft, W.B. Document clustering: An evaluation of some experiments with the Cranfield 1400 collection. Information Processing & Management 11:171-182; 1975.
78. El-Hamdouchi, A.; Willett, P. Comparison of hierarchic agglomerative clustering methods for document retrieval. Computer Journal (in press).
79. Voorhees, E.M. The Effectiveness and Efficiency of Agglomerative Hierarchic Clustering in Document Retrieval. PhD thesis, Cornell University, Ithaca, NY; 1985.
80. Croft, W.B. Organizing and Searching Large Files of Document Descriptions. PhD thesis, University of Cambridge, England; 1979.
81. Griffiths, A.; Robinson, L.A.; Willett, P. Hierarchic agglomerative clustering methods for automatic document classification. Journal of Documentation 40:175-205; 1984.
82. Murray, D.M. Document Retrieval Based on Clustered Files. PhD thesis, Cornell University, Ithaca, NY; 1972.
83. Yu, C.T.; Luk, W.S. Analysis of effectiveness of retrieval in clustered files. Journal of the ACM 24:607-622; 1977.
84. Yu, C.T.; Luk, W.S.; Siu, M.K. On the estimation of the number of desired records with respect to a given query. ACM Transactions on Database Systems 3:41-56; 1978.
85. Salton, G.; Wong, A. Generation and search of clustered files. ACM Transactions on Database Systems 3:321-346; 1978.
86. Kar, G.; White, L.J. A distance measure for automatic document classification by sequential analysis. Information Processing & Management 14:57-66; 1978.
87. Hamill, K.A.; Zamora, A. The use of titles for automatic document classification. Journal of the American Society for Information Science 31:396-402; 1980.
88. van Rijsbergen, C.J. Further experiments with hierarchic clustering in document retrieval. Information Storage and Retrieval 10:1-14; 1974.
89. Blashfield, R.K. Mixture model tests of cluster analysis: accuracy of four hierarchical methods. Psychological Bulletin 83:377-388; 1976.
90. Kuiper, F.K.; Fisher, L. A Monte Carlo comparison of six clustering procedures. Biometrics 31:777-783; 1975.
91. Mezzich, J.E. Evaluating clustering methods for psychiatric diagnosis. Biological Psychiatry. 13:265-281; 1978.
92. Milligan, G.W. An examination of the effect of six types of error perturbation on fifteen clustering algorithms. Psychometrica 45:325-342; 1980.
93. Milligan, G.W. A review of Monte Carlo tests of cluster analysis. Multivariate Behavioral Research 16:379-407; 1981.
94. Milligan, G.W.; Isaac, P.D. The validation of four ultrametric clustering algorithms. Pattern Recognition 12:41-50; 1980.
95. Morey, L.C.; Blashfield, R.K.; Skinner, H.A. A comparison of cluster analysis techniques within a sequential validation framework. Multivariate Behavioral Research 18:309-329; 1983.
96. Adamson, G.W.; Bawden, D. Comparison of hierarchical cluster analysis techniques for the automatic classification of chemical structures. Journal of Chemical Information and Computer Sciences 21:204-209; 1981.
97. Willett, P. A comparison of some hierarchical agglomerative clustering algorithms for structure-property correlation. Analytica Chimica Acta 136:29-37; 1982.
98. Murtagh, F. A New Approach to the Comparison of Hierarchic Clusterings. Technical report, University College, Dublin; 1983.

99. Murtagh, F. Structure of hierarchic clusterings: implications for information retrieval and multivariate data analysis. Information Processing & Management 20:611–617; 1984.
100. Salton, G.; Fox, E.A.; Wu, H. Extended Boolean information retrieval. Communications of the ACM 26:1022–1036; 1983.
101. Voorhees, E.M. The efficiency of inverted index and cluster searches. Proceedings of the Ninth International Conference on Research and Development in Information Retrieval 164–174; 1985.
102. Garland, K. An experiment in automatic hierarchical document classification. Information Processing & Management 19:113–120; 1983.
103. Enser, P.G.B. Automatic classification of book material represented by back-of-the-book index. Journal of Documentation 41:135–155; 1985.
104. Becker, D.S. Enhancing the retrieval effectiveness of large information systems. In: Current Research on Scientific and Technical Information Transfer. New York: Jeffrey Norton Publishers; 1977.
105. Willett, P. Query-specific automatic document classification. International Forum on Information and Documentation 10:28–32; 1985.
106. Willett, P.; Winterman, V.; Bawden, D. Implementation of non-hierarchic cluster analysis methods in chemical information systems. Selection of compounds for biological testing and clustering of substructure search output. Journal of Chemical Information and Computer Sciences 26:109–118; 1986.
107. Wade, S.J.; Willett, P. INSTRUCT: A teaching package for experimental methods in information retrieval. Part III. Browsing, clustering and query expansion. Program 22:44–61; 1988.
108. Croft, W.B.; Wolf, R.; Thompson, R. A Network Organization used for Document Retrieval. Technical report COINS 83-05, University of Massachusetts, Amherst, MA; 1983.
109. Croft, W.B.; Parenty, T.J. A comparison of a network structure and a database system used for document retrieval. Information Systems 10:377–390; 1985.
110. Crouch, D.B. A file organization and maintenance procedure for dynamic document collections. Information Processing & Management 11:11–21; 1975.
111. Goffman, W. An indirect method of information retrieval. Information Storage and Retrieval 4:361–373; 1969.
112. Croft, W.B.; van Rijsbergen, C.J. An evaluation of Goffman's indirect retrieval method. Information Processing & Management 12:327–331; 1976.
113. Mansur, O. An associative search strategy for information retrieval. Information Processing & Management 16:129–137; 1980.
114. Oddy, R.N. Information retrieval through man-machine dialogue. Journal of Documentation 33:1–14; 1977.
115. Preece, S.E. An online associative query modification methodology. Online Review 4:375–382; 1980.
116. Croft, W.B.; Thompson, R.H. The use of adaptive mechanisms for selection of search strategies in document retrieval systems. In: Research and Development in Information Retrieval (Edited by C.J. van Rijsbergen). Cambridge: Cambridge University Press; 1984.
117. McCall, F.M. and Willett, P. Criteria for the selection of search strategies in best match document retrieval systems. International Journal of Man-Machine Studies 25:317–326; 1986.
118. Robertson, S.E. and Sparck Jones, K. Relevance weighting of search terms. Journal of the American Society for Information Science 27:129–146; 1976.
119. Wong, Y.C.A. Studies on Clustered Files. PhD thesis, Cornell University; 1978.