Journal of Pattern Recognition Research 1 (2013) 66-84

Received November 1, 2011. Revised August 7, 2013. Accepted August 7, 2013.

CUES: A New Hierarchical Approach for Document Clustering

Tanmay Basu *mailtanmaybasu@gmail.com*
Machine Intelligence Unit
Indian Statistical Institute
203, B. T. Road. Kolkata-700108. India

C. A. Murthy *murthy@isical.ac.in*
Machine Intelligence Unit
Indian Statistical Institute
203, B. T. Road. Kolkata-700108. India

Abstract

Objective of the document clustering techniques is to assemble similar documents and segregate dissimilar documents. Unlike document classification, no labeled documents are provided in document clustering. One of the main challenges of any document clustering algorithm is the selection of a good similarity measure. Traditionally, using the vector space model, the number of words common between two documents is used for determining their similarity. This paper introduces a document similarity measure, *extensive similarity* between the documents. In this approach two documents are considered to be similar if they share a minimum number of common words and they have almost same distance with every other document in the corpus i.e., both are either similar or dissimilar to the other documents. A hierarchical document clustering algorithm, using extensive similarity between the documents is proposed in this article. It is experimentally found on several text data sets that the proposed document clustering algorithm performs significantly better than the traditional document clustering techniques, comparisons for which are based on f-measure and normalized mutual information.

Keywords: Text Document Clustering, Document Similarity, Text Mining, Pattern Recognition

1. Introduction

The objective of conventional document clustering is automatic grouping of documents so that the documents within a cluster are very similar, but dissimilar to the documents in other clusters. When applied to text data, clustering algorithms try to identify inherent grouping of the documents to produce good quality clusters. Document clustering algorithms are generally unsupervised learning techniques. They are totally different from supervised learning methods like document classification. In document classification a set of labeled documents is provided to train the classifier. The performance of the classifier is dependent on the quality of the labeled samples. But document clustering categorizes the documents without using any labeled samples. Hence the quality of the document clustering techniques is mainly dependent upon how they are finding similarity between two documents. In most of the document clustering techniques, cosine value between the document vectors is used as the similarity measure which is based on the number of common words present in the documents. If two documents contain many common words then it is likely that the documents are very similar. But there are no crisp explanation that how many common words can identify the similarity between documents. It has been recognized that partitional clustering techniques (e.g., k-means) are well suited for clustering a large

document corpus due to their low computational complexity [2]. But partitional clustering algorithms need the knowledge of total number of clusters initially, which is generally not available for a sparse document corpus with high dimensionality [13]. Hierarchical clustering algorithms can produce good quality clusters, but it need to know the stopping criterion. It is not very easy to predict a good stopping criterion for hierarchical document clustering.

A similarity measure is proposed to identify the similarity between the documents and it is named as *extensive similarity*. The proposed *extensive similarity* between documents is described in section 3 of this article. The main idea is as follows - two documents, say a and b will be similar if they have sufficient content similarity and they have almost same behavior with the other documents in the corpus i.e., a is similar to b and a is similar to c but b is dissimilar to c; simultaneously these three states should not happen. Initially a threshold is set on the cosine similarity of the document vectors. But two documents with some content similarity should not be in the same cluster always, both of the documents should have almost same kind of similarities with the other documents as well. So a score is assigned to each pair of documents depending on their content similarity and their distances with every other documents in the corpus. Hence the similarity is named as *extensive similarity*. If the content similarity between two documents is very low (i.e., less than a threshold) then the extensive similarity of the documents will be negative. Two documents with negative extensive similarity implies that the documents are dissimilar. The proposed hierarchical document clustering technique - *Clustering Using Extensive Similarity*(CUES) is introduced using the extensive similarity between documents. Intuitively the idea of CUES is, two documents with high extensive similarity between them should be in the same cluster. It will initially treat every document as a cluster. Then the algorithm will merge two clusters which have a minimum cluster distance and again finds two minimum distant clusters and will merge them and so on. The distance between two clusters is determined from the extensive similarity between the documents of the clusters. The cluster distance will be negative if the extensive similarity between the documents (taking one from each cluster) is negative. Two clusters with negative cluster distance will never be merged by CUES. CUES will stop merging the clusters when the distance between every two clusters is negative. Thus CUES determines the number of clusters automatically. The main contributions in this article are, a new document similarity measure (extensive similarity) and an agglomerative hierarchical document clustering technique using a new cluster distance measure which can identify two dissimilar clusters. The performance of CUES is compared with several partitional and hierarchical clustering algorithms and two types of spectral clustering methods using various well known text data sets in the experimental evaluation. The analysis shows that CUES performed significantly better than the other methods.

The paper is organized as follows - Section 2 describes various document clustering techniques with their pros and cons and some related works.

Section 3 explains the proposed extensive similarity between documents and the document clustering method CUES. The experimental results are described in Section 4, with a detailed discussion on cluster validity measures.

2. Document Clustering Methods

Document clustering methods partition a set of documents into clusters such that the documents in the same cluster are more similar to each other than documents in different clusters according to some similarity or dissimilarity measure. A pairwise document similarity measure plays the most significant role in any document clustering technique. Any document

clustering algorithm first finds the document similarity and then groups similar documents into a cluster. Numerous document similarity measures have been proposed, most of which treat each document as a set of words [3], often with frequency information. Each document is represented by a vector whose length is equal to the number of unique words of the corpus. The vector is often sparse as most of the terms do not occur in a particular document. In one of the representations, each component of the vector has a value equal to the number of occurrences of the word in that particular document. A popular similarity measure is cosine similarity which computes the cosine of the angle between two document vectors.

There are two basic types of document clustering techniques available in the literature - *hierarchical* and *partitional* clustering techniques [1].

Hierarchical clustering produces a hierarchical tree of clusters [6]. Each individual level can be viewed as a combination of clusters in the next lower level. This hierarchical tree structure is also known as dendrogram [11]. The hierarchical clustering techniques can be divided into two parts - *agglomerative* and *divisive*. In an agglomerative hierarchical clustering (AHC) method [2], starting with each data point as individual cluster, at each step, it merges the most similar clusters until a given termination condition is satisfied. In a divisive method, starting with the whole set of data points as a single cluster, the method splits a cluster into smaller clusters at each step until a given termination condition is satisfied.

Several stopping criteria for AHC algorithms have been proposed [14]. In principle these algorithms are very sensitive to the stopping criteria, but practically there is no widely acceptable stopping criterion. Hence multiple good clusters may be merged by the AHC method, which will be eventually meaningless to the user.

In *single-link* method the similarity between a pair of clusters is calculated as the similarity between the two most similar documents where each document represents each individual cluster. The *complete-link* method measures the similarity between a pair of clusters as the least similar documents, one of which is in each cluster. The *group average* method merges two clusters if they have least average similarity than the other clusters. Average similarity means the average of the similarities between the documents of each cluster. In a *divisive hierarchical clustering* technique, initially, the method assumes the whole data set as a single cluster.

Then at each step, the method chooses one of the existing clusters and splits it into two. The process continues till only singleton clusters remain or it reaches a given halting criterion. Generally the cluster with the least overall similarity is chosen for splitting [2].

In contrast to hierarchical clustering techniques, partitional clustering techniques allocate data into a previously known fixed number of clusters. The commonly used partitional clustering technique is *k-means* algorithm [12], where k is the desired number of clusters. Here initially k seed points are chosen from the data set randomly. Then each data point is assigned to the nearest center. This will continue until the clustering does not change, or the procedure will run for a fixed number of iterations. The partitional algorithms, like k-means are advantageous due to their low computational complexity. Generally it takes linear time to build the clusters. But sometimes it suffers from high computational cost (due to repeated iteration for convergence to a solution) when the data set size is huge or the dimensionality of the data set is very high [13]. The main disadvantage of this method is that the number of clusters is fixed and it is very difficult to select a valid k for an unknown data set.

Also there is no proper way of choosing the initial seed points. The method is sensitive to the initial seeds and may get stuck in the local optima [12]. An improper choice of seed points may lead to clusters of poor quality.

Buckshot [3] is a combination of basic k-means and hierarchical clustering method. It tries to improve the performance of k-means algorithm by choosing better initial seed points. Initially it randomly selects $\sqrt{kN}$ (N is the number of documents in the corpus) documents from the data set as sample documents and performs AHC on these sample documents. The centroids of the k clusters on the sample documents are the initial seeds for the whole collection. The basic k-means algorithm with these seed points is applied to partition the whole document set. Repeated calls to this algorithm may produce different partitions. If the initial random sampling does not represent the whole dataset properly, the resulting clusters will be of poor quality. Note that appropriate value of k is necessary for this method too.

Bisecting k-means [2] is a variation of basic k-means algorithm. This algorithm tries to improve the quality of clusters in comparison to k-means clusters. In each iteration, it selects the largest existing cluster (the whole data set in the first iteration) and divides it into two subsets using k-means (k=2) algorithm.

This process is continued till k clusters are formed. It sometimes found to perform better than the basic k-means because it produces almost uniform sized clusters. But this method also faces difficulties like k-means, in choosing the initial centroids and a proper value of the parameter k.

The *k-Nearest Neighbor* (kNN) technique is mostly known to be used for classification [15], it has also been used for clustering [16]. It utilizes the property of k nearest neighbors, i.e., a document should be put in the same cluster to which most of its k nearest neighbors belong. Merge the document d_1 and d_2 to form a cluster, if d_1 and d_2 share at least k nearest neighbors and d_1, d_2 are k-nearest neighbors of each other. The performance of the algorithm is highly dependent on the parameter k and choosing a proper value of k is difficult for text data sets.

Spectral clustering is a very popular clustering method which works on the similarity matrix rather than the original data matrix using the idea of graph cut. It uses the top eigenvectors of the similarity matrix derived from the similarity between points [28]. The basic idea is to construct a weighted graph from the initial data set where each node represents a pattern and each weighted edge represents the similarity between two patterns. The clustering problem is formulated as a graph cut problem in this methodology, which can be tackled by means of the spectral graph theory. The core of this theory is the eigenvalue decomposition of the Laplacian matrix of the weighted graph obtained from data [29]. Let $X = \{x_1, x_2, ..., x_N\}$ be the set of N points to cluster. Let S be the $N \times N$ similarity matrix where S_{ij} represents the similarity between the points x_i and x_j and $S_{ii} = 0$. Define D to be the diagonal matrix where $D_{ii} = \sum_{j=1}^{N} S_{ij}$. Then construct the Laplacian matrix L= D-A and compute the eigenvectors of L. The data set will be partitioned using $D^{-1/2}e_2$ where e_2 is the eigenvector corresponding to the second largest eigenvalue of L. The same process will be continued until k partitions are obtained. But experimentally it has been observed that using more eigenvectors and directly computing a k way partitioning is better than recursively partition the data into two partitions [27]. Another problem is to find a proper stopping criterion for a huge and sparse text data sets.

Ng. et. al. [28] proposed a spectral clustering algorithm which simultaneously partitions the Laplacian data matrix into k subsets using the k largest eigenvectors and they have used a gaussian kernel on the similarity matrix. The algorithm is as follows:

- Form the similarity matrix $S \in \mathbb{R}^{N \times N}$ by using a gaussian kernel, defined by

$$S_{ij} = exp(-\frac{\rho(x_i, x_j)}{2\sigma^2}),$$

  where $\rho(x_i, x_j)$ denotes the similarity between x_i and x_j and σ is the scaling parameter. Note that $S_{ii} = 0$.

- Compute the diagonal matrix D as described above.
- Construct the Laplacian matrix $L = D^{-1/2}SD^{-1/2}$.
- Find the k largest eigenvectors of L and construct the matrix $Z \in \mathbb{R}^{N \times k}$ with the eigenvectors as its column.
- Form the matrix Y by re-normalizing the rows of X to have unit length.
- Partition Y into k clusters by treating each row of Y as a point in $\mathbb{R}^k$ using k-means algorithm.
- Assign $x_i,\ i = 1, 2, ..., N$ to cluster j, if and only if the i^{th} row of Y is assigned to j.

The gaussian kernel was used to to get rid of the curse of dimensionality. The main difficulty of using a gaussian kernel is that, it is very sensitive to the parameter σ [30]. A wrong value of σ may highly degrade the quality of the clusters. It is extremely difficult to select a proper value of σ for a document collection, since the document data sets are generally sparse with high dimension. In the experiments the value of σ is set by search over values from 10 to 20 percent of the total range of the similarity values and the one that gives the tightest clusters is picked, as suggested by Ng. et. al. [28]. It should be noted that the method will also suffers from the disadvantages of the k-means method, discussed above. In the experimental evaluation we have shown the performance of this two types of spectral clustering method - the first one using the similarity matrix and the second one, by applying a gaussian kernel to the similarity matrix (proposed by Ng. et. al.).

Document clustering has been traditionally investigated as a means of improving the performance of search engines by pre-clustering the entire corpus [7]. But it can also be seen as a post retrieval document browsing technique [3]. Various document clustering algorithms are available in the literature which we have already discussed. There are some more document clustering algorithms like the one proposed by Hammouda et. al. [4]. They used a graph structure and a document index graph to represent documents and also proposed an incremental clustering algorithm by representing each cluster with a similarity histogram. Huang et. al. proposed a clustering method with active learning using Wikipedia [22]. They utilized Wikipedia to create a concept based representation of a text document with each concept associated to a Wikipedia article rather than words. Banerjee et. all [25] investigated a method of improving the accuracy of clustering short texts by enriching their representation with additional features from Wikipedia. Cao et. al. [31] proposed an extended vector space model with multiple vectors defined over spaces of entity names, types, name-type pairs, identifiers, and keywords for text searching and clustering. Oikonomakou

et. al. [32] have shown a comparative study of various document clustering approaches with their merits and demerits. Wang et. al. [33] proposed a new method for improving text clustering accuracy based on enriching short text representation with keyword expansion. Jing et. al. developed a new knowledge-based vector space model (VSM) for text clustering. In the new model, semantic relationships between terms (e.g., words or concepts) are included in representing text documents as a set of vectors [34].

3. Proposed Document Clustering Technique

All the document clustering algorithms discussed above are mainly based on similarity between two documents. Two documents will belong to the same cluster if they are dissimilar to the documents in other clusters and similar to the documents in that cluster. The distance between two documents is dependent on the number of common words present in the documents. But there are no crisp bounds on the content similarity. Most of the traditional clustering methods identify two documents as similar if their similarity is more than the similarity of other pairs. This may lead, sometimes, to a case where the similarity between two documents is a low value (i.e., the content similarity is a small value), but they are considered to be similar. In this article we try to define the distance between two documents after extensively checking their distances with every other document in the corpus.

3.1 Extensive Similarity between Documents

A new idea is introduced here to develop a measure for extensive similarity between two documents depending on their similarity with every other document. Two documents will be similar if they share a minimum number of common words (i.e., they have sufficient content similarity) and they have almost same distances with every other documents in the corpus (i.e., both are either similar or dissimilar to all the other documents). This intuition is expressed below using two concepts. Initially we shall give the definition of two documents being surely dissimilar with the help of the distance between two documents d_1, d_2 based on a threshold $\theta \in (0, 1)$ as

$$\mathrm{dis}(d_1, d_2) = \begin{cases} 1 & \text{if } cos(\vec{d_1}, \vec{d_2}) \leq \theta \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $\vec{d_1}$ and $\vec{d_2}$ are the vectors corresponding to the documents d_1, d_2. If the cosine of the angle between two document vectors is very low i.e., two documents have a very few number of words in common then the distance is 1 i.e., the documents are dissimilar. On the other hand, distance 0 indicates that there exists some similarity between documents d_1 and d_2 i.e., they share a minimum number of common words. θ is a threshold value providing meaning to the phrase *minimum number of common words*. The value of θ will be determined from the nature of the document corpus. A corpus dependent method for estimating the value of θ is discussed later.

We shall now find an expression for extensive similarity between two documents based on their distances with every other documents. Let, $l = \sum_{k=1}^{N} |\mathrm{dis}(d_1, d_k) - \mathrm{dis}(d_2, d_k)|$. The extensive similarity (ES) between documents d_1 and d_2 is defined as

$$\mathrm{ES}(d_1, d_2) = \begin{cases} N - l & \text{if } \mathrm{dis}(d_1, d_2) = 0 \\ -1 & \text{otherwise} \end{cases} \tag{2}$$

In this extensive similarity, two documents will be more similar than the others if they have sufficient content similarity and they have similar distances with the rest of the documents in the data set, i.e., they have a very high ES value. l indicates the number of documents where similarity with d_1 is not the same as the similarity with d_2. As the l value increases, the similarity between the documents d_1 and d_2 decreases. If $l = 0$ then the documents are totally similar. Actually l is a grade of dissimilarity and it indicates that any two documents d_1 and d_2 have different behavior with l number of documents. This l is used in the stopping criterion of the proposed hierarchical document clustering algorithm. Unlike other similarity measures, ES takes into account the distances of the said two documents d_1, d_2 with respect to every other document in the corpus when measuring the distance between them. The new document clustering algorithm CUES will be discussed later using this extensive similarity between documents.

3.2 Remarks

We shall discuss some typical cases below regarding the definition of *dis* (equation 1).

$\star$ If the cosine value between two documents is less than θ then we can strictly say that the documents are dissimilar as they are not sharing a minimum number of common words.

$\star$ If the cosine value between two documents d_1 and d_2 is very high (e.g., 0.85), then they share sufficiently many common words and hence the documents are very similar, and d_1, d_2 are likely to behave similarly with most of the other documents in the corpus. As a result, the ES value will be very high which indicates that d_1 and d_2 are similar.

$\star$ Now consider the situation where $\text{dis}(d_1, d_2) = 0$, but the number of common words between d_1 and d_2 is neither very high nor very low, i.e., $\theta \leq cos(d_1, d_2) \leq \theta'$, where θ' is another threshold. ES is very useful in this particular situation which occurs frequently. It will check the behavior of d_1 and d_2 with the other documents in the data set and will assign a grade (l) to the similarity between the documents. This l could group documents with a high extensive similarity which is introduced in our document clustering algorithm.

3.3 Properties of Extensive Similarity

Extensive similarity has some interesting properties which are as follows.

a) ES is symmetric. Forevery pair of documents d_1 and d_2, we have $\text{ES}(d_1, d_2) = \text{ES}(d_2, d_1)$.

b) If $d_1 = d_2$ then $\text{ES}(d_1, d_2) = 0$. However $\text{ES}(d_1, d_2) = 0 \Rightarrow \text{dis}(d_1, d_2) = 0$ and $\sum_{k=1}^{N} |\text{dis}(d_1, d_k) - \text{dis}(d_2, d_k)| = 0$. But $\text{dis}(d_1, d_2) = 0 \not\Rightarrow d_1 = d_2$. Hence ES is not a metric.

c) Let d_1, d_2, d_3 be any three documents and $\text{ES}(d_i, d_j) \geq 0, \ \forall\, i, j$. Then from equation 2 we have

$$\begin{aligned}\text{ES}(d_1, d_2) + \text{ES}(d_2, d_3) - \text{ES}(d_1, d_3) = \sum_{k=1}^{N} (&|\text{dis}(d_1, d_k) - \text{dis}(d_2, d_k)| + \\ &|\text{dis}(d_2, d_k) - \text{dis}(d_3, d_k)| - \\ &|\text{dis}(d_1, d_k) - \text{dis}(d_3, d_k)|) \\ &\geq 0 \ ,\end{aligned}$$

since the operator *modulus* is a metric. Thus ES satisfies the triangular inequality i.e.,

$$\text{ES}(d_1, d_2) + \text{ES}(d_2, d_3) \geq \text{ES}(d_1, d_3), \text{ if } \text{ES}(d_i, d_j) \geq 0, \ \forall\, i, j$$

So the triangular inequality holds good for non negative ES values.

d) Note that $\mathrm{ES}(d_1, d_2) \ngeq 0$ for any two documents d_1 and d_2. However the only negative value of ES is -1 and it has been used as a symbol to denote the complete dissimilarity between two documents.

3.4 Cluster Distance

Let us first discuss about some basic ideas of the proposed document clustering algorithm. The algorithm initially assumes each document as a single cluster. The cluster distance between two clusters C_1 and C_2 is the maximum of the set of non negative ES values between a pair of documents, one of which is from C_1 and the other is from C_2. The cluster distance will be -1 if there are no two documents which have a non negative ES value i.e., no similar documents are present in C_1 and C_2.

Let $SES(C_1, C_2) = \{ES(d_1, d_2) : ES(d_1, d_2) \geq 0,\ \forall\ d_1 \in C_1 \text{ and } \forall\ d_2 \in C_2\}$ be the *Set of Extensive Similarities*(SES) between the documents of C_1 and C_2. SES will be a null set if there exists two elements, one in C_1, and the other in C_2 such that their ES value is -1. Now the distance between C_1 and C_2 is defined as

$$\text{cluster\_dis}(C_1, C_2) = \begin{cases} -1 \quad \text{, if } SES(C_1, C_2) = \phi \\ \\ N - \max(SES(C_1, C_2)), \text{ otherwise} \end{cases} \tag{3}$$

Let us consider an example of two clusters C_1 and C_2, where C_1 contains four documents and C_2 contains three documents . So totally 12 ES values are there between C_1 and C_2. The cluster distance between C_1 and C_2 will be -1, if at least one of these 12 values is negative. The intuition behind negative cluster distance is, documents of C_1 and C_2 either share a very few number of words, or no word is common between them i.e., they have a very low content similarity. The essence of cluster distance lies in the fact that it would never merge two sets of dissimilar documents to the same cluster. This phenomenon of cluster distance makes the clustering task easier by segregating the dissimilar documents. We may observe from the experiments in next section that some clusters remain singleton clusters when CUES terminates. Basically these clusters (rather documents) have negative cluster distance with all other clusters.

3.5 Clustering Procedure

Initially a similarity matrix is required whose ij^{th} entry is the $\mathrm{ES}(d_i, d_j)$ value where d_i and d_j are i^{th} and j^{th} documents respectively. It is a square matrix and has N rows and N columns for N number of clusters. Each row or column represents a cluster. Initially each document is taken as a cluster. CUES will start with N individual clusters. Note that at the end of the algorithm, the number of clusters remaining is not necessarily 1 since some clusters may not be merged with any other clusters. Sometimes, some of the singleton clusters may remain singleton clusters when the algorithm is terminated.

At the first step CUES will merge the clusters whose cluster distance is minimum and the similarity matrix is updated. Then, the second minimum distant clusters is to be merged and so on. This process is continued till no more merges take place i.e., till there exist no two clusters with non negative cluster distance. In other words, the algorithm is terminated when negative cluster distance is observed between every pair of clusters. Algorithm 1 describes the steps of the proposed document clustering method in detail. The merging procedure stated in step 15 of Algorithm 1 merges two rows say i and j and the corresponding columns of the similarity matrix. Note that the row index represents a cluster and column index also represents a cluster. For every cell numbered k, $k = 1, 2, ..., n$, the

Algorithm 1 Clustering using Extensive Similarity (CUES)

Input: a) A set of N clusters, $C = \{C_1, C_2, ..., C_N\}$ and $noc = |C|$, number of clusters.
b) $C_i = \{d_i\}\ \forall i \in N$, where d_i is the i^{th} document of the data set.
c) A similarity matrix $Sim[i][j] = \text{cluster\_dis}(C_i, C_j),\ \forall i, j \in [1, N]$.

Steps of the Algorithm:

1: $X \leftarrow 0,\ \ Y \leftarrow 0$
2: **while** $noc > 1$ and $X \geq 0$ and $Y \geq 0$ **do**
3: min_dist $\leftarrow N$
4: $X \leftarrow -1,\ \ Y \leftarrow -1$
5: **for** $i = 1$ to $noc - 1$ **do**
6: **for** $j = i + 1$ to noc **do**
7: **if** min_dist $\geq$ cluster_dis(C_i, C_j) and cluster_dis$(C_i, C_j) \geq 0$ **then**
8: min_dist $\leftarrow$ cluster_dis(C_i, C_j)
9: $X \leftarrow i,\ \ Y \leftarrow j$
10: **end if**
11: **end for**
12: **end for**
13: **if** $X \geq 0$ and $Y \geq 0$ **then**
14: $C_X \leftarrow C_X \cup C_Y$
15: $Sim \leftarrow$ merge(Sim, i, j)
16: $noc \leftarrow noc - 1$
17: **end if**
18: **end while**
19: return C

method finds the minimum value between $Sim[i][k]$ and $Sim[j][k]$, and replace $Sim[i][k]$ with the minimum value. The jth row is removed. Similar procedure is repeated for the columns too, resulting in a symmetric matrix. When two clusters, say C_X and C_Y are merged in step 14 of Algorithm 1, then C_X is replaced by $C_X \cup C_Y$, and C_Y is removed and the index structure of the clusters are updated accordingly.

It is to be noted that the algorithm does not merge two clusters if the distance between them is -1. They remain separate till the end of the algorithm. Traditional document clustering algorithms can not identify two dissimilar clusters. They always generate a grade of similarity between the clusters which eventually merges those clusters. But the proposed cluster distance can identify two dissimilar clusters and never merges them. By this property, CUES can automatically identify the natural clusters in the data set and does not require a prior information of number of actual clusters for implementation.

In the section on experimental results, we shall observe that CUES produces some singleton clusters. These singleton clusters have negative cluster distance with the other clusters and so that they remain single and could be treated as outliers of the data set. If the θ value is very high then the documents within a particular cluster must have high extensive similarity, on the other hand it may produce huge number of singleton clusters. In such cases the quality of the clusters including these singleton clusters would need to be evaluated. Consequently, we can decrease the value of θ to reduce the total number of singleton clusters. Cluster distance is inherited from extensive similarity between documents. The extensive similarity not only determines the similarity between documents but also describes the underlying structure of the corpus. Ideally within a cluster the ES values between each pair of documents are close to each other and the extensive similarity between every pair of clusters is high at the end of the clustering. Algorithm 1 will normally continue if there

remains some clusters with a non-negative cluster distance, otherwise it will stop there. Note that no stopping criterion is needed for CUES.

3.6 Discussions

The structure of the proposed document clustering algorithm is quite similar to single-link hierarchical clustering (SLHC) technique. But the main difference between SLHC and CUES is the negative cluster distance between two clusters. Two clusters with insignificantly low similarity may be merged at any hierarchy of SLHC, but CUES will never merge them if they have negative cluster distance. Secondly, the SLHC technique needs to know the stopping criterion externally, but CUES is automatically stopped if there are no two clusters with non negative cluster distance.

The single-link algorithm, by contrast, suffers from a chaining effect [8]. It has a tendency to produce clusters that are straggly or elongated. The clusters that are separated by a bridge (thin line) of noisy patterns may be merged by single link clustering. CUES is designed like single-link algorithm, but it never suffers from chaining effect. In Single-link algorithm, the similarity between two clusters is taken as the similarity between two most similar documents of the two clusters. This sometimes gives raise to merger of two clusters where two points in two clusters possessing very small similarity values. CUES merges two clusters where the similarities with respect to all the documents are taken into consideration, and consequently, the low similarities are also considered. This results in merging of two clusters when every similarity value exceeds a threshold. Thus, chaining effect is not present in the resultant clusters since similarities between every pair of documents are taken into consideration for calculating the cluster distance.

It may be observed that whenever two clusters are merged, the similarity between any two documents in the merged cluster will at least be equal to θ. This interesting property of CUES can be observed from the following theorem.

Theorem 1. *Let C_1 and C_2 be two resulting clusters of the proposed scheme then*
a) $d_1, d_2 \in C_1 \Rightarrow dis(d_1, d_2) = 0$
b) $\exists\ d_1 \in C_1$ *and* $d_2 \in C_2$ *such that* $dis(d_1, d_2) = 1$

Proof. **1.a)** Let us assume that $d_1, d_2 \in C_1$ and $\text{dis}(d_1, d_2) = 1$. Initially we have two singleton clusters $\{d_1\}$ and $\{d_2\}$. After some iterations we would have clusters C_{11} and C_{12} in such a way that

1) $d_1 \in C_{11}$ and $d_1 \notin C_{12}$, 2) $d_2 \in C_{12}$ and $d_2 \notin C_{11}$, 3) $C_{11}, C_{12} \subseteq C_1$ and C_{11}, C_{12} are merged according to the proposed criterion.

Now $\text{dis}(d_1, d_2) = 1 \Rightarrow \text{ES}(d_1, d_2) = -1$ and as a result $\text{cluster\_dis}(C_{11}, C_{12}) = -1$. So C_{11} and C_{12} can not be merged, which is a contradiction and thus $\text{dis}(d_1, d_2) \neq 1$. Hence $\text{dis}(d_1, d_2) = 0$.

1.b) This is also proved here by the method of contradiction. Let us assume that the statement 1.b) is not true. That means there exists no $d_1 \in C_1$ and $d_2 \in C_2$ such that $\text{dis}(d_1, d_2) = 1$ i.e., $\forall\ d_1 \in C_1$ and $\forall\ d_2 \in C_2$, $\text{dis}(d_1, d_2) = 0$. So $\text{ES}(d_1, d_2) \geq 0$, $\forall\ d_1 \in C_1$ and $\forall\ d_2 \in C_2$. As a result $\text{cluster\_dis}(d_1, d_2) \geq 0$, and C_1, C_2 will be merged, contradicting the assumption. Hence $\exists\ d_1 \in C_1$ and $d_2 \in C_2$ such that $\text{dis}(d_1, d_2) = 1$. □

The quality of the resultant clusters of the proposed method may be observed from the above theorem.

3.7 An Estimation of θ

There are several methods available in literature to find a threshold for a two-class (one class corresponds to similar points, and the other corresponds to dissimilarpoints) classification problem. A popular method for such classification is histogram thresholding [17].

Let, for a given dataset, the number of distinct similarity values be n, and they are divided into p class intervals $s_0, s_1, ..., s_{p-1}$. Let $g(s_i)$ denote the number of occurrences of the similarity values in the class intervals $s_i,\ \forall i = 0, 1, ..., (p-1)$. Our aim is to find a threshold θ on the similarity values so that a similarity value $s < \theta$ implies the corresponding documents are practically dissimilar, otherwise they are similar. The aim is to make the choice of threshold to be data dependent. Without loss of generality, let us assume that (a) $s_i < s_j$ if $i < j$ and (b) $(s_{i+1} - s_i) = (s_1 - s_0),\ \forall i = 1, 2, ..., (p-2)$.

The basic steps of the histogram thresholding technique are as follow:

- Obtain the histogram corresponding to the given problem.
- Reduce the ambiguity in histogram. Usually this step is carried out using a window. One of the earliest such techniques is the moving average technique in time series analysis [18], which is used to reduce the local variations in a histogram. It is convolved with the histogram resulting in a less ambiguous histogram. We have used the weighted moving averages using window length 5 of the $g(s_i)$ values as,
$$f(s_i) = \frac{g(s_i)}{\sum_{j=0}^{p-1} g(s_j)} \times \frac{g(s_i) + g(s_{i+1}) + g(s_{i+2}) + g(s_{i+3}) + g(s_{i+4})}{5},\ \forall i = 0, 1, ..., p-5$$
- Find the valley regions in the modified histogram. A class interval s_i corresponding to the weight function $f(s_i)$ is said to be a valley region if $f(s_{i-1}) > f(s_i)$ and $f(s_i) < f(s_{i+1})$.
- If there is a single valley region, then the minimum value of the valley region is taken as the threshold. If the number of valley region is greater than 1, then the minimum value of the first valley region will be taken as the threshold.

In the experiments we are considering class intervals of length 0.05 for similarity values, and finding frequencies for the class intervals. We are also assuming that similarity (cosine similarity) value greater than 0.5 means the corresponding two documents are similar. Thus, the issue here is to find a $\theta,\ 0 < \theta < 0.5$ such that similarity value grater than θ denotes that the documents are similar. In the experiments we have used the method of moving averages with the window length of 5 for convolution. It has been found that several local peaks and local valleys are removed by this method. Even then, in the experiments the number of valley regions, some times, is found to be more than 1 (may be 3 or 4). In every such case we have taken the value of θ from the first valley region.

3.8 Time and Space Complexity

Here we discuss about the time and space complexity of the proposed clustering algorithm for N input documents. In the initialization phase of Algorithm 1, a similarity matrix has been built up which takes $O(N^2)$ time. The merging procedure takes $O(N)$ time to merge two clusters and rest of the steps take $O(1)$ time to execute. The algorithm finds two clusters with minimum cluster distance in step 7 in at most $N \times (N-1)/2$ iterations and the merging procedures of step 15 will take at most $N-1$ iterations. Rest of the steps take $O(1)$ time to execute. So to find two minimum distant clusters and then merge the clusters

to build a single cluster, it will take a total of at most $((N \times (N-1)/2) + (N-1))$ i.e., $O(N^2)$ time. Let there be m merges between two clusters, i.e., we have m iterations of the loop of step 2. So the time complexity of CUES is $O(mN^2)$. Practically $m << N$ and the time complexity will be $O(N^2)$. In principle there may be $N-1$ merges and thus the worst case time complexity of CUES is $O(N^3)$. Generally the text data sets are huge in size and the number of clusters are too less in comparison to the number of documents. Hence in reality it is very unlikely for CUES to have the time complexity $O(N^3)$.

The similarity matrix will require $N \times N$ space and to store N clusters, initially N space is required. Thus the space complexity of CUES is $O(N^2)$.

4. Experimental Evaluation

4.1 Experimental Setup

Four data sets, namely, *20-newsgroups*[1], Reuter-21578[2], Ohsumed[3] and Wap[1] are considered for experiment here. 20-newsgroups data is a collection of news articles collected from 20 different sources. Each news source constitutes a different category. In this dataset, articles with multiple topics are cross posted to multiple newsgroups i.e., there are overlaps between several classes. There are about 20,000 documents in the corpus. We have randomly selected 100 documents from each category and developed the data set *20ns* consists of 2000 documents.

Table 1: Data Sets Overview

Data Set	No. of Doc.	No of Terms	Categories	Avg Words/Doc.
20ns	2000	31086	20	258
oh	1150	19791	23	98
rcv1	2017	12906	30	66
rcv2	2017	12912	30	67
rcv3	2017	12820	30	66
rcv4	2016	13181	30	68
wap	1560	8460	20	216

Reuters-21578 is a collection of documents that appeared on Reuters newswire in 1987. The documents were originally assembled and indexed with categories by Carnegie Group Inc., and Reuters, Ltd. The corpus contains 21578 documents in 135 categories. Here we considered the ModApte version used in [9], in which there are 30 categories and 8067 documents. We have divided this corpus into four groups and with the name as rcv1, rcv2, rcv3 and rcv4. The detailed description of the four groups is given in Table 1.

The *Ohsumed* test collection is a set of 348,566 references from MEDLINE, the on-line medical information database, consisting of titles and/or abstracts from 270 medical journals over a five-year period (1987-1991). The available fields are title, abstract, MeSH indexing terms, author, source, and publication type. From the 50216 documents in 1991 which contain only the abstracts, Joachims [21] used the first 20,000 documents and divided them in training and test set, each containing 10000 documents. The specific task was to categorize the 23 cardiovascular diseases categories. Here we have randomly chosen 50 documents from each category of the 20,000 documents to build the data set *oh*.

1 http://www.cs.cmu.edu/∼TextLearning/datasets.html
2 http://www.daviddlewis.com/resources/testcollections/reuters21578/
3 http://disi.unitn.it/moschitti/corpora.htm

The *wap* data set [26] is from the WebACE (WAP) project and contains web pages listed in the subject hierarchy of Yahoo[4].

For each of the above data sets, the stop words have been extracted using the standard English stop word list [5]. Then, by applying the standard porter stemmer algorithm [10] for stemming, the inverted index is developed. Table 1 presents a brief overview of the data sets. It shows the number of documents, number of categories, the vocabulary size and average number of words per document of each data set.

4.2 Evaluation Criteria

If the documents within a cluster are similar to each other and dissimilar to the documents in the other clusters then the clustering algorithm is considered to be performing well. The data sets under consideration have labeled documents. Hence quality measures based on labeled data are used here for comparison. These measures are *f-measure* and *normalized mutual information*.

F-measure and normalized mutual information are very popular and are used by a number of researchers [2] [24] to measure the quality of a cluster using the class information of the document collection. Let us assume that R is the set of classes and S is the set of clusters. Consider there are I number of classes in R and J number of clusters in S. A total of N number of documents are there in the document corpus i.e., both R and S individually contains N documents. Let n_i is the number of documents belonging to class i, m_j is the number of documents belonging to cluster j and n_{ij} is the number of documents belonging to both class i and cluster j, for all *i=1, 2, ..., I* and *j=1, 2, ..., J*.

Mutual information is a symmetric measure to quantify the statistical information shared between two distributions which provides a sound indication of the shared information between a set of classes and a set of clusters. Let I(R,S) denotes the mutual information between R and S and E(R) and E(S) be the entropy of R and S respectively. I(R,S) and E(R) can be defined as

$$I(R,S) = \sum_{i=1}^{I}\sum_{j=1}^{J}\frac{n_{ij}}{N}\log\big(\frac{Nn_{ij}}{n_i m_j}\big), \qquad E(R) = \sum_{i=1}^{I}\frac{n_i}{N}\log\big(\frac{n_i}{N}\big)$$

There is no upper bound for I(R,S), so for easier interpretation and comparisons a normalized mutual information that ranges from 0 to 1 is desirable. The Normalized Mutual Information (NMI) as described by Strehl et. al. [23] is as follows:

$$NMI(R,S) = \frac{I(R,S)}{\sqrt{E(R)E(S)}}$$

Note that at least one document must be there in each class and each cluster i.e., $n_i > 0\ \forall i \in I$ and $m_j > 0\ \forall j \in J$. If there is no common elements between a class i and a cluster j (i.e., $n_{ij} = 0$) then we use the convention $0\log(0) = 0$. Note that NMI(S,S)=1, and thus normalized mutual information ranges from 0 to 1.

F-measure determines the recall and precision value of each cluster with a corresponding class. Let, for a query the set of relevant documents be from class i and the set of retrieved

4 http://www.yahoo.com

5 http://www.textfixer.com/resources/common-english-words.txt

documents be from cluster j. Then recall, precision and f-measure are given as :

$$Recall_{ij} = \frac{n_{ij}}{n_i}, \quad \forall\, i, j \quad \text{and} \quad Precision_{ij} = \frac{n_{ij}}{m_j}, \quad \forall\, i, j$$

$$F_{ij} = \frac{2 \times Recall_{ij} \times Precision_{ij}}{Recall_{ij} + Precision_{ij}}, \quad \forall\, i, j$$

If there is no common instance between a class and a cluster (i.e., $n_{ij} = 0$) then $F_{ij} = 0$. The value of F_{ij} will be maximum when $Precision_{ij} = Recall_{ij}$ and $n_{ij} \neq 0$ for a class i and cluster j. Thus the value of F_{ij} lies between 0 and 1. The best f-measure among all the clusters is selected as the f-measure for the query of a particular class is $F_i = \max_{j \in [0,J]} F_{ij}$, $\forall i$. The f-measure of all the clusters is weighted average of the sum of the f-measures of each class, $F = \sum_{i=1}^{I} \frac{n_i}{N} F_i$. We would like to maximize f-measure and normalized mutual information to achieve good quality clusters.

4.3 Analysis of Results

The documents are represented using the vector space model [5]. Let there are n terms in the vocabulary and N be the number of documents in each data set. The weight of each document vector is represented as follows by the *tf-idf* score.

$$w_{ij} = tf_{ij} \times idf_i, \ \forall\, i = 1, 2, ..., n \text{ and } \forall\, j = 1, 2, ..., N$$

Here tf_{ij} is the term frequency of the i^{th} term in j^{th} document and idf_i is the inverse document frequency of the i^{th} document i.e., $idf_i = log(\frac{N}{df_i})$, $\forall\, i = 1, 2, ..., n$, where df_i indicates the number of documents in which the term i occurs. The similarity between two documents is determined as the cosine of the angle between the document vectors.

In order to evaluate extensive similarity based clustering, eight basic clustering algorithms - bisecting k-means (BKM) clustering, k-means (KM) clustering, buckshot (BS) clustering, single-link hierarchical clustering (SLHC), average-link hierarchical clustering (ALHC), k nearest neighbor (KNN) clustering and two types of spectral clustering method (simple spectral clustering (SC) and spectral clustering using a kernel (SCK), as discussed in Section 2) are selected for comparison. K-means and bisecting k-means were executed 10 times to reduce the effect of random initialization of seed points. Buckshot was also executed 10 times to reduce the effect of random initialization of initial $\sqrt{kN}$ documents. The f-measure and NMI values shown here are the average of 10 different results. The number of clusters of the other methods are same as the number of clusters produced by CUES. Note that the proposed method finds the total number of clusters automatically from the corpus. The value of θ for each data set is chosen using the histogram thresholding method described above. We have chosen k = 10 for KNN clustering method. Table 2 and Table 3 shows the f-measure and NMI values respectively of all the data sets. Number of clusters (NC), number of singleton clusters (NSC) developed by CUES are also shown. Here the number of clusters includes the singleton clusters also i.e., NC = NSC + the number of merged clusters. The f-measure and NMI are calculated using these NC values.

Table 2 and Table 3 show the comparison of CUES with the other eight clustering methods for seven data sets. So for Table 2 and Table 3 there are 56 comparisons in each Table for the proposed method. Out of these 112 cases CUES performed better than the other methods in 106 cases and for the rest 6 cases other methods (e.g., buckshot, k-means) have an edge over CUES. The exceptions where the other methods have an edge over CUES are, i) buckshot,

Table 2: Comparison of Various Clustering Methods Using F-measure

Data Sets	θ	NC[6]	NSC[7]	F-measure								
				BKM[8]	KM	BS	SLHC	ALHC	KNN	SC	SCK	CUES (Proposed)
20ns	0.029	24	3	0.217	**0.439**	0.406	0.095	0.115	0.095	0.224	0.394	0.274
oh	0.0348	25	2	0.130	0.112	0.152	0.083	0.088	0.083	0.101	0.138	**0.193**
rcv1	0.077	34	3	0.188	0.212	0.307	0.408	0.362	0.418	0.301	0.318	**0.522**
rcv2	0.075	33	3	0.165	0.202	0.286	0.407	0.350	0.417	0.419	0.439	**0.551**
rcv3	0.085	32	2	0.218	0.239	0.355	0.409	0.372	0.413	0.211	0.331	**0.578**
rcv4	0.087	34	5	0.222	0.286	0.287	0.409	0.379	0.414	0.229	0.297	**0.590**
wap	0.037	20	0	0.283	0.412	0.417	0.177	0.180	0.178	0.174	0.381	**0.427**

[6] NC stands for number of clusters.
[7] NSC stands for number of singleton clusters.
[8] BKM, KM, BS, SLHC, ALHC, KNN, SC and SCK are - bisecting k-means, k-means, buckshot, single-link agglomerative hierarchical clustering, average-link hierarchical clustering, k nearest neighbor clustering , spectral clustering, and spectral clustering using kernel respectively and CUES is the proposed method.

k-means and spectral clustering (SCK) for *20ns* (0.406, 0.439 and 0.394 respectively for buckshot, k-means and SCK and the value of CUES is 0.274) when f-measure is used for performance evaluation, and ii) buckshot, k-means and SCK for *20ns* (0.357, 0.397 and 0.401 for buckshot, k-means and SCK and the value of CUES is 0.233) when NMI is used for performance evaluation.

Table 3: Comparison of Various Clustering Methods Using NMI

Data Sets	θ	NC[9]	NSC	Normalized Mutual Information								
				BKM	KM	BS	SLHC	ALHC	KNN	SC	SCK	CUES (Proposed)
20ns	0.029	24	3	0.228	0.397	0.357	0.070	0.126	0.071	0.180	**0.401**	0.233
oh	0.0348	25	2	0.130	0.132	0.145	0.100	0.118	0.097	0.099	0.209	**0.185**
rcv1	0.077	34	3	0.300	0.434	0.444	0.083	0.094	0.160	0.190	0.381	**0.476**
rcv2	0.075	33	3	0.301	0.412	0.401	0.079	0.147	0.148	0.242	0.374	**0.466**
rcv3	0.085	32	2	0.312	0.394	0.401	0.078	0.289	0.133	0.194	0.390	**0.415**
rcv4	0.087	34	5	0.331	0.403	0.391	0.073	0.073	0.134	0.206	0.378	**0.416**
wap	0.037	20	0	0.268	0.412	0.423	0.075	0.041	0.072	0.126	0.426	**0.456**

[9] All the symbols in this Table are the same symbols used in Table 2.

It needs to be checked for case (i) and (ii) whether the values for buckshot, k-means and SC are significantly different from the respective values of CUES (e.g., whether any one of the 10 values whose average is 0.439 (k-means, case (i), *20ns* data set) is significantly different from 0.274). For all the other cases where CUES performs better than the other clustering methods, it is required to check whether CUES performs significantly better than the other methods.

A generalized version of paired *t-test* is suitable for testing the equality of means when the variances are unknown. This problem is the classical Behrens-Fisher problem in hypothesis

testing and a suitable test statistic[10] is described and tabled in [19] and [20], respectively. It has been found that out of those 106 cases where CUES performed better than the other methods, in 96 cases the difference was statistically significant for the level of significance 0.05. The difference was statistically significant for the rest 6 cases and for the same level of significance. So the performance of the proposed method is found to be *significantly better* than the other methods in 90.56% cases.

It is to be noted that the result of t-test were significant in 20ns data for all the methods and buckshot, k-means and SCK performed better than CUES. In *20ns* data set there are overlaps between several classes. The centroid based algorithms, like k-means, buckshot are generally produce better results than the hierarchical algorithms when there are overlap between two clusters [1]. So buckshot, k-means and SCK have produced better clusters in *20ns* data sets. Otherwise the overall experimental evaluation shows the effectiveness of the extensive similarity based document clustering approach. The extensive similarity based method groups two documents not only based on their mutual similarity but also on their similarity with the other documents in the document collection. This fact is observed in the experimental results.

Table 4: Performance of the Proposed Method on Different Values of θ

	Data	**20ns**	**oh**	**rcv1**	**rcv2**	**rcv3**	**rcv4**	**wap**
$\theta = 0.1$	NC	362	229	63	68	47	47	157
	NSC	136	65	4	10	5	8	20
	F-measure	0.28	0.225	0.532	0.549	0.518	0.598	0.358
	NMI	0.345	0.461	0.464	0.499	0.437	0.464	0.515
$\theta = 0.2$	NC	774	562	387	382	431	386	532
	NSC	423	306	159	150	192	159	225
	F-measure	0.229	0.161	0.461	0.473	0.469	0.413	0.231
	NMI	0.495	0.571	0.540	0.549	0.554	0.532	0.574
$\theta = 0.4$	NC	1329	856	1139	1170	1187	1200	1044
	NSC	960	642	890	953	956	989	766
	F-measure	0.117	0.075	0.214	0.281	0.164	0.197	0.149
	NMI	0.602	0.608	0.536	0.542	0.536	0.535	0.599

Table 4 shows the performance of the proposed clustering algorithm on different θ values, say $\theta = 0.1, 0.2, 0.4$. Since the sizes of the data sets are small and average number of words per document is very low in compare to the total number of words, the average similarity between the documents will be very low. As a result total number of singleton clusters will be more on high values of θ, which is not desirable in practice. It can be seen from Table 4 that for $\theta = 0.1$, 0.2 and 0.4 the NC and NSC values are very high. So $\theta = 0.1$, 0.2 and 0.4 can not be used for comparison with the other methods. In this situation, low values of θ have to be taken for experiment. The values of θ chosen for experiments using the proposed histogram thresholding method are close to 0 in most of the data sets due to the sparsity of the data sets. Even then the experimental analysis shows that the proposed clustering technique outperforms the other methods. Thus the proposed histogram thresholding approach has been found to yield a good estimate of θ.

10 The test statistic is of the form $t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{s_1^2/n_1 + s_2^2/n_2}}$, where $\bar{x}_1, \bar{x}_2$ are the means, s_1, s_2 are the standard deviations and n_1, n_2 are the number of observations

4.4 Processing Time

We have measured the processing time of each method on a quad core Linux workstation. The time (in seconds) taken by different clustering methods to cluster each text data set are reported in Table 6. The time shown here for CUES is the sum of the time taken to estimate the value of θ, to build the similarity matrix, and to perform the proposed clustering. The time shown for bisecting k-means, buckshot and k-means are the average of the processing times for 10 iterations. It is to be mentioned that the codes for all the algorithms are written in C++ and the data structures for all the algorithms are developed by the authors. The time taken by CUES is less than the time taken by ALHC, KNN and SCK for each data set. The processing time of CUES is comparable with BKM, KM, BS, SLHC, and SC for all the data sets. It may be noted that the proposed method outperforms all the other methods in terms of cluster quality. Hence we can allow this much cost expenditure of CUES due to time to get such a good performance for clustering of documents. The data sets used in the experiments have several dimensions starting from 8460 (wap) to 31086 (20ns), and the proposed method is found to perform well in the presence of high dimensional data.

Table 5: Processing Time (in seconds) of Different Clustering Methods

Data	BKM	KM	BS	SLHC	ALHC	KNN	SC	SCK	CUES
20ns	425	439	435	419	437	434	416	430	426
oh	146	159	148	153	161	164	152	163	161
rcv1	279	278	276	275	291	286	268	288	280
rcv2	279	289	287	279	295	287	286	290	281
rcv3	246	255	249	269	296	288	277	284	275
rcv4	268	256	262	284	301	296	283	296	290
wap	197	206	196	229	237	234	224	236	230

5. Conclusions

We demonstrated a system composed of extensive similarity between documents to improve the accuracy of measuring the similarity between documents and used this similarity to perform document clustering. Intuitively the similarity between two documents can not be obtained totally from the content but from their inherent extensive similarity in a corpus. We have incorporated extensive similarity on the basis of similarity between two documents and their distances with every other document in the document collection. Thus the documents with high extensive similarities (determined by ES) may be grouped in the same cluster. The proposed clustering method CUES is developed along this line. The salient features of CUES are, (i) the algorithm can identify two dissimilar clusters and will never merge them, (ii) the algorithm can be stopped if the distance between two clusters becomes very high, since at each step CUES checks the cluster distance to merge two clusters and (iii) there is no need to input the desired number of clusters prior to implement the algorithm. The range of similarity between every two documents in a cluster is known to us and it is between θ and 1. The total number of clusters is determined automatically by the proposed method, but on requirement the total number of clusters can be bounded by varying the value of θ.

The performance of CUES is mainly dependent on θ and the selection of θ is dependent upon two factors - nature of the data sets and the choice of the user. We have described a histogram thresholding based method for selecting a value of θ from the actual similarity

matrix of a data set. The results reveal the fact that using those θ values the proposed approach performs significantly better than other clustering techniques. Cosine similarity is used in this article to find the content similarity between documents, since it can find the exact content similarity (even) between the documents with different lengths. But one can use any other similarity or dissimilarity measure as per convenience instead of cosine similarity in the distance function of Equation 1 to perform CUES (viz. The method presented here is aimed at document clustering, but it can be easily generalized to any data set as well). Document similarity can be measured by finding semantic relatedness between documents instead of content similarity. In future we shall study the performance of CUES using semantic similarity between documents. The incoming and outgoing links play an important role in finding the similarity between web pages. The merit of extensive similarity can be extended to draw a relation between web pages using their incoming and outgoing links.

References

[1] Richard C. Dubes and Anil K. Jain, Algorithms for Clustering Data, Prentice Hall, 1988.
[2] M. Steinbach, G. Karypis, V. Kumar, A Comparison of Document Clustering Techniques, Text Mining Workshop, KDD, 2000.
[3] D. R. Cutting, D. R. Karger, J. O. Pedersen, J.W. Tukey, Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections, In Proceedings of the International Conference on Research and Development in Information Retrieval, SIGIR'93, pp. 126-35, 1993.
[4] K. M. Hammouda, and M. S. Kamel, Efficient Phrase-Based Document Indexing for Web Document Clustering, IEEE Transaction on Knowledge and Data Engineering, vol.16, pp. 1279-1296, 2004.
[5] G. Salton, A. Wong, and C. Yang, A Vector Space Model for Automatic Indexing, Communications of ACM, vol. 18 (11), pp. 613-620, 1975.
[6] C. D. Manning, P. Raghavan and H. Schutze, Introduction to Information Retrieval. Cambridge University Press, New York, 2008.
[7] C. J. V. Rijsbergen, Information Retrieval, Butterworths, London, Second Edition, 1979.
[8] A. K. Jain, M. N. Murty, and P. J. Flynn, Data Clustering: A Review, ACM Computing Surveys, vol. 31 (3), pp. 264-323, 1999.
[9] D. Cai, X. He, and J. Han, Document Clustering Using Locality Preserving Indexing, IEEE Transaction on Knowledge and Data Engineering, vol. 17 (12), pp. 1624-1637, 2005.
[10] M. F. Porter, An Algorithm for Suffix Stripping, Program. vol. 14 (3), pp. 130-137, 1980.
[11] P. Berkhin, Survey of Clustering Data Mining Techniques, Accrue Software, Inc, 2002.
[12] M. A. Ismail, and S. Z. Selim, K-Means Type Algorithms: A Generalized Convergence Theorem and Characterization of Local Optimality, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 6 (1), pp. 81-87, 1984.
[13] D. Arthur, B. Manthey, and H. Roglin Smoothed Analysis of the k-Means Method. Journal of ACM, vol. 58 (5), 2011.
[14] G. W. Milligan and M. C. Cooper, An Examination of Procedures for Detecting the Number of Clusters in a Data Set, Psychometrika, vol. 50, pp. 159-179, 1985.
[15] J. J. Carlson, M. R. Muguira, J. B. Jordan, G. M. Flachs, and A. K. Peterson, Final Report: Weighted Neighbor Data Mining. SANDIA Report, SAND 2000-3122, December 2000.
[16] B.V. Dasarathy, Nearest Neighbor NN Norms: NN Pattern Classification Techniques. McGraw-Hill Computer Science Series. IEEE CS Press, 1991.
[17] C.A. Glasbey, An Analysis of Histogram-Based Thresholding Algorithms. Graphical Models and Image Processing, vol. 55(6), pp. 532-537, 1993.
[18] R. G. Brown, Smoothing, Forecasting and Prediction of Discrete Time Series. Prentice-Hall, Englewood Cliffs, NJ, 1962.
[19] E.L. Lehmann, Testing of Statistical Hypotheses. New York: John Wiley, 1976.
[20] C. R. Rao, S. K. Mitra, A. Matthai and K.G. Ramamurthy, Editors, Formulae and Tables for Statistical Work Statistical Publishing Society, Calcutta, 1966.

[21] T. Joachims, Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In Proceedings of the Tenth European Conference on Machine Learning (ECML 98), pp. 137-142, Berlin, Germany, 1998.
[22] A. Huang, D. Milne, E. Frank, and I. H. Witten, Clustering Documents with Active Learning using Wikipedia, In Proceedings of the Eighth IEEE International Conference on Data Mining, pp. 839-844, 2008.
[23] A. Strehl and J. Ghosh, Cluster Ensembles - A Knowledge Reuse Framework for Combining Multiple Partitions. The Journal of Machine Learning Research, vol. 3, pp. 583-617, 2003.
[24] N. X. Vinh, J. Epps and J. Bailey, Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance. The Journal of Machine Learning Research, vol. 11, pp. 2837-2854, 2010.
[25] S. Banerjee, K. Ramanathan, A. Gupta, Clustering Short Texts using Wikipedia, In Proceedings of the 30^{th} International Conference on Research and Development in Information Retrieval, SIGIR'2007, pp. 787-788, 2007.
[26] Y. Zhao and G. Karypis, Empirical and Theoretical Comparisons of Selected Criterion Functions for Document Clustering. Machine Learning, Springer, vol. 55(3), pp. 311-331, 2004.
[27] C. J. Alpert, S. Z. Yao, Spectral Partitioning: The More Eigenvectors, The Better. In Proceedings of the 32^{th} ACM/IEEE Design Automation Conference, pp. 195-200, 1995.
[28] A. Y. Ng , M. I. Jordan and Y. Weiss, On Spectral Clustering: Analysis and An Algorithm. In Proceedings of Neural Information Processing Systems (NIPS'2001), pp. 849-856, 2001.
[29] M. Filipponea, F. Camastrab, F. Masullia and S. Rovettaa, A Survey of Kernel and Spectral Methods for Clustering. Pattern Recognition, vol. 41 (1), pp. 176-190, 2008.
[30] X. Liu, X. Yong and H. Lin, An Improved Spectral Clustering Algorithm Based on Local Neighbors in Kernel Space. Computer Science and Information Systems. vol. 8 (4), pp. 1143-1157, 2011.
[31] T. H. Cao and T. M. Tang and C. K. Chau, Text Clustering with Named Entities. Data Mining: Foundations and Intelligent Paradigms, vol. 23, pp. 267-287, 2012.
[32] N. Oikonomakou and M. Vazirgiannis, A Review of Web Document Clustering Approaches. Data Mining and Knowledge Discovery Handbook, vol. 6, pp. 931-948, 2010.
[33] J. Wang, Y. Zhou, L. Li, B. Hu and X. Hu, Improving Short Text Clustering Performance with Keyword Expansion. Advances in Intelligent and Soft Computing, vol. 56, pp. 291-298, 2009.
[34] L. Jing, M. K. Ng and J. Z. Huang, Knowledge Based Vector Space Model for Text Clustering. Knowledge and Information Systems, vol. 25(1), pp. 35-55, 2010.