Query Clustering in a Large Document Space

S. Worona

Abstract

The Cranfield 424 document collection is clustered using queries and known relevance judgments. This clustering method is compared to a full search of the collection, and several searches using a standard clustering technique. Several new evaluation parameters are defined and applied to the experiment.

1. Introduction

One of the most important aspects of any information retrieval system is time — how quickly a user's request can be processed, the specified information generated, and the output returned to the user. This is especially true in a real-time system, where the optimum time is measured in seconds. For a large-sized document collection, search-time — the time spent scanning and correlating against the members of the collection — is critical, since it can become excessive, often varying with the size of the collection. Because of this, various techniques have been developed to shorten search-time. "Batching", that is, searching the document collection only once for several queries, has proven effective in reducing per-query search-time. This must be considered unworkable, however, in a real-time system, when only single queries are available. "Clustering" techniques, which use one "centroid" to represent many documents, also lower search time, and are, in addition, well suited for single-query real-time systems. Clustering is the operation which consists in dividing a document space into several groups, each of which is considered as a unit. Each cluster is represented by a centroid, similar in form to the documents it represents. On the surface, then, a collection of centroids is no different than a normal document collection.

All clustering operations can be divided into two parts. The first controls how the clusters are to be generated from the document collection and how centroids are to be assigned to the clusters. The second determines a search-scheme by which the collection of centroids is scanned and certain clusters chosen for expansion. In addition, the final ranking of retrieved documents and the subsequent use of relevance feedback techniques [2,8] may become part of a clustering system. These last considerations, however, are not peculiar to clustering, and are not taken up in this report.

2. Generating Clusters

Several methods of generating document clusters are currently being used in experimental systems, among which are those developed by Bonner, Rocchio, and Dattola. [3,4,5,6,7] Most of these make use of correlations between the documents to be clustered, grouping those which correlate highest, and then forming each cluster centroid from the concept vectors of the documents included in that cluster. Thus, these techniques produce clusters of documents whose concept vectors are highly related to each other, each cluster being represented by another vector which is a mathematical combination of the documents it represents. Parameters for these clustering routines include the number of clusters desired, the number of loose documents permitted, the level of correlation between cluster members, and the degree of "overlap" of the clusters.

In [1], V. R. Lesser suggests that a different clustering method be used. His method is a two-pass algorithm, consisting of the following steps: First all queries previously processed by a system are clustered by a standard method. The resulting query-clusters are used to cluster the document collection in one of three ways:

1. All documents correlating highly with the centroid of a query cluster form a cluster.
2. All documents correlating highly with one or more queries of any one query-cluster form a cluster.
3. All documents judged relevant to one or more queries of any one query-cluster form a cluster.

The centroids of the resulting document clusters are the centroids of the corresponding query-clusters. In this way, each document is represented in its cluster by a centroid formed from queries rather than documents. According to Lesser, this process is effective since incoming queries are more likely to be similar to past queries than to documents. Thus, Lesser believes, new queries are less likely to fall between query clusters than between document clusters. (See Fig. 1)

In addition to this property, the query clustering method, espec-cially when performed with relevance judgments, may enable a retrieval system to "mature" as more and more queries are entering into the system. As in all clustering schemes, updating the clusters would be periodical, depending on both the number of queries processed, and the number of new documents received.

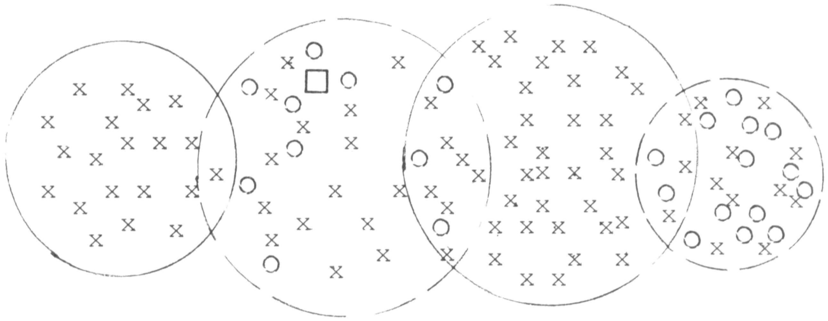
x Document

○ Query

—— Standard document cluster

--- Query cluster with associated documents

□ "New" query falling <u>within</u> query cluster,
but <u>between</u> document clusters



Query-Document Space (From [1])

Fig. 1

3. Searching Clustered Collections

In general, collections of centroids can themselves be clustered to form "super-centroids", etc. With each new clustering, another "level" is added to the degree of the required search. Only simple centroids of level two are considered here.

When searching such centroids, one parameter is crucial — the number of clusters to be expanded. Of course, numerous other considerations are also important, including the method of determining the "goodness" of the centroids searched. These are, however, superseded in importance by the former, which controls the portion of the collection that is to be used in the search. If this portion is too large, the search is likely to be successful, but the resulting saving in search time may be insignificant. On the other hand, taking too small a piece of the document collection may produce poor, although rapidly obtained results.

4. Parameters for Evaluating Cluster Searches

As in any search attempt, it is important to determine the recall and precision of a clustered search. However, other considerations also enter the picture as full searches are replaced by centroid matches. Perhaps the most important, and possibly the most difficult to measure, is the amount of savings in machine-time offered by the centroid search. All other values used to decide the effectiveness of a search must be considered in combination with the statistics of how much time is saved. In this paper, no attempt is made to combine such time considerations with any other parameters. Rather, all parameters are presented separately. This is done because no acceptable method of combining these parameters has been decided upon. Indeed, the desired results may vary with the application: given the decision of whether a search retrieving 45% of all relevant documents while scanning 45% of a collection is better or worse than one retrieving 30% while using only 30% of the collection, different users would undoubtedly give different answers.

The factor used in this paper to measure time savings is *correlation percentage*, the ratio of the number of documents *and centroids* scanned to the number of documents in the collection. This will, in most applications, be a number between 0 and 1, with a full search always evaluated at 1.

Given any particular query, it is reasonable to ask how different cluster-generating procedures rate as creators of good "targets" for a search. For example, a scheme generating clusters, none of which contain a large number of the relevant documents for that query, will yield poor results no matter what the search technique, because several clusters must then be expanded before all the relevant documents are retrieved, thus destroying the effectiveness of clustering. It is then necessary to examine the "target value" of the tested clustering schemes. For a given query, the "target clusters" are those *n* clusters which, between them, contain the largest number of relevant documents, where *n* is the number of clusters to be expanded. Given two clusters with equal numbers of relevant documents, the smaller is chosen. When more than 1 cluster is to be expanded, the target clusters are those which have the smallest total of (different) documents, while still containing the most relevant possible. The target value of a clustering scheme for a particular query is the ratio of the number of relevant documents in the target clusters to the number of relevant for that query.

The ideal system is one in which the target value for all queries is 1, and the correlation percentage is minimized. This alone, however, will not assure good results. After ideal clusters have been formed for each query, it is necessary that they correlate in the proper way. The "aim" is then defined as a measure of how well a centroid was assigned to each cluster. The "aim clusters" of a given query are those n clusters which are expanded during a search. As with "target value", the "aim value" is the ratio of relevant documents in the aim clusters to the total number of relevant for the query. This should not be confused with the "recall ceiling", a similar concept, but one which yields different results. (The recall ceiling does not take into account relevant documents dropped because they did not correlate highly enough with the query.)

Although this paper does not deal with a wide enough range of experimental data to make full use of aim and target values, these concepts make it possible to separate judgments on clustering from those on centroid assignment, and may be valuable in an in-depth study of clustering techniques.

Perfect values for aim and target should do much to optimize a search scheme, and when combined with low correlation percentages may be even more effective. One more consideration is important, however. Take, for example, a collection of clusters, all of which contain all the relevant documents for a particular query. Another set of clusters may contain only one cluster including all such documents. Quite conceivably, aim, target, and correlation percentage values may be identical for the two schemes on the given query, yet, the two schemes may be quite different. The former may have a great deal of "wasted" documents where they are not needed by the query. The term "rejection" is used to refer to the tendency of a clustering scheme to "reject" relevant documents from all but the target cluster(s) of a given query. It is defined as the ratio of occurrences (not necessarily different) of relevant documents in the target clusters to occurrences (not necessarily different) of relevant documents throughout the clustered collection. Again, a value of 1 is optimal.

5. The Experiment

Lesser's attempt to demonstrate the effectiveness of query clustering yielded encouraging results. The limitations of the experiment, however, put the results on a less-than-solid basis. Since the most damaging of these limitations was the small size of the collection used, (only 35 queries and 82 documents), it was decided that an experiment on a larger collection was in order. In the present experiment, the Cranfield 424 collection, containing 424 documents and 155 queries, is used. As in Lesser's approach, the procedure is in two phases — first query clusters are formed, and then document clusters are generated from these. Unlike Lesser, who associated documents in a cluster if they correlated highly with one or more queries in any one query cluster, the current experiment uses relevance judgments to form document clusters.

The 155 queries are split into two groups, one of 130 and one of 25, by choosing every sixth query for the smaller group. (This process is used because the collection is arranged in order of subject area, so that taking any continuous subgroup would destroy generality.) The 130 queries were clustered using Dattola's clustering algorithm [7], producing 11 clusters with an overlap of 13.9%. Clusters range in size from 17 to 37 queries, with an average of 28. Queries in this collection have from 3 to 22 relevant documents, averaging 6½. Document clusters are then formed by replacing the list of queries with a list of relevant documents for each cluster. Since this experiment is being done using Cornell University's SMART system, each centroid is easily associated with a different collection. Both documents and queries are generally specified by a four-digit integer, and both have the same general appearance. It is thus possible to use documents and queries interchangeably in almost all applications.

The resulting document collection is described below (Table 1). Overlap was not calculated for this collection, although it is estimated to be about twenty to thirty percent. Statistics are available giving the number of times a document appears in a given number of collections (for example, only 102 out of 424 documents appear in exactly 1 cluster), from which the overlap is estimated.

It is interesting to note that a collection of query clusters with an overlap of only 14% is turned into document clusters with an overlap nearly twice as high. The reasons for this include the fact that many documents are relevant to a great many queries, and that sets of co-relevant documents are common.

In a clustering algorithm, the question of "loose documents" must be considered. Loose documents are those which, at some point in the clustering procedure, belong to no cluster. If such documents are not "blended in" in one way or another, subsequent queries are likely to have artificially low recall ceilings. After associating all of the relevant documents with the queries of the initial query collection, it is found that some 29 documents remain loose. Fifteen of these documents are found to be relevant to one or more of the 25 test queries, so these documents can be blended in. This is done by correlating all 15 documents with all 11 centroids, and including each document in the two clusters, whose centroids are closest to the documents; in addition, each document is also included in any cluster with whose centroid it correlates by .1500 or higher. The figures of two clusters and .1500 are chosen to maintain the characteristic overlap of the collection at its original level, and are, for the most part, a product of intuition.

Since a clustered-search is inherently different from a full search, it is desirable that other clustering methods be used for comparison. Thus, the Cranfield 424 document collection was itself clustered using Dattola's algorithm. The results of this operation appear in Table 1. Notice, in particular, the great difference in the number of concepts appearing in an average cluster for the two cluster schemes. This points up the fact that Dattola's algorithm produces clusters with document-related centroids, while query-clustering techniques produce centroids resembling queries rather than documents.

Four test searches are made, each with the same initial parameters: All documents correlating greater than 0 are considered; all other values are set at default conditions. One full search is done, one clustered search using clusters generated by query-clustering, and two clustered searches using Dattola's algorithm to generate clusters. The first of these two calls for one cluster only to be expanded for each query, while the second calls for two. (A trial was made on which three clusters were to be expanded for each query, but this run failed because insufficient space was available on the program disc storage unit.) Complete statistics are available (including aim, target, and rejection values — see Appendix A — where applicable) for the full search, query-clustered search, and the first of the two normally-clustered searches. Statistics for the remaining clustered-search are limited to recall and precision values. (See Table 2.)

Clustering Method Parameter	Document Clusters Generated by Dattola's Algorithm	Document Clusters Generated using Query-Clusters and Relevance Judgments
Number of clusters	21	11
Number of documents in largest cluster	124	160
Percent of collection in largest cluster	29	38
Number of documents in smallest cluster	25	52
Percent of collection in smallest cluster	6	12
Number of documents in average cluster	81	119
Percent of collection in average cluster	19	28
Percent overlap of clusters	18.5	(see text)
Number of concepts in average clusters	374	127

Statistics of Clustered Collections

Table 1

6. Results

As the graph in Appendix B indicates, the query-clustered search results in recall/precision values rivalling a full search, and surpassing it at one point, up to a recall level of .4000. The search with normal clusters setting n=2 passes the query-cluster graph at recall .3000 and remains close to the full search graph from that point on. The standard clusters with n=1 generate values quite a bit lower than the others.

A preliminary observation is that these results follow directly the correlation percentages of Table 2: The higher the CP, the better the results on the Appendix B graph. Of course, this relationship is not linear, as the full search is only slightly better than both the query-cluster search and the standard-cluster search with n=2; the full search has however a CP nearly three times the size of the others. Obviously, other factors are involved here.

It is suggested that, with "good" enough clusters and centroids, a clustered search need not loose a great deal of the recall compared with a full search. Notice in Appendix A that the n=1 normal-cluster search has a very low aim value, completely cancelling out the high target value. Thus, although for most queries there is a cluster which "suits" it very well, that cluster is seldom found in the search. The problem might be in the construction of the centroid. On the other hand, the query-clustered documents maintain both high aim and target values, and achieve markedly better results. Of course, these differences are not independent of the correlation percentage. Yet, it is a matter of question whether document clusters may be constructed with high aim and target values, and at the same time low correlation percentages. For several queries, it appears that normal document clus-

Parameter Search Method	Normalized Recall	Normalized Precision	Rank Recall	Log Precision	Average Correlation Percentage
Full Search	0.8258	0.5968	0.1920	0.4327	100
Clustered Search using query clusters	0.5538	0.4500	0.0621	0.3328	31.2
Clustered Search using Dattola's algorithm. n=1	0.3378	0.3040	0.0179	0.2712	21.1
Clustered Search using Dattola's algorithm. n=2	0.6072	0.4893	0.1034	0.3665	38.8

Results of Four Searches

n = number of clusters expanded

Table 2

tering is inferior to query-clustering, even with similar correlation percentages. (Queries 6,8,20,24,25.) On the other hand, other queries show the opposite trend. (Queries 7,9,22.) Additional results are needed, particularly of query-clustering methods generating relatively small clusters. Until such tests are carried out, the present results must remain inconclusive.

References

[1] V. R. Lesser, A Modified Two-Level Search Algorithm Using Request Clustering, Report No. ISR-11 to the National Science Foundation, Section VII, Department of Computer Science, Cornell University, June 1966.

[2] W. Riddle, T. Horwitz, and R. Dietz, Relevance Feedback in an Information Retrieval System, Report No. ISR-11 to the National Science Foundation, Section VI, Department of Computer Science, Cornell University, June 1966.

[3] J. D. Broffitt, H. L. Morgan, and J. V. Soden, On Some Clustering Techniques for Information Retrieval, Report No. ISR-11 to the National Science Foundation, Section IX, Department of Computer Science, Cornell University, June 1966.

[4] J. J. Rocchio, Jr., Document Retrieval Systems — Optimization and Evaluation, Report No. ISR-10 to the National Science Foundation, Harvard University Doctoral Thesis, March 1966.

[5] G. Salton, Search Strategy and the Optimization of Retrieval Effectiveness, Report No. ISR-12 to the National Science Foundation, Section V, Department of Computer Science, Cornell University, June 1967.

[6] R. T. Grauer and M. Messier, An Evaluation of Rocchio's Clustering Algorithm, Report NO. ISR-12 to the National Science Foundation, Section VI, Department of Computer Science, Cornell University, June 1967.

[7] R. T. Dattola, A Fast Algorithm for Automatic Classification, Report No. ISR-14 to the National Science Foundation, Section V, Department of Computer Science, Cornell University, October 1968.

[8] E. Ide, New Experiments in Relevance Feedback, Report No. ISR-14 to the National Science Foundation, Section VIII, Department of Computer Science, Cornell University, October 1968.

Appendix A

Aim, Target, and Rejection Values, by Query

Query Number	Number of Relevant Documents	Target Clusters			Aim Clusters			Aim to Target Ratio	Rejection
		No. of Relevant Docs. in Cluster	Corre-lation Percen-tage*	Target Value	No. of Relevant Docs. in Cluster	Corre-lation Percen-tage*	Aim Value		
1	3	3	29.2	1.0000	3	29.2	1.0000	1.0000	0.1765
2	10	7	40.3	0.7000	7	40.3	0.7000	1.0000	0.3889
3	10	10	40.3	1.0000	10	40.3	1.0000	1.0000	0.2439
4	10	10	29.2	1.0000	10	31.4	1.0000	1.0000	0.1282
5	6	4	34.9	0.6667	3	29.2	0.5000	0.7500	0.1905
6	5	4	35.4	0.8000	3	31.4	0.6000	0.7500	0.2000
7	4	4	34.9	1.0000	1	30.9	0.2500	0.2500	0.2352
8	5	5	40.3	1.0000	5	40.3	1.0000	1.0000	0.2941
9	3	2	31.4	0.6667	0	30.9	0.0000	0.0000	0.2500
10	6	5	34.9	0.8333	5	34.9	0.8333	1.0000	0.3571
11	5	4	35.1	0.8000	3	40.6	0.6000	0.7500	0.2500
12	6	6	21.0	1.0000	0	40.6	0.0000	0.0000	1.0000
13	4	4	40.3	1.0000	4	40.6	1.0000	1.0000	0.5000

*See text for explanation of these values.

a) Documents Clustered by Query Clusters (Queries 1-13)

Number of Clusters expanded = 1

		Target Clusters			Aim Clusters				
Query Number	Number of Relevant Documents	No. of Relevant Docs. in Cluster	Correlation Percentage*	Target Value	No. of Relevant Docs. in Cluster	Correlation Percentage*	Aim Value	Aim to Target Ratio	Rejection
14	9	9	31.4	1.0000	5	21.0	0.5556	0.5556	0.4286
15	12	12	14.9	1.0000	10	30.9	0.8333	0.8333	0.2308
16	6	6	24.1	1.0000	6	24.1	1.0000	1.0000	0.5000
17	5	3	34.9	0.6000	0	30.9	0.0000	0.0000	0.2727
18	7	6	40.2	0.8571	3	40.6	0.4268	0.5000	0.3529
19	3	3	35.4	1.0000	0	21.0	0.0000	0.0000	0.5000
20	4	3	24.1	0.7500	3	24.1	0.7500	1.0000	0.6000
21	14	14	35.4	1.0000	14	35.4	1.0000	1.0000	0.3684
22	5	5	24.1	1.0000	1	21.0	0.2000	0.2000	0.6250
23	8	5	21.0	0.6250	2	34.9	0.2500	0.4000	0.2174
24	7	7	14.9	1.0000	7	14.9	1.0000	1.0000	0.5000
25	12	12	29.2	1.0000	12	21.0	1.0000	1.0000	0.2791
Averages	—	—	31.1	0.8920	—	31.2	0.6200	0.6796	0.3636

* See text for explanation of these values.

b) Documents Clustered by Query Clusters (Queries 14-25)

Number of Clusters expanded = 1

		Target Clusters			Aim Clusters				
Query Number	Number of Relevant Documents	No. of Relevant Docs. in Cluster	Correlation Percentage*	Target Value	No. of Relevant Docs. in Cluster	Correlation Percentage*	Aim Value	Aim to Target Ratio	Rejection
1	3	3	18.9	1.0000	0	13.7	0.0000	0.0000	0.1667
2	10	9	20.8	0.9000	1	14.7	0.1000	0.1111	0.4286
3	10	5	29.7	0.5000	5	29.7	0.5000	1.0000	0.1250
4	10	8	34.2	0.8000	3	13.7	0.3000	0.3750	0.1143
5	6	5	19.6	0.8333	1	13.7	0.1667	0.2000	0.1389
6	5	4	28.8	0.8000	2	32.6	0.4000	0.5000	0.1739
7	4	3	19.6	0.7500	3	29.7	0.7500	1.0000	0.1304
8	5	4	7.1	0.8000	0	35.7	0.0000	0.0000	0.3333
9	3	1	10.9	0.3333	1	35.7	0.3333	1.0000	0.2000
10	6	5	14.7	0.8333	5	14.7	0.8333	1.0000	0.6250
11	5	5	14.7	1.0000	5	14.7	1.0000	1.0000	0.8333
12	6	4	35.7	0.6667	2	17.5	0.3333	0.5000	0.4000
13	4	2	24.8	0.5000	0	17.5	0.0000	0.0000	0.3333

*See text for explanation of these values.

c) Documents Clustered by Dattola's Algorithm (Queries 1-13)

Number of Clusters expanded = 1

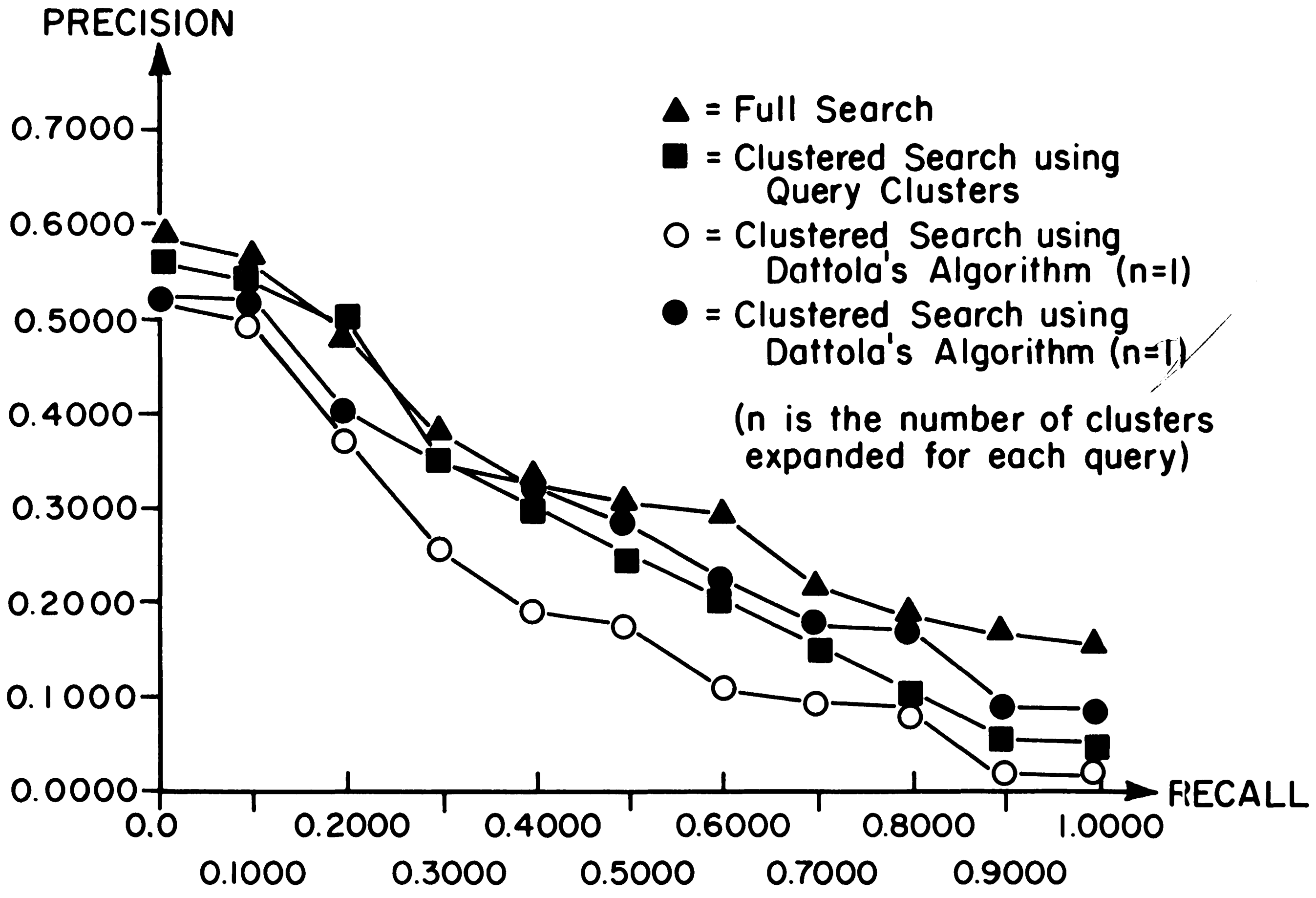
		Target Clusters			Aim Clusters				
Query Number	Number of Relevant Documents	No. of Relevant Docs. in Cluster	Corre-lation percen-tage*	Target Value	No. of Relevant Docs. in Cluster	Corre-lation Percen-tage*	Aim Value	Aim to Target Ratio	Rejection
14	9	9	35.7	1.0000	0	10.9	0.0000	0.0000	0.5294
15	12	7	17.5	0.5833	7	17.5	0.5833	1.0000	0.5000
16	6	3	32.6	0.5000	2	17.5	0.3333	0.6667	0.1667
17	5	4	21.0	0.8000	1	13.7	0.2000	0.2500	0.1212
18	7	3	21.0	0.4286	3	35.7	0.4286	1.0000	0.1364
19	3	3	22.6	1.0000	0	35.7	0.0000	0.0000	0.1875
20	4	2	7.1	0.5000	1	28.5	0.2500	0.5000	0.2857
21	14	7	32.6	0.5000	3	10.9	0.2143	0.4286	0.1489
22	5	4	17.5	0.8000	4	17.5	0.8000	1.0000	0.2500
23	8	6	28.5	0.7500	2	10.9	0.2500	0.3333	0.2000
24	7	5	17.5	0.7143	5	17.5	0.7143	1.0000	0.4167
25	12	9	26.4	0.7500	9	26.4	0.7500	1.0000	0.1323
Averages	—	—	22.4	0.7217	—	21.1	0.3696	0.5546	0.2831

*See text for explanation of these values.

d) Documents Clustered by Dattola's Algorithm (Queries 14-25)

Number of Clusters expanded = 1

Appendix B



Recall-Level Averages for Clustered Search