Analysis of Documents Clustering Using Sampled Agglomerative Technique

Omar H. Karam, Ahmed M. Hamad, and Sherin M. Moussa

***Abstract*—In this paper a clustering algorithm for documents is proposed that adapts a sampling-based pruning strategy to simplify hierarchical clustering. The algorithm can be applied to any text documents data set whose entries can be embedded in a high dimensional Euclidean space in which every document is a vector of real numbers. This paper presents the results of an experimental study of the proposed document clustering technique. The performance of the method is illustrated in terms of quality of clusters.**

***Index Terms*— Agglomerative clustering, Document clustering, and k-means clustering.**

I. Introduction

Document Clustering has been extensively investigated as a methodology for improving document search and retrieval. The general assumption is that mutually similar documents will tend to be relevant to the same queries, and hence, that automatic determination of groups of such documents can improve recall and precision by effectively broadening a search request [10].

Many techniques are used for document clustering [1],[5],[6],[7]. The two main approaches used for document clustering are the agglomerative hierarchical and the k-means clustering techniques. k-means is a partitional clustering technique based on the idea that a center point (centroid), representing the average point of the data points grouped in one cluster, can represent a cluster. Given the number of required clusters (k) and the initial k center points (centroids), documents can be clustered by assigning each document to the cluster having the nearest centroid. This is done by measuring similarity through the distance function between the document and each of the k centroids. A new centroid is then re-computed for each newly created cluster, and the whole process is repeated until no change in the k centroids or the detected change is less than a determined threshold.

On the other hand, the agglomerative clustering algorithm is a hierarchical technique that produces a nested sequence of partitions, with a single, all-inclusive cluster at the top and singleton clusters of individual points at the bottom [3],[9]. It starts with each document as an individual cluster, and at each iteration, merges the most similar or closest pair of clusters until one cluster is obtained. Therefore, clustering should stop before obtaining the single cluster depending on a specified condition, such as the number of clusters required, or the quality of the clusters to obtain.

Agglomerative hierarchical clustering is often portrayed as "better" than K-means, although slower. It is also superior to k-means clustering, especially for building document hierarchies. [1],[2],[3],[8]. The traditional agglomerative hierarchical clustering procedure is summarized as follows [3]:

1. Compute the similarity between all pairs of clusters, i.e., calculate a similarity matrix whose ij_{th} entry gives the similarity between the i_{th} and j_{th} clusters.
2. Merge the most similar (closest) two clusters.
3. Update the similarity matrix to reflect the pairwise similarity between the new cluster and the original clusters.
4. Repeat steps 2 and 3 until only a single cluster remains.

There are different hierarchical clustering algorithms [1],[3],[8]. The only real difference between these different hierarchical schemes is how they choose which clusters to merge, i.e., how they choose to define cluster similarity.(i.e. Intra-Cluster Similarity Technique (IST), Centroid Similarity Technique (CST), and Unweighted Pair Group Method with Arithmetic mean (UPGMA)).

Among the several obstacles to be faced in text mining, the amount of preprocessing that has to be applied to a document for text mining purposes, and the documents mathematical representation [4].

In this paper, an agglomerative technique is applied to document clustering that adapts a sampling-based pruning strategy to simplify hierarchical clustering. The algorithm can be applied to any text documents data set whose entries can be embedded in a high dimensional Euclidean space (i.e. in which every document is a vector of real numbers). The proposed algorithm is followed by an experimental study and the obtained results while varying its different parameters and observing their effect on the resulting cluster qualities.

Omar H. Karam, Department of Information Systems, Faculty of Computer and Information Sciences; Ain Shams University, Cairo, 11566, EgyptPhone: (202) 684 4284 Fax: (202) 682 8298 e-mail: ohkaram@hotmail.com).

Ahmed M. Hamad, Professor of Computer Systems, Department of Information Systems, Faculty of Computer and Information Sciences, Ain Shams University, Cairo, 11566, Egypt Phone: (202) 684 4284 Fax: (202) 682 8298 (e-mail: amhamad13@yahoo.com).

Sherin M. Moussa, Department of Information Systems, Faculty of Computer and Information Sciences Ain Shams University, Cairo, 11566, Egypt Phone: (202) 684 4284 Fax: (202) 682 8298 (e-mail: shero@usa.com).

This paper is organized as follows. Section 2 discusses the preparation of text documents in order to be clustered. Section 3 explains the method used in our system for document clustering, and section 4 presents the results of experiments evaluating the performance of the system. Finally, section 5 concludes the paper.

II. Documents Preparation

In order to apply any clustering algorithm on text documents, some steps should be followed [2],[3],[4]. These are:

a) Preprocessing

Initially, Very common words, (i.e. prepositions and non-content bearing words, often known as stop words), are removed completely from documents. In addition, stemming is applied where different forms of a word are reduced to a single stem form. In our work, Porter's suffix stripping algorithm is used [11]. Now documents are ready to be mathematically represented.

b) Mathematical Representation

In our system, the vector space model is used, where each document is represented by the (TF) vector, **d**, in the multidimensional space of document words. The Term-Frequency vector is:

$$\mathbf{d}_{tf} = (tf_1, tf_2, \cdots, tf_n) \qquad (1)$$

where tf_i is the frequency of the ith term in the document. Given a set, S, of documents and their corresponding vector representations, the centroid vector, **c**, is defined to be:

$$\mathbf{c} = \frac{1}{|S|}\sum_{\mathbf{d}\in S}\mathbf{d} \qquad (2)$$

which is obtained by averaging the weights of the various terms present in the documents of S, where **d** is the document vector obtained from the set S. Thus, the vector space model of a text data set is considered to be a word-by-document matrix whose rows are words and columns are document vectors.

c) Dimension Space Construction

Since the document vectors are of high dimensionality, pruning can be used to reduce the dimensionality. A threshold P is given, and words having a frequency less than this threshold are removed completely from the document vectors.

At this stage, document vectors lengths are not unified. Therefore, a superset vector is created that contains all words in all the documents after pruning. This superset vector is then compared to each document vector, and each document vector is mapped from its own space to the superset vector space. If a word found in the superset vector is missing in the document vector, it is added to the document vector with term frequency zero. Thus a dimension space is constructed, where each word in a document vector is considered to be a dimension.

d) Similarity Measure:

Document vectors are now ready for clustering, where similar document vectors are grouped together in the same cluster. The similarity measure between any two vectors is computed using the Minkowski distance function,

$$d(i,j) = (|x_{i1} - x_{j1}|^q + |x_{i2} - x_{j2}|^q + \cdots + |x_{ip} - x_{jp}|^q)^{1/q} \qquad (3)$$

where i represents the ith document vector and j represents the jth document vector, x_{ip} is the word number p in the ith document vector; x_{jp} is the word number p in the jth document vector, q is the power of the distance function. This similarity measure can be one of the following approaches: a) Single linkage, b) Complete linkage, and c) Centroids similarity [8]. In this work, the third approach is used, where similarity between two clusters is measured between the centroids of each. The centroid of a cluster is its average element.

III. The Proposed Clustering Algorithm

The proposed clustering algorithm reads the whole collection of documents to create the corresponding document vectors, which are pruned by the given threshold, P. The dimension space is then constructed as explained, and a random sample, R, is chosen from the data set. Agglomerative clustering is then applied to this chosen sample using the Centroid Similarity Technique (CST), where at each successive iteration, pairs of clusters having their similarity measurement value below an incrementing step value, are merged together into one cluster. At the next iteration, the incrementing step is increased and the procedure is repeated until the given agglomerative distance threshold, A, is exceeded. This agglomerative distance threshold at which the agglomeration stops, that is, the distance between clusters, is taken as the quality measure for the agglomerative clusters.

When agglomeration stops, the number of the created clusters is considered to be the number of the clusters that will contain all the document vectors of the whole data set, and the computed centroids of the created clusters are the initial centroids for the document vectors of the whole data set. Individual documents that have not been merged yet with other clusters are considered to be individual clusters, having the individual document vector as the centroid vector of the considered cluster.

Hence, the first run of k-means algorithm is applied, where each document vector of the whole data set is assigned to the cluster having the nearest or the most similar centroid vector. The quality of an obtained cluster for these documents assignment is taken as the intra-cluster similarity measure, that is, the sum of the distances between each document vector and

the centroid vector of the cluster to which this document is attached. This value is then normalized by dividing it by the number of documents assigned to the considered cluster. The total quality value is computed by summing up the quality values of all the created clusters, which is normalized by further dividing this total value by the number of the created clusters. That is, cluster quality equals:

$$\text{Cluster Quality} = \frac{1}{k}\left[\sum_{j=1}^{k}\frac{1}{n}\left[\sum_{i=1}^{n}(\mathbf{d}_{ij}-\mathbf{c}_j)\right]\right] \qquad (4)$$

where $\mathbf{d}_{ij}$ is the ith document vector in the data set and belongs to the jth cluster, $\mathbf{c}_j$ is the centroid vector of the jth cluster, k is the number of the required clusters and n is the number of documents in the whole data set. The complexity of the proposed clustering algorithm is $O(m^2 + kn)$, where m is the number of chosen documents in the random sample, k is the number of created clusters, and n is the number of documents in the whole data set.

IV. Experiments and Results

The proposed clustering algorithm is studied by examining it on the NSF (National Science Foundation) data set obtained by downloading 725 abstracts of the grants awarded by the NSF. These abstracts included subjects Such as astronomy, population studies, undergraduate education, materials, mathematics, biology, health, oceanography, computer science and chemistry. Different pruning thresholds are applied on the data set with P = 2, 3, 4, 5, 6, 7 where most of the document vectors contained no words at P = 7.

After creating the corresponding superset vector and constructing the dimension space, samples of R = 10%, 20% were chosen to run the agglomerative clustering approach on these sample document vectors. At this stage, the Euclidean distance function (q = 2) is used to compute distances between the vectors. Different agglomerative distance thresholds are also applied, A = 5, 10, 15, and 20 where the number of the created clusters at A = 20 was only two. The effect of varying these parameters on the number of the created agglomerative clusters and the quality of the final resultant clusters are monitored. Figure (1) represents a graph showing the sizes of the created matrices of the dimension spaces constructed for the chosen sample at R = 10% and 20% and for the whole data set. The rows represent the number of contained words (obtained from the superset vector) and the columns represent the number of considered document vectors. It is clear that as the pruning threshold value (P) increases, the matrices size decreases. At higher pruning thresholds, the matrices size are almost similar for both sample sizes R = 10% and 20%.

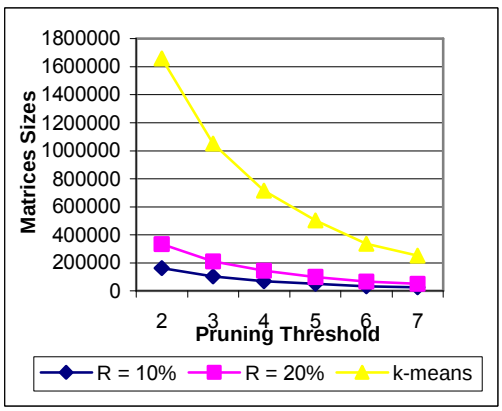

Fig. 1 Matrices Sizes for Constructed Dimension Spaces

TABLE I
K value At Different Agglomerative Thresholds with Sample Size = 10%

Pruning Threshold	A = 5	A = 10	A = 15	A = 20
2	69	47		
3	67	29	16	
4	61	26	12	3
5	51	23	9	2
6	39	11	7	2
7	27	19	5	2

TABLE 2
K value At Different Agglomerative Thresholds with Sample Size =20%

Pruning Threshold	A = 5	A = 10	A = 15
2	137		
3	117	64	29
4	108	47	26
5	84	39	20
6	69	28	15
7	48	20	14

TABLE 3
K value At Different Pruning Thresholds with Sample Size = 10%

TABLE 4

Agglomerative Threshold	P = 2	P = 3	P = 4	P = 5	P = 6	P = 7
5	69	67	61	51	39	27
10	47	29	26	23	11	19
15		16	12	9	7	5
20			3	2	2	2

K value At Different Pruning Thresholds with Sample Size = 20%

Agglomerative Threshold	P = 3	P = 4	P = 5	P = 6	P = 7
5	117	108	84	69	48
10	64	47	39	28	20
15	29	26	20	15	14

Tables (1) and (2) shows k value at different agglomerative thresholds A, for R = 10% and 20% respectively. The more the pruning threshold P increases, in other words, the more the document vectors are reduced, the less the number of created agglomerative clusters obtained (k value). On the other hand, tables (3) and (4) shows k value at different pruning thresholds P for R = 10% and 20% respectively. It was found that the more the agglomerative distance threshold A increases, the less the number of created agglomerative clusters obtained (k value).

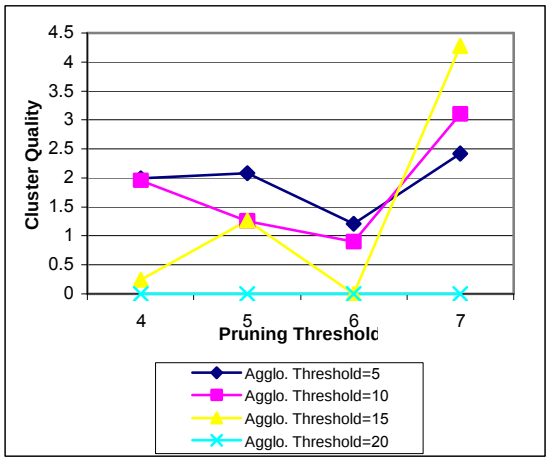

Fig. 2 Pruning Threshold Effect on Cluster Quality At Different Agglomerative Thresholds with Sample Size = 10%

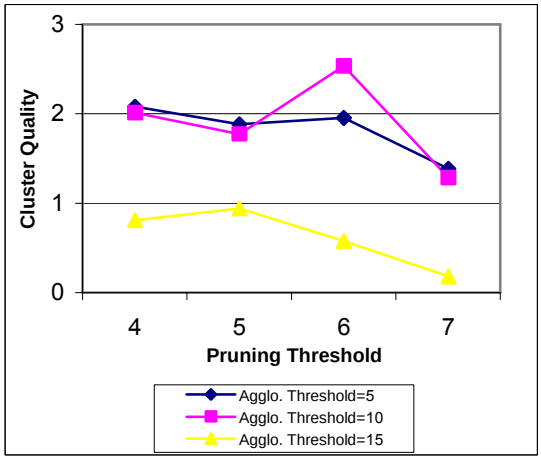

Fig. 3 Pruning Threshold Effect on Cluster Quality At Different Agglomerative Thresholds with Sample Size = 20%

By studying the pruning threshold effect on the cluster quality of the resultant clusters from our proposed algorithm as shown in figures (2) and (3) for R = 10% and 20% respectively and for different values of A, it was found that as the pruning threshold P increases, the resultant cluster quality increases except for the highest pruning threshold P = 7 where the resultant cluster quality decreased again. This is due to the removal of many words from the document vectors that distinguish documents from one another.

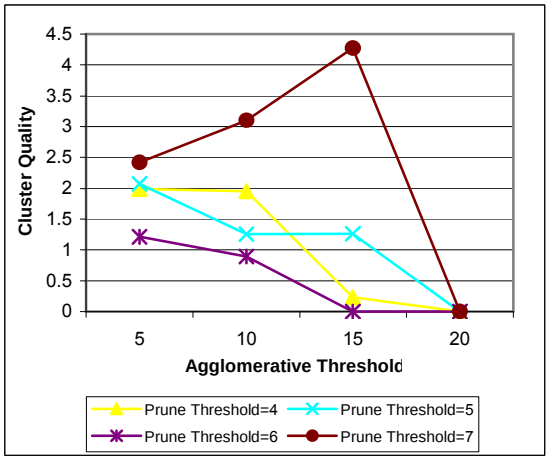

Fig. 4 Agglomerative Threshold Effect on Cluster Quality At Different Pruning Thresholds with Sample Size = 10%

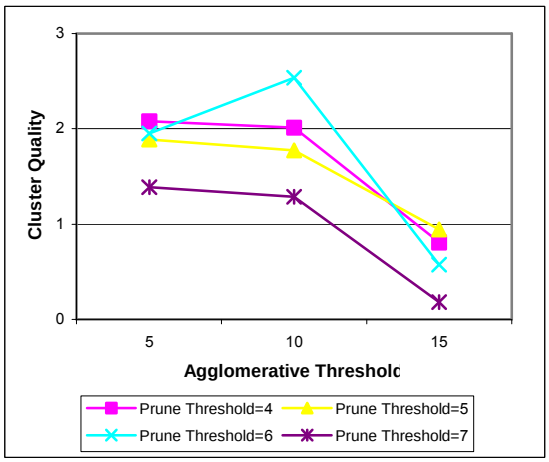

Fig. 5 Agglomerative Threshold Effect on Cluster Quality At Different Pruning Thresholds with Sample Size = 20%

As for the effect of the agglomerative distance threshold, A, on the resultant cluster quality for R = 10% and 20% respectively and for different P values as shown in figures (4) and (5), it was found that the more the agglomerative distance threshold A increases, the higher is the cluster quality of the resultant clusters.

After studying the effect of both pruning and agglomerative distance thresholds on the resultant cluster quality, it can be summarized that at the same value of the agglomerative threshold, A, the variation in the resultant cluster quality at the different pruning thresholds, P is high. But at the same value of the pruning threshold, P, the variation in the resultant cluster quality at the different agglomerative thresholds, A, is low. This is valid for both the sample sizes, R = 10%, 20%.

V. CONCLUSION

In this paper, we applied an agglomerative technique to document clustering that adapts a sampling-based pruning strategy to simplify hierarchical clustering depending on the required quality for the obtained clusters. The proposed algorithm combines the agglomerative hierarchical approach with the first run of k-means approach to provide k disjoint clusters. An experimental study was conducted to evaluate the different parameters affecting the resultant cluster quality of the obtained clusters specifically the pruning threshold, the agglomerative threshold for the sample sizes of 10% and 20%. It was shown that the variation in the resultant cluster quality at the different pruning thresholds is high, versus the variation in the resultant cluster quality at the different agglomerative thresholds, thus pruning threshold effect is dominant.

REFERENCES

[1] [1] J. Han, M. Kamber. *Data Mining Concepts and Techniques.* Morgan Kaufmann Publishers. Pages 335 – 388 and 428 – 435, 2001.

[2] [2] I. S. Dhillon and D. S. Modha. *Concept Decompositions for Large Sparse Text Data using Clustering*. Technical Report RJ 10147, IBM Almadan Research center, 2000.

[3] [3] M. Steinbach, G. Karypis, ans V. Kumar. *A Comparison of Document Clustering Techniques*. In KDD workshop on Text Mining, 2000.

[4] [4] J.L. Neto, A.D. Santos, C.A.A. Kaestner, and A.A. Freitas. *Document Clustering and Text Summarization.* In Proceedings, 4th International Conference on Practical Applications of Knowledge Discovery and Data Mining (PADD-2000), 41-55. London: The Practical Application Company, 2000.

[5] [5] N. Turenne and F. Rousselot. *Evaluation of Four Clustering Methods used in Text Mining*, 1999.

[6] [6] M. Meila, and D. Hackerman, *An Experimental Comparison of Several Clustering and Initialization Methods,* (Technical Report 98-06). Microsoft Research Redmond, WA, 1998.

[7] [7] C. Wei, Y. Lee and C. Hsu. *Empirical Comparison of Fast Clustering Algorithms for Large Data Sets*. Proceedings of the 33rd Hawaii International Conference on System Sciences, 1998.

[8] [8] M. J. A. Berry, G. Linoff. *Data Mining Techniques for Marketing, Sales, and Customer Support.* John Wiley & Sons. Pages 187 – 215, 1997.

[9] [9] D. Fisher. *Iterative optimization and simplification of hierarchical clusterings*. J. Art. Intell. Res., 4:147--179, 1996.

[10] [10] D. Cutting, D. Karger, J. Pedersen, and J. Tukey. *Scatter/gather: a cluster-based approach to browsing large document collections*. In 15th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'92), pages 318--329, 1992.

[11] [11] M.F. Porter, *An Algorithm for Suffix Stripping*, Program 14 (3), Pages 130-137, July 1980.