Comparison of Hierarchic Agglomerative Clustering Methods for Document Retrieval

A. EL-HAMDOUCHI AND P. WILLETT*

Department of Information Studies, University of Sheffield, Western Bank, Sheffield S10 2TN

This paper considers the use of the single linkage, complete linkage, group average and Ward hierarchic agglomerative clustering methods for document retrieval. The methods are used to cluster seven document test collections for which queries and relevance judgements are available. Several retrieval strategies are described which allow searches to be carried out of the clustered document files resulting from the use of the four methods. These searches suggest that the group average method is the most suitable for document clustering purposes; however, searches of the unclustered document collections and of a simpler type of clustered file (based on pairs of nearest neighbours) usually result in better levels of retrieval effectiveness than searches of the clustered collections.

Received March 1987, revised June 1987.

1. INTRODUCTION

Cluster analysis, or automatic classification, is a technique which allows the identification of groups, or clusters, of similar objects in multi-dimensional space. Cluster analysis methods were first developed for use in the life sciences[1] but have since been used in a very wide range of application areas.[2-5] One such area is that of document retrieval systems, which are designed to retrieve those documents from a database which are relevant to a user's query. 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 organisation and the search strategy take some account of the relationships which exist between the documents in a database.[6, 7]

Early experiments in document clustering involved the use of partitioning methods, primarily because such methods are computationally attractive when large datasets need to be processed.[8, 9] However, the methods typically require the specification of several experimental parameters, e.g. the number of clusters present in the dataset, and the partitions obtained are often strongly dependent upon the order in which the objects have been clustered. More importantly, retrieval experiments show that searches of the clustered files of documents resulting from the use of these methods are substantially less effective than conventional searches of the corresponding, non-clustered files.[10, 11] Interest has hence focussed upon the use of hierarchic agglomerative methods which result in classifications containing small clusters of very similar documents nested within larger clusters of less similar documents.[12] Jardine and van Rijsbergen[13] demonstrated that searches of the classifications resulting from one such method, the single linkage method, have the potential to be more effective than conventional nearest neighbour searches in which documents are ranked in order of decreasing similarity with a query. Similar results were obtained by van Rijsbergen and Croft,[14] and Croft[15] demonstrated that such improved effectiveness can actually be obtained in practice in some cases. All of these studies involved the use of the single linkage method, owing to its desirable theoretical characteristics[16, 17] and to the ease with which it can be implemented on a large scale, an important feature when very large document collections need to be processed.[18-20]

Despite the undoubted advantages of the single linkage method, many empirical studies, using both real and simulated data, have suggested that it is by no means the most generally useful method.[21-24] This conclusion has recently been tested in the context of document clustering by Griffiths *et al.*[25, 26] and by Voorhees[27] who have made extended comparative studies of several hierarchic agglomerative methods; this work has considered the structures of the hierarchies produced by the different methods, the extent to which the methods distort the inter-document similarity matrices during the generation of the classifications, and the retrieval effectiveness of a range of cluster searching strategies. Griffiths *et al.* used the single linkage, complete linkage, group average and Ward methods, and found that the latter three methods gave broadly comparable results which were far superior to those obtained with the single linkage method; of these three, group average gave slightly better results than the other two methods in recall-oriented searches.[25] However, later work by this group, using a slightly different search strategy, suggested that Ward's method was the best of the four which were tested.[26] Voorhees considered the single linkage, complete linkage and group average methods; she again found that the single linkage classifications were inferior to the others, but suggested that complete linkage, rather than group average, was the best method of those tested.[27] Thus, although the work to date has identified single linkage as being the least useful method for document clustering, there is still disagreement as to which is the most generally useful of the available methods; in addition, the work of Griffiths *et al.* was restricted to small collections containing only 800 documents or less.

In this paper, we present the results of a comparative study of the single linkage, complete linkage, group average and Ward methods when they are used to cluster seven collections of documents for which queries and judgements of relevance are available. Section 2 describes the clustering methods and document collections which were used, and discusses the implementation of the

* To whom all correspondence should be addressed.

methods when large datasets need to be processed. Section 3 describes the search techniques which were used and presents the results of the retrieval experiments which were carried out. The discussion of these results and our conclusions are presented in sections 4 and 5.

2. DOCUMENT COLLECTIONS AND CLUSTERING METHODS

2.1. Document collections

The experiments used seven collections of documents, queries and relevance judgements to ensure that the results were not unduly influenced by the characteristics of a particular dataset. The collections were as follows:

Keen. A set of 800 document titles, augmented by manually assigned index terms, and 63 queries on the subject of librarianship and information science.

Cranfield. A set of 1400 documents and 225 queries on the subject of aerodynamics. These documents and queries are characterised by lists of manually assigned index terms, whereas all of the following datasets have been automatically indexed from natural language query statements and abstracts and/or titles.

Evans. A set of 2542 document titles and 39 queries from the INSPEC database.

Harding. A set of 2472 documents and 65 queries from the INSPEC database. The documents used are a sub-set of those in the Evans collection, but with the titles augmented by abstracts and with a larger set of queries.

LISA. A set of 6004 document titles and abstracts from the Library and Information Science Abstracts database, together with 35 queries obtained from students and staff in this department.

INSPEC. A set of 12684 document titles and abstracts from the INSPEC database, together with 77 queries collected at Cornell and Syracuse universities.

UKCIS. A set of 27361 document titles from the Chemical Abstracts Service database, together with 182 queries collected by the United Kingdom Chemical Information Service in the early 1970s.

In each collection, the words in the documents and queries were stemmed using a suffix-stripping algorithm (after the elimination of common words on a stopword list). Duplicate stems were then removed, and the documents and queries stored for processing as lists of binary stem numbers. These collections have been used for several previous research projects into information retrieval systems (both in our laboratory and elsewhere); they cover a wide range of types of collection, including both long and short query and document descriptions, and have been shown to exhibit a range of clustering tendencies.[26, 28]

2.2. Clustering methods

A simple algorithm for generating hierarchic document classifications is as follows:[12]

(1) Calculate the set of inter-document similarity coefficients.

(2) Put each document in a cluster on its own.

(3) Form a new cluster by the fusion of the most similar pair of current clusters, i and j say.

(4) Update the inter-document similarity matrix by deleting the rows and columns corresponding to i and j, and by calculating the entries in the row and column corresponding to the new cluster $i+j$.

(5) Go to (3) if the number of clusters is greater than one.

The hierarchic agglomerative methods differ in the manner in which the third and fourth steps in the algorithm above are carried out. The four methods used in this work are as follows.

Single linkage. In this method, the similarity between a pair of clusters is taken to be the similarity between the most similar pair of documents, one of which is in each of the clusters. The clusters formed have the property that any document in a cluster is more similar to at least one other document in that cluster than to any document in another cluster. A characteristic of this method is its tendency to form loosely bound clusters with little internal cohesion, the phenomenon which is generally referred to as chaining.

Complete linkage. This is the converse of single linkage since the least similar pair of documents in two clusters forms the basis for the measurement of inter-cluster similarity: thus, each document in a cluster is more similar to the most dissimilar document in that cluster than to the most dissimilar document in any other cluster. This definition of cluster membership is very much stricter than that for single linkage, and thus the large straggly clusters in the latter case are here replaced by large numbers of small, tightly bound clusterings.

Group average. This method results in clusters such that each document in a cluster has a greater average similarity to the remaining members of its cluster than it has to all the documents in any other cluster. It thus represents a mid-point between the two extreme types of linkage method, i.e. single linkage and complete linkage. There are several types of 'average' linkage method:[1] group average was chosen since it satisfies the reducibility principle[12] (which ensures that the procedure cannot result in inversions in the dendrogram representing the hierarchy). Group average is also known to minimise the distortion imposed on the inter-object similarity matrix when a hierarchic classification is generated.

Ward's method. Those two clusters are fused which result in the least increase in the sum of the distances from each document to the centroid of the cluster containing it. This method tends to result in spherical, tightly bound clusters; while these clusters may not truly reflect the underlying structure in a data set, they have been found to give excellent results in many comparative studies of clustering methods.[21–26]

2.3. Implementation of the methods

The use of clustering methods for document retrieval poses severe implementation problems, owing to the computational requirements associated with the generation of the classifications. Hierarchic agglomerative methods involve the processing of the inter-document similarity coefficients, and thus algorithms for their implementation must have a time requirement of at least $O(N^2)$ for a collection containing N documents; moreover, some of the algorithms which have been described in the literature have $O(N^2)$ storage requirements (e.g. the algorithm given previously in Section 2.2). Recent developments in hierarchic clustering algorithms are reviewed by Murtagh[12, 29] who has emphasised the central role of nearest neighbour searching in efficient clustering algorithms. However, most of the currently available nearest neighbour procedures are appropriate only when the dimensionality of the data is low, and there has thus been considerable effort devoted to the development of efficient nearest neighbour and clustering procedures which can be used for the processing of bibliographic databases.[30-33] This work now allows hierarchic clustering methods to be used for the clustering of large document collections, and the programs that were used in this work are summarised below.

The single linkage and complete linkage methods were implemented using Sibson's SLINK and Defays' CLINK algorithms respectively[34, 35] with the inter-document similarities calculated using the fast inverted file algorithm described by Willett.[19] The similarity measure used was the Dice Coefficient.[7]

Ward's method was implemented using the reciprocal nearest-neighbour (RNN) algorithm described by Murtagh;[12, 29] the modification of this algorithm for document clustering applications is presented in detail by El-Hamdouchi and Willett.[33]

The SLINK, CLINK and RNN algorithms have computational requirements of $O(N^2)$ time and $O(N)$ storage; analogous algorithms for the group average method have $O(N^2)$ time but also require $O(N^2)$ space, making them very difficult to implement on large datasets unless special steps are taken. The fusion step in the group average method for some pair of clusters, i and j, involves the calculation of the mean similarity coefficient across all pairs of documents for which one document is in the ith cluster and one document in the jth cluster; this becomes very time-consuming if the clusters are at all large. Voorhees[27] presents an algorithm for the group average method in which all of the pair-wise comparisons are replaced by a single calculation that measures the similarities between the *centroids* of the two clusters (where the centroid of a cluster is the sum of the term vectors corresponding to the documents contained in that cluster); this algorithm gives the same results as the conventional procedure if, and only if, the Cosine Coefficient[7] is used as the similarity coefficient. Voorhees' observation allows the group average method to be implemented efficiently using the RNN program developed for generating the Ward classifications.

The four clustering algorithms were encoded in FORTRAN and run on an IBM 3083 computer using level-3 optimisation. The CPU times required to generate single linkage, complete linkage, group average and Ward classifications for the seven document collections of Section 2.1 are listed in Table 1. It will be seen that the two groups of algorithm (SLINK and CLINK as against the two RNN algorithms) generally differ considerably in their computational requirements, the obvious exceptions being the Evans and UKCIS datasets. For these two collections, the documents are represented by keyword stems extracted from the document titles, and the exhaustivity of the document indexing, i.e. the number of index terms assigned to each document, is quite low (an average of 6.6 terms per document for Evans and 6.7 for UKCIS). The efficiency of nearest-neighbour searching, which lies at the heart of the RNN algorithm, is known to be strongly affected by indexing exhaustivity[19, 30] and thus the RNN algorithm is most efficient in operation for these two document collections.

Table 1. Run times in CPU seconds on an IBM 3083 for generation of hierarchic agglomerative classifications for seven document test collections

Collection	SL	CL	GA	WM
Keen	4	4	8	8
Cranfield	14	15	54	56
Evans	25	27	25	20
Harding	36	40	159	150
LISA	196	215	1001	997
INSPEC	840	929	3322	3416
UKCIS	968	990	1208	1238

SL, single linkage; CL, complete linkage; GA, group average; WM, Ward's method.

3. CLUSTER SEARCHING

3.1. Searching strategies

Two main types of strategy have been described in the literature for the searching of hierarchic document classifications.[14, 15] A *top-down* search starts at the root of the binary tree representing a classification, and compares the term vector representing the query with the centroid term vectors corresponding to the documents in the right and left sub-trees; that sub-tree is chosen for which the query-centroid similarity is greater, and the search continues down the tree in the same manner until some retrieval criterion is satisfied, typically until the query-cluster similarity or the size of the current cluster falls below some user-specified threshold value. A *bottom-up* search commences at the base of the tree, and moves upwards until the retrieval criterion is satisfied. Croft[15] and Griffiths *et al.*[26] have demonstrated that searches which are based on the small clusters at the base of the tree give the best search results, and our experiments have hence considered only the use of bottom-up search strategies.

The implementation of a bottom-up search requires that some means must be available for deciding where the search should commence. Three procedures were used in our work. The simplest approach, which we shall refer to subsequently as a Type A search, is to assume that a single relevant document is already available; this is often the case in a practical retrieval situation. Alternatively, if a relevant document is not available, the starting point in a Type B search is obtained by carrying out a conventional, non-clustered, best match search in which the documents are ranked in order of decreasing similarity with the query; the document at the top of the ranking is then chosen as the starting point for the bottom-up search. Rather than using an individual document, the Type C search uses the *bottom-level clusters*.[15] A bottom-level cluster is the smallest cluster in a hierarchy which contains a specified document; thus, for N documents, there are N bottom-level clusters (up to $N/2$ of which can be duplicates) and the starting point for the bottom-up search is that bottom-level cluster whose centroid is most similar to the query.

Given a starting point, the bottom-up search identifies the relevant document, best-matching document, or best-matching bottom-level cluster at the base of the tree. The search then moves upwards, adding into the retrieved set the documents associated with each parent node in turn. The search continues until a sufficient number of documents has been retrieved, this number being specified as required by the user. In this paper we include the results only for a threshold of 10 documents (so as to keep the volume of results to an acceptable level); El-Hamdouchi[36] presents analogous results obtained with other thresholds.

For comparison with these tree searching procedures, the Type D search involved matching the bottom-level cluster centroids against the query, and then ranking the clusters in order of decreasing similarity with the query. The required number of documents were then taken from the top of the ranking; this procedure has been used previously and shown to be highly effective in operation.[15] The similarities between the cluster centroids and the queries in the Type B, C and D searches were calculated using the Cosine Coefficient, with the query terms weighted using inverse document frequency (IDF) weighting.[7]

3.2. Measurement of retrieval effectiveness

Associated with each of the queries in a collection is a list of the documents which have been judged previously to be relevant to that query; these relevance judgements have been compiled by e.g., pooling the output of different types of search or by extended manual searches. It is hence possible to evaluate the performance of a search by comparing the documents retrieved with those which should have been retrieved.[6, 7]

The primary evaluation measure for the searches was the effectiveness measure, E, which has been used extensively in previous studies of document clustering.[6, 13, 14] For a search that retrieves a set of documents that give rise to recall and precision figures of R and P respectively, E is defined to be

$$1-(1+\beta^2)\,PR/(\beta^2 P+R)$$

where β is a parameter reflecting the relative importance attached to recall and to precision by the user. A value for β of 0.5 or 2.0 (the two values used in the results reported here) corresponds to attaching twice or one half as much importance to precision as to recall; the reader should note that low E values correspond to high retrieval performance. Following the practice of previous researchers,[13–15, 18] we have summarised the experiments by quoting the mean E values when averaged over the complete set of queries for a test collection (or over the complete set of relevant documents for a test collection in the case of the Type A searches). This measure has the

Table 2. E values for $\beta = 0.5$ and $\beta = 2.0$ in cluster searches

	SL		CL		GA		WM	
Collection	$\beta = 0.5$	$\beta = 2.0$	$\beta = 0.5$	$\beta = 2.0$	$\beta = 0.5$	$\beta = 2.0$	$\beta = 0.5$	$\beta = 2.0$
(*a*) Type A searches (commencing with a known relevant document)								
Keen	0.83	0.87	0.83	0.88	0.78	0.83	0.78	0.84
Cranfield	0.78	0.76	0.84	0.82	0.68	0.65	0.72	0.70
Evans	0.88	0.92	0.88	0.93	0.85	0.91	0.86	0.92
Harding	0.88	0.93	0.90	0.94	0.82	0.89	0.84	0.91
LISA	0,88	0.90	0.90	0.91	0.83	0.87	0.87	0.89
INSPEC	0.92	0.96	0.92	0.96	0.87	0.93	0.89	0.94
UKCIS	0.95	0.98	0.94	0.97	0.93	0.97	0.93	0.97
(*b*) Type B searches (commencing with the top-ranking document from a conventional, non-clustered best-match search)								
Keen	0.89	0.87	0.88	0.86	0.85	0.84	0.84	0.83
Cranfield	0.87	0.83	0.93	0.92	0.80	0.74	0.83	0.79
Evans	0.91	0.94	0.94	0.96	0.91	0.94	0.91	0.94
Harding	0.93	0.95	0.95	0.97	0.90	0.94	0.92	0.95
LISA	0.95	0.94	0.96	0.95	0.93	0.93	0.94	0.93
INSPEC	0.94	0.96	0.96	0.97	0.90	0.93	0.92	0.95
UKCIS	0.96	0.97	0.96	0.97	0.94	0.96	0.94	0.96
(*c*) Type C searches (commencing with the best-matching bottom-level cluster)								
Keen	0.86	0.86	0.85	0.85	0.81	0.81	0.80	0.79
Cranfield	0.85	0.80	0.90	0.87	0.78	0.72	0.81	0.76
Evans	0.90	0.94	0.93	0.95	0.90	0.93	0.89	0.92
Harding	0.92	0.94	0.93	0.95	0.89	0.93	0.90	0.94
LISA	0.94	0.94	0.95	0.95	0.91	0.91	0.91	0.91
INSPEC	0.94	0.97	0.95	0.97	0.87	0.93	0.90	0.94
UKCIS	0.96	0.98	0.96	0.97	0.94	0.96	0.93	0.96
(*d*) Type D searches (where the bottom-level clusters are ranked in order of decreasing similarity with the query)								
Keen	0.82	0.83	0.80	0.79	0.78	0.78	0.78	0.77
Cranfield	0.82	0.77	0.85	0.80	0.77	0.70	0.79	0.73
Evans	0.89	0.93	0.88	0.92	0.85	0.90	0.85	0.89
Harding	0.90	0.93	0.90	0.94	0.85	0.90	0.87	0.91
LISA	0.87	0.87	0.87	0.86	0.87	0.86	0.87	0.87
INSPEC	0.90	0.94	0.89	0.93	0.88	0.92	0.88	0.93
UKCIS	0.93	0.96	0.93	0.96	0.93	0.96	0.92	0.95

Table 3. *T* and *Q* values in cluster searches

	SL		CL		GA		WM	
Collection	T	Q	T	Q	T	Q	T	Q
				(*a*) Type B searches				
Keen	80	25	83	22	105	21	111	22
Cranfield	279	109	148	136	444	89	364	96
Evans	46	19	30	20	44	19	45	18
Harding	63	38	40	38	88	37	69	34
LISA	17	21	16	21	26	21	25	19
INSPEC	68	37	50	38	123	27	94	33
UKCIS	137	114	124	111	177	104	185	107
				(*b*) Type C searches				
Keen	101	24	104	16	139	14	143	13
Cranfield	325	93	222	96	473	72	405	75
Evans	54	20	38	18	50	19	54	13
Harding	76	38	40	43	100	32	89	35
LISA	24	22	18	22	39	19	37	15
INSPEC	79	40	54	43	150	27	119	33
UKCIS	132	134	133	123	226	105	241	101
				(*c*) Type D searches				
Keen	129	18	138	11	159	11	157	11
Cranfield	390	72	323	67	490	50	448	50
Evans	59	14	62	14	76	10	78	5
Harding	95	36	84	32	133	24	120	28
LISA	48	15	47	12	51	12	49	13
INSPEC	127	40	126	43	179	27	138	33
UKCIS	226	101	221	85	228	98	270	87

limitation that it varies over a relatively small range, as can be seen from the figures in the tables of results; other limitations are discussed by El-Hamdouchi.[36] Accordingly, two further measures of retrieval effectiveness were noted. These are the total number of relevant documents, *T*, retrieved at each threshold size when summed over all of the queries associated with each collection, and the number of queries, *Q*, for which no relevant documents at all were identified in the search. Thus, effective searches are those with high *T* and low *Q* or *E* values.

The main experimental results are presented in Tables 2 and 3. Each element of Table 2 contains the *E* value calculated with $\beta = 0.5$ and $\beta = 2.0$ for each of the four types of search strategy. Each element of Table 3 contains the *T* and *Q* for (*a*) Type B, (*b*) Type C and (*c*) Type D searches; no *T* and *Q* values are listed for the Type A search since this retrieval strategy assumes that full relevance data is already available prior to commencing the search.

4. DISCUSSION

The variation in the results in Tables 2 and 3 reflects the characteristics not only of the four clustering methods but also of the document test collections and the search strategies. Thus, the Keen and Cranfield collections are known to respond well to clustering since the relevant documents for these test collections are similar to each other and group together when a clustering method is applied to the dataset;[28] thus, the *E* values for these two collections are noticeably lower than for the other five collections.

An inspection of the results in Tables 2 and 3 suggests that the group average method gives the best retrieval performance. We have used the Kendall Coefficient of Concordance, *W*, to test this conclusion. Kendall's *W* tests the extent to which *k* rankings of the same set of *M* objects are in agreement with each other; a full description of the procedure is given by Siegal.[37] In the present context, we are interested in the ranking of the four clustering methods ($M = 4$) by the seven document test collections ($k = 7$) and the significance of the agreement between the collections was calculated for each of the search strategies and each of the evaluation measures. The agreements are significant at the 0.01 level of statistical significance for all of the sets of results in Tables 2 and 3, with the single exception of the *Q* values obtained in the Type D searches (where the agreement is significant at the 0.05 level). This is a striking result, given the very disparate natures of the document collections and search strategies. If a statistically significant value of *W* is obtained in the Kendall test, then the best estimate that can be made of the 'true' ranking of the *M* objects is provided by the sum of the ranks for each object when totalled across the *k* sets of rankings.[37] It is thus possible to obtain an overall ranking of the four methods on the basis of their retrieval effectiveness. This has been done in Table 4 which lists the rankings (estimated by the sums across all seven test collections) of the four methods for all combinations of search strategy and evaluation measure. It will be seen that, with very few exceptions indeed, the order of retrieval effectiveness is, first, group average, then Ward's method, then single linkage and, finally, complete linkage. Thus,

Table 4. Estimated rankings (based on the Kendall *W* test) of the four clustering methods

Search	Evaluation	SL	CL	GA	WM
Type A	$\beta = 0.5$	3	4	1	2
	$\beta = 2.0$	3	4	1	2
Type B	$\beta = 0.5$	3	4	1	2
	$\beta = 2.0$	3	4	1	2
	T	3	4	1	2
	Q	3	4	1.5	1.5
Type C	$\beta = 0.5$	3	4	2	1
	$\beta = 2.0$	3	4	1	2
	T	3	4	1	2
	Q	3	4	2	1
Type D	$\beta = 0.5$	4	3	1.5	1.5
	$\beta = 2.0$	4	3	1	2
	T	3	4	1	2
	Q	4	3	1	2

the results obtained here suggest that the group average method is the most appropriate for use in document retrieval systems.

As noted in the Introduction, previous work on the comparison of hierarchic document clustering methods has identified single linkage as being noticeably inferior to the other three methods considered here. While single linkage has performed badly in our experiments, the results obtained with it are still superior to those obtained with the complete linkage method. This finding is totally at variance with Voorhees' results[27] which suggested that the complete linkage method was superior to group average (and also to single linkage). We believe that the generally poor results obtained here with the complete linkage method are due to our use of the CLINK algorithm of Defays.[35] This algorithm is efficient in operation, having time and storage requirements of $O(N^2)$ and $O(N)$ respectively, but it is known to be dependent upon the order in which the documents in the database are processed.[35] For comparison with the CLINK classifications, we have carried out comparative tests using the complete linkage algorithm of Voorhees[27] to cluster the small Keen, Cranfield and Evans datasets. The results of these experiments are listed in Tables 5 and 6. A comparison with the corresponding results in Tables 2 and 3 shows clearly that Voorhees' algorithm gives results which are often superior to those resulting from the use of the CLINK algorithm. Some of the differences in effectiveness are substantial; indeed, the results of the Type D searches for the Keen and Cranfield collections are the best of all of the methods. It was not possible to use the Voorhees algorithm with the other document collections since its worst case $O(N^3)$ time and $O(N^2)$ space requirements make it very demanding of computer resources; for example, it took 38 times as long as the CLINK algorithm to cluster the small Keen collection. Thus, it seems that the complete linkage method can give good levels of retrieval performance, but only if an appropriate algorithm is used for the generation of the classifications.

The experiments so far have considered only the comparison of one cluster search with another, without consideration of the effectiveness obtainable from alternative retrieval techniques. Two such techniques were used here. The first of these was a conventional best match search in which a query is matched against each of the documents in a database and the documents ranked in decreasing order of similarity with the query. The second approach was based on a very simple form of clustered file in which each of the documents in a collection is grouped with its nearest neighbour, i.e. that document with which it is most similar. These pairs of documents, or *nearest-neighbour clusters* (NNCs), can then be ranked in relation to a query in just the same way as can a set of bottom-level clusters. NNCs represent a simple way of using inter-document similarity information; they can be generated much more efficiently than can a hierarchic agglomerative classification and can also be updated quite easily. Most importantly, retrieval tests have shown that they give search results which are comparable in effectiveness to conventional best-match searching.[26, 38] The results of using these two types of search have been taken from Griffiths *et al.*[26] and are listed in Table 7. A comparison of these results with those in Tables 2, 3, 5 and 6 shows that the NNC searches are superior, and often substantially so, to searches of all of the hierarchic classifications. A similar comment applies to the non-clustered best match search; the only exception to this observation is in the case of the Cranfield collection which has long been known to give good results in document clustering research.[13–15, 20, 28]

In view of the computational demands required for the generation of the classifications and the poor effectiveness of cluster searches, one may question whether the four hierarchic agglomerative clustering methods tested here provide appropriate means of structuring a document collection for information retrieval. This conclusion does not imply that alternative clustering methods, e.g. the NNCs discussed previously, or alternative cluster search strategies could not give better results. Of the strategies which we have considered, the Type D searches are noticeably superior to the Type B and C searches and rival many of the Type A searches (which could not be used in a practical environment since they assume the availability of relevance data prior to the search). The Type D search strategy would thus seem to be the most useful for retrieval purposes, despite the fact that it takes

Table 5. *E* values for $\beta = 0.5$ and $\beta = 2.0$ in complete linkage cluster searches using Voorhees' algorithm

	Type A		Type B		Type C		Type D	
Collection	$\beta = 0.5$	$\beta = 2.0$	$\beta = 0.5$	$\beta = 2.0$	$\beta = 0.5$	$\beta = 2.0$	$\beta = 0.5$	$\beta = 2.0$
Keen	0.83	0.88	0.89	0.87	0.86	0.86	0.75	0.74
Cranfield	0.80	0.78	0.90	0.87	0.88	0.85	0.80	0.74
Evans	0.88	0.93	0.93	0.95	0.92	0.94	0.82	0.88

Table 6. *T* and *Q* values in complete linkage cluster searches using Voorhees' algorithm

	Type B		Type C		Type D	
Collection	T	Q	T	Q	T	Q
Keen	78	24	95	18	171	6
Cranfield	224	111	252	93	436	53
Evans	35	20	39	20	94	3

minimal account of the hierarchic structure of the classification tree (since the search is based solely upon the penultimate level immediately above the documents which comprise the leaves of the tree). The bottom-level clusters are generally very small, containing only a pair of documents (except in the case of the single linkage method where the chaining phenomenon can result in the bottom-level clusters containing many hundreds of documents[26, 39]); since the bottom-level clusters are those which are generally formed first in an agglomerative clustering program, there is clear scope for increasing the efficiencies of such programs by terminating cluster growth once the initial clusters have been identified. A disadvantage of using the bottom-level clusters for a collection is that they are overlapping (since the bottom-level cluster for some document which joins the hierarchy at a fairly low similarity value will include the bottom-level clusters for some of the documents which are already connected into the cluster hierarchy).

An alternative approach would be to apply a similarity (or dissimilarity) threshold to a hierarchy and then to search the set of non-overlapping clusters defined by the resulting partition. Table 8 gives retrieval results for searches of group average and Ward classifications of the Keen and Cranfield collections where a partition has been created from the classification by applying a threshold. The search involved ranking the clusters in the partition in descending order of similarity with the query (i.e. in a manner analogous to the Type D searches); these results are noticeably superior to those in Tables 2 and 3 and rival those in Table 7. However, there are two problems with such an approach. Firstly, the partitions have been obtained by the purely empirical procedure of applying a series of thresholds to the hierarchies, and testing the retrieval effectiveness in each case to determine the best partition. There has been some interest in non-empirical means for the identification of partitions in hierarchies,[23] but it is not at present clear whether these methods could be used in the context of document retrieval, where partitions are required which contain very large numbers of small, tightly bound clusters (rather than the restricted numbers of clusters required in most applications of cluster analysis). The second problem is that many of the clusters in these partitions are, in fact, singletons; the most extreme case of this is with the Ward classification of the Keen collection where 61% of the documents are in a cluster on their own. Thus, while it is possible to identify clusters in a hierarchic agglomerative classification which can give levels of retrieval effectiveness comparable to those obtainable from NNC or non-clustered searching, the latter techniques seem more appropriate for use in operational environments.

5. CONCLUSIONS

In this paper we have considered the use of the single linkage, complete linkage, group average and Ward clustering methods in document retrieval systems. We have demonstrated that it is possible to implement these methods on a sufficiently large scale to allow the clustering of document collections of non-trivial size. Four different retrieval techniques were used to search the classifications resulting from the use of these clustering methods on seven document test collections. A comparison of the cluster searches demonstrates that the group average and complete linkage methods seems to offer the best levels of retrieval effectiveness, although the latter method requires substantial computation for the generation of the clusters if good results are to be

Table 7. *E, T* and *Q* values in conventional best-match and NNC searches

	Non-cluster searches				NNC searches			
Collection	$\beta = 0.5$	$\beta = 2.0$	*T*	*Q*	$\beta = 0.5$	$\beta = 2.0$	*T*	*Q*
Keen	0.73	0.72	186	4	0.72	0.71	202	6
Cranfield	0.80	0.73	433	52	0.75	0.68	533	35
Evans	0.78	0.85	113	3	0.80	0.85	103	4
Harding	0.83	0.88	155	19	0.83	0.89	149	24
LISA	0.80	0.78	74	9	0.79	0.77	78	7
INSPEC	0.80	0.87	233	10	0.83	0.89	203	11
UKCIS	0.89	0.92	340	75	0.90	0.94	316	77

Table 8. *E, T* and *Q* values for Type D-like searches of the optimal partitions for the Keen and Cranfield collections

	GA				WM			
Collection	$\beta = 0.5$	$\beta = 2.0$	*T*	*Q*	$\beta = 0.5$	$\beta = 2.0$	*T*	*Q*
Keen	0.75	0.74	175	7	0.73	0.73	195	7
Cranfield	0.75	0.68	547	44	0.77	0.71	489	48

obtained. All of the cluster search strategies tested here were usually inferior to best match searches of the unclustered document collections and of a simple form of clustered file in which documents are grouped with their nearest neighbours.

REFERENCES

1. P. H. A. Sneath and R. R. Sokal, *Numerical Taxonomy.* Freeman, San Francisco (1973).
2. M. R. Anderberg, *Cluster Analysis for Applications.* Academic Press, London (1973).
3. R. Dubes and A. K. Jain, Clustering methodologies in exploratory data analysis. *Advances in Computers* **19**, 113–228 (1980).
4. A. D. Gordon, *Classification.* Chapman and Hall, London (1981).
5. R. C. T. Lee, Clustering analysis and its applications. *Advances in Information Systems Science* **8**, 169–292 (1981).
6. C. J. van Rijsbergen, *Information Retrieval.* Butterworth, London (1979).
7. G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval.* McGraw-Hill, New York (1983).
8. J. A. Hartigan, *Clustering Algorithms.* Wiley, New York (1975).
9. H. Spath, *Cluster Analysis Algorithms.* Ellis Horwood, Chichester (1980).
10. G. Salton, *The SMART System.* Prentice-Hall, Englewood Cliffs (1971).
11. M. Fritsche, *Automatic Clustering Techniques in Information Retrieval.* Report no. EUR 5051e, Commission of the European Communities, Luxembourg (1974).
12. F. Murtagh, A survey of recent advances in hierarchical clustering algorithms. *The Computer Journal* **26**, 354–359 (1983).
13. N. Jardine and C. J. van Rijsbergen, The use of hierarchical clustering in informaton retrieval. *Information Storage and Retrieval* **7**, 217–240 (1971).
14. C. J. van Rijsbergen and W. B. Croft, Document clustering: an evaluation of some experiments with the Cranfield 1400 collection. *Information Processing and Management* **11**, 171–182 (1975).
15. W. B. Croft, A model of cluster searching based on classification. *Information Systems* **5**, 189–195 (1980).
16. N. Jardine and R. Sibson, *Mathematical Taxonomy.* Wiley, New York (1971).
17. L. Fisher and J. van Ness, Admissible clustering procedures. *Biometrika* **58**, 91–104 (1971).
18. W. B. Croft, Clustering large files of documents using the single-link method. *Journal of the American Society for Information Science* **28**, 341–344 (1977).
19. P. Willett, A fast procedure for the calculation of similarity coefficients in automatic classification. *Information Processing and Management* **17**, 53–60 (1981).
20. P. Willett, A note on the use of nearest neighbours for implementing single linkage classifications. *Journal of the American Society for Information Science* **35**, 149–52 (1984).
21. F. K. Kuiper and L. Fisher, A Monte Carlo comparison of six clustering procedures. *Biometrics* **31**, 777–783 (1975).
22. R. K. Blashfield, Mixture model tests of cluster analysis: accuracy of four agglomerative hierarchical methods. *Psychological Bulletin* **83**, 377–388 (1976).
23. R. Mojena, Hierarchical grouping methods and stopping rules: an evaluation. *The Computer Journal* **20**, 359–363 (1977).
24. G. W. A. Milligan, A review of Monte Carlo tests of cluster analysis. *Multivariate Behavioural Research* **16**, 379–407 (1981)
25. A. Griffiths, L. A. Robinson and P. Willett, Hierarchic agglomerative clustering methods for automatic document classification. *Journal of Documentation* **40**, 175–205 (1984).
26. A. Griffiths, H. C. Luckhurst and P. Willett, Using inter-document similarity information in document retrieval systems. *Journal of the American Society for Information Science* **37**, 3–11 (1986).
27. E. M. Voorhees, The Effectiveness and Efficiency of Agglomerative Hierarchic Clustering in Document Retrieval. Ph.D. thesis, Cornell University (1985).
28. A. El-Hamdouchi and P. Willett, Techniques for the measurement of clustering tendency in document retrieval systems. *Journal of Information Science* **13**, 361–365 (1987).
29. F. Murtagh, Complexities of hierarchic clustering algorithms: state of the art. *Computational Statistics Quarterly* **1**, 101–113 (1984).
30. S. A. Perry and P. Willett, A review of the use of inverted files for best match searching in information retrieval systems. *Journal of Information Science* **6**, 59–66 (1983).
31. F. Murtagh, Clustering and nearest neighbour searching. *Proceedings of INFORMATICS 8*, ASLIB, London, pp. 54–65 (1985).
32. C. Buckley and A. F. Lewit, Optimization of inverted vector searches. *Proceedings of the Eighth International Conference on Research and Development in Information Retrieval.* ACM, Washington, pp. 97–110 (1985).
33. A. El-Hamdouchi and P. Willett, Hierarchic document clustering using Ward's method. *Proceedings of the Ninth International Conference on Research and Development in Information Retrieval.* ACM, Washington, pp. 149–156 (1986).
34. R. Sibson, SLINK: an optimally efficient algorithm for the single link cluster method. *The Computer Journal* **16**, 30–34 (1973).
35. D. Defays, An efficient algorithm for a complete link method. *The Computer Journal* **20**, 364–366 (1977).
36. A. El-Hamdouchi, Using Inter-Document Relationships in Information Retrieval. Ph.D. thesis, University of Sheffield (1987).
37. S. Siegal, *Nonparametric Statistics.* McGraw-Hill,Tokyo (1956).
38. F. M. McCall and P. Willett, Criteria for the selection of search strategies in best match document retrieval systems. *International Journal of Man-Machine Studies* **25**, 317–326 (1986).
39. F. Murtagh, Structure of hierarchic clusterings: implications for information retrieval and multivariate data analysis. *Information Processing and Management* **20**, 611–617 (1984).