Information Technology Journal, 2010
ISSN 1812-5638

A Literature Review on Document Clustering

K. Premalatha and A.M. Natarajan
Bannari Amman Institute of Technology, Erode, TN, India

Abstract: Information Retrieval (IR) is the discipline of searching for documents, for information within documents and metadata about documents. The document clustering improves the retrieval effectiveness of the IR System. If documents can be clustered together in a sensible order, then indexing and retrieval operations can be optimized. This study presents a review on document clustering.

Key words: Document clustering, information retrieval, partitional clustering, hierarchical clustering, precision, recall

INTRODUCTION

Document clustering has become an increasingly important task in analyzing huge numbers of documents distributed among various sites. The challenging aspect is to organize the documents in a way that results in better search without introducing much extra cost and complexity. The Cluster Hypothesis is fundamental to the issue of improved effectiveness. It states that relevant documents tend to be more similar to each other than to non-relevant documents and therefore tend to appear in the same clusters (Jardine and van Rijsbergen, 1971). If the cluster hypothesis holds for a particular document collection, then relevant documents will be well separated from non-relevant ones. A relevant document may be ranked low in a best-match search because it may lack some of the query terms. In a clustered collection, this relevant document may be clustered together with other relevant items that do have the required query terms and could therefore be retrieved through a clustered search (Croft, 1978). According to best-match IR systems, if a document does not contain any of the query terms then its similarity to the query will be zero and this document will not be retrieved in response to the query. Document clustering offers an alternative file organization to that of best-match retrieval and it has the potential to address this issue, thereby increase the effectiveness of an IR system.

Document clustering has traditionally been applied statically to an entire document collection before querying (static clustering). An alternative application of clustering is to only cluster documents that have been retrieved by an IR system in response to a query (post-retrieval clustering) (Preece, 1973). Under post-retrieval clustering the resulting groups of documents are likely to be different for different queries. Document clustering typically operates based on the notion of inter document similarity. The set of terms shared between a pair of documents is typically used as an indication of the similarity of the pair.

Document clustering has been investigated for use in different areas of text mining and IR. Initially, document clustering was investigated for improving the precision or recall in IR systems (Rijsbergen *et al*., 1981) and as an efficient way of finding the nearest neighbors of a document (Buckley and Lewit, 1985). Clustering has been proposed for use in browsing a collection of documents (Cutting *et al*., 1992) or in organizing the results returned by a search engine in response to a user's query (Zamir and Etzioni, 1998).

PROBLEM FORMULATION

The clustering problem is expressed as follows:

The set of N documents $D = \{D_1, D_2, ... D_N\}$ is to be clustered. Each $D_i \in \Re^{N_d}$ is an attribute vector consisting of N_d real measurements describing the object. The documents are to be grouped into non-overlapping clusters $C = \{C_1, C_2, ... C_N\}$ (C is known as a clustering), where, K is the number of clusters, $C_1 \cup C_2 \cup ... \cup C_K$, $C_i \neq \phi$ and $C_1 \cap C_2 = \phi$ for $i \neq j$.

Assuming $f: D \times D \rightarrow \Re^+$ is a measure of similarity between document feature vectors. Clustering is the task of finding a partition $\{C_1, C_2, ..., C_K\}$ of D such that $\forall i, j \in \{1, ...K\}, j \neq i, \forall x \in C_i: f(x, O_i) \geq f(x, O_j)$ where, O_i is one cluster representative of cluster C_i.

The goal of clustering is stated as follows:

Given:

- A set of documents $D = \{D_1, D_2, ... D_N\}$
- A desired number of clusters K
- An objective function or fitness function that

Corresponding Author: K. Premalatha, Bannari Amman Institute of Technology, Erode, TN, India

evaluates the quality of a clustering, the system has to compute an assignment g: D→(1,2,...,K} and maximizes the objective function

VECTOR-SPACE MODEL OF DOCUMENTS

In IR, a document refers generically to the unit of text indexed in the system and available for retrieval. A collection refers to a set of documents being used to satisfy user requests. A term refers to a lexical item that occurs in a collection, but it may also include phrases.

The representation of a set of documents as vectors in a common vector space is known as the vector space model. In the vector space model of IR, documents are represented as vectors of features representing the terms that occur within the collection (Salton, 1971). The value of each feature is called the term weight and is usually a function of the term's frequency (or tfidf) in the document, along with other factors.

The vector space model procedure can be divided in to three stages. The first stage is the document indexing where content bearing terms are extracted from the document text. The second stage is the weighting of the indexed terms to enhance retrieval of document relevant to the user. The final stage is identifying the similarities between the document and centroid of the cluster.

Document indexing: More generally, a vector for a document D_i is represented as:

$$D_i = \left(w_{1,i}, w_{2,i}, ..., w_{N_d,i}\right) \quad (1)$$

where, D_i denotes a particular document (or feature vector), the individual scalar components $w_{j,i}$ of a document D_i are called features and N is the dimensionality of the document space. That is a document vector contains a weight feature for each of the N_d terms that occur in the collection as whole; $w_{1,i}$ thus refers to the weight that term 1 has in document i. It is useful to view the features used to represent documents in a multi-dimensional space, where the feature weights serve to locate documents in that space. The set of documents in a collection then turns into a vector space, with one axis for each term. Premalatha and Natarajan (2008b) introduced concept based indexing for dimensionality reduction in documents. In this method the concept hierarchy is created and the documents are indexed based on the concept rather keywords.

Term weighting: The term weights are set as the simple frequency counts of the terms in the documents. This method is used to assign terms weights in the document. The term frequency is simplest form the raw frequency of a term within a document. This reflects the intuition that terms occur frequently within a document may reflect its meaning more strongly than terms that occur less frequently and should thus have higher weights.

The second factor is used to give a higher weight to words that only occur in a few documents. Terms that are limited to a few documents are useful for discriminating those documents from the rest of the collection, while terms that occur frequently across the entire collection aren't helpful. The Inverse Document Frequency or IDF term weight (Sparck, 1972) is one way of assigning higher weights to these more discriminative words. IDF is defined via the fraction N/n_i, where, N is the total number of documents in the collection and n_i is the number of documents in which term i occurs. The fewer documents a term occurs in, the higher this weight. The lowest weight of 1 is assigned to terms that occur in all the documents. Due to the large number of documents in many collections, this measure is usually squashed with a log function. The resulting definition IDF is thus:

$$idf_i = \log\left(\frac{N}{n_i}\right) \quad (2)$$

Combining term frequency with IDF results in a scheme known as tf-idf weighting.

$$w_{i,j} = tf_{i,j} \times idf_i \quad (3)$$

In tfidf weighting, the weight of term i in the vector for document j is the product of its overall frequency in j with the log of its inverse document frequency in the collection. The tfidf thus prefers words which are frequent in the current document j but rare overall in the collection. The characterization of documents as vectors of term weights allows us to view the document collection as a whole as a matrix of weights, where $w_{i,j}$ represents the weight of term i in document j.

Similarity measure: In vector-based information retrieval the similarity between two documents measured by the cosine of the angle between their vectors. When two documents are identical they will receive a cosine of one; when they are orthogonal that is it shares no common terms they will receive a cosine of zero.

The equation for cosine is:

$$sim(D_k, D_j) = \frac{D_k \cdot D_j}{\|D_k\| \times \|D_j\|}$$

$$= \frac{\sum_{i=1}^{N_d} w_{i,k} \times w_{i,j}}{\sqrt{\sum_{i=1}^{N_d} w_{i,k}^2} \times \sqrt{\sum_{i=1}^{N_d} w_{i,j}^2}} \quad (4)$$

The cosine is the normalized dot product. That is, cosine is the dot product between the two vectors divided by the lengths of each of the two vectors. This is because the numerator of the cosine is the dot product. The denominator of the cosine contains terms for the lengths of the two vectors.

The similarity between the document D_i and centroid O_j of the cluster C_j is measured as shown in Eq. 5:

$$\text{sim}(O_j, D_i) = \frac{O_j \cdot D_i}{\|O_j\| \times \|D_i\|} \quad (5)$$

The Distance Measures are used to identify the distance between the document and centroid O_j. Some of the distance measures are:

Distance measure	Function
Euclidean distance measure	$\text{dist}(D_i, O_j) = \sqrt{(d_{i1} - o_{j1})^2 + (d_{i2} - o_{j2})^2 + \cdots + (d_{ni} - o_{nj})^2}$
Manhattan distance measure	$\text{dist}(D_i, O_j) = \sqrt{(d_{i1} - o_{j1})^2 + (d_{i2} - o_{j2})^2 + \cdots + (d_{ni} - o_{nj})^2}$
Minkowski distance measure	$\text{dist}(D_i, O_j) = ((d_{i1} - o_{j1})^p + (d_{i2} - o_{j2})^p + \cdots + (d_{ni} - o_{nj})^p)^{1/p}$

Term selection: The words that occur in a collection are used to represent the documents in the collection. The common variations on this assumption involve the tokenization, a stop list and use of stemming is used to identify words in the documents.

Stemming: For grammatical reasons, documents are going to use different forms of a word, such as organize, organizes and organizing. Additionally, there are families of derivationally related words with similar meanings, such as democracy, democratic and democratization. The goal of stemming is to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form. Removing suffixes by automatic means is an operation which is especially useful in the field of IR. The suffix stripping process will reduce the total number of terms in the IR system and hence reduce the size and complexity of the data in the system. The Porter stemmer (Porter, 1980) is frequently used for retrieval from collections of English documents. The proposed work uses Porter Stemmer algorithm for stemming.

CONVENTIONAL CLUSTERING ALGORITHM

Hierarchical and partitional clustering are two clustering techniques that are commonly used for document clustering. This section also discusses some other important clustering algorithms.

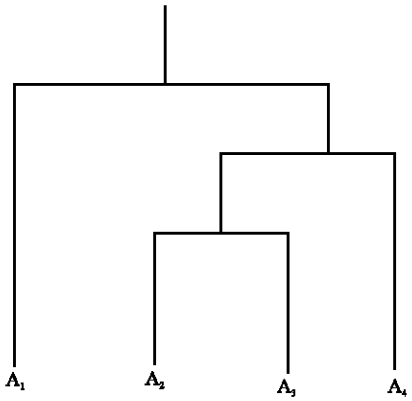

Fig. 1: Hierarchical agglomerative clustering

Hierarchical clustering: Hierarchical clustering techniques proceed by either a series of successive merges or a series of successive divisions. For both methods, the number of clusters is needed to select a clustering from the hierarchy. However the difference between the levels of the hierarchy may be an indication of the correct number of clusters. Hierarchical clustering generates a hierarchical tree of clusters. This tree is also called a dendrogram (Berkhin, 2005). Hierarchical methods can be further classified into agglomerative methods and divisive methods.

Agglomerative method: In an agglomerative method, originally, each object forms a cluster. Then the two most similar clusters are merged iteratively until some termination criterion is satisfied. Figure 1 shows an example for Hierarchical Agglomerative method. This figure depicts four patterns labeled A_1, A_2, A_3 and A_4. Initially agglomerative method places each object into a cluster of its own. The clusters are merged step-by-step according to some criterion. In the given example A_2 and A_3 are merged because A_2 and A_3 form the minimum Euclidean distance. The cluster merging process repeats until all of the objects are eventually merged to form one cluster. A hierarchical algorithm yields a dendrogram representing the nested grouping of patterns and similarity levels at which groupings change. The dendrogram can be broken at different levels to yield different clusterings of the data.

The following are the steps in an agglomerative hierarchical clustering algorithm for grouping N objects.

Step 1: Begin with N clusters, each containing one object

Step 2: Calculate the distance between each pair of clusters. These distances are usually stored in a symmetric distance matrix

Step 3: Merge the two clusters with the minimum distance
Step 4: Update the distance matrix
Step 5: Repeat steps 3 and 4 until a single cluster remains

Agglomerative algorithms are usually classified according to the inter-cluster similarity measure they use. The most popular of these are single-link, complete-link and group average. Common for all agglomerative methods is high computational complexity, often quadratic or worse.

Single-link clustering algorithms based on this similarity measure join the two clusters containing the two closest documents that are not yet in the same cluster. Complete-link clustering algorithms using this similarity measure join the two clusters with the minimum most-distant pair of documents. In this way, clusters are kept small and compact since all documents within a cluster have a maximum distance to each other (Frakes and Baeza-Yates, 1992). Group Average Clustering algorithms using this similarity measure join the two clusters with the minimum average document distance.

Divisive method: In a divisive method, from a cluster which consists of all the objects, one cluster is selected and split into smaller clusters recursively until some termination criterion is satisfied. Top-down clustering requires a method for splitting a cluster and proceeds by splitting clusters recursively until individual documents are reached. Top-down clustering is conceptually more complex than bottom-up clustering.

It is evident that divisive algorithms produce more accurate hierarchies than bottom-up algorithms in some circumstances (Steinbach *et al*., 2000). Bottom-up methods make clustering decisions based on local patterns without initially taking into account the global distribution. These early decisions cannot be undone. Top-down clustering benefits from complete information about the global distribution when making top-level partitioning decisions.

Partitional clustering: Partitional clustering divides data into several subsets. Because checking all possible subset systems is computationally infeasible, certain greedy heuristics are used in the form of iterative optimization. Specifically, this means different relocation schemes that iteratively reassign points between the K clusters. Unlike hierarchical methods, in which clusters are not revisited after being constructed, relocation algorithms gradually improve clusters. With appropriate data, this results in high quality clusters.

A problem accompanying the use of a partitional algorithm is the choice of the number of desired output clusters. The partitional techniques usually produce clusters by optimizing a criterion function defined either locally or globally. The algorithm is typically run multiple times with different starting states and the best configuration obtained from all of the runs is used as the output clustering.

K-means clustering algorithm: The K-means algorithm is by far the most popular clustering tool used in scientific and industrial applications. The name comes from representing each of K clusters C_j where $j \in \{1,2 \ldots , K\}$, by the mean of its points, the so-called centroid. A centroid almost never corresponds to an actual data point.

The sum of differences between a point and its centroid expressed through appropriate distance is used as the objective function. It is the sum of the squares of errors between the points and the corresponding centroids, is equal to the total intra-cluster variance. This measure is used to show how well the centroids represent the members of their cluster.

$$E(C)=\sum_{j=1}^{K}\sum_{x_i \in c_j}\left\|x_i - C_j\right\|^2 \quad (6)$$

K-means usually starts with selecting as initial cluster centers K randomly selected documents, the seeds. It then moves the cluster seed centers around in space in order to minimize E(C). This is done iteratively by repeating two steps until a stopping criterion is met: reassigning documents to the cluster with the closest centroid; and recomputing each centroid based on the current members of its cluster. Several termination conditions have been proposed:

- A fixed number of iterations have been completed
- Assignment of documents to clusters does not change between iterations
- Centroids do not change between iterations

REVIEW OF RELATED WORKS

General references about clustering included in Hartigan (1975), Jain and Dubes (1988), Kaufman and Rousseeuw (1990), Everitt (1993), Mirkin (1996), Jain *et al*. (1999), Fasulo (1999), Han *et al*. (2001) and Ghosh (2002). An introduction to data mining clustering techniques can be found in Han *et al*. (2001). There is a close relationship between clustering techniques and many other disciplines. Clustering has been used in statistics (Arabie and Hubert, 1996) and science (Massart and

Kaufman, 1983). The introduction into pattern recognition framework is given in Duda and Hart (1973). Machine learning clustering algorithms were applied to image segmentation and computer vision (Jain and Flynn, 1966). Statistical approaches to pattern recognition found in Dempster *et al.* (1977) and Fukunaga (1990). Clustering is widely used for data compression in image processing (Gersho and Gray, 1992). Clustering in data mining was brought to life by intense developments in information retrieval and text mining (Dhillon *et al.*, 2001) spatial database applications (Sander *et al.*, 1998) sequence and heterogeneous data analysis (Cadez *et al.*, 2001), Web applications (Cooley *et al.*, 1999) DNA analysis in computational biology (Ben-Dor and Yakhini, 1999) and many others.

The papers published in the document clustering field covers a number of diverse areas, such as the development of efficient algorithms for document clustering (Larsen and Aone, 1999), the visualization of clustered document spaces (Allan *et al.*, 2001), the application of document clustering to browsing large document collections (Hearst and Pedersen, 1996). Document clustering has also been used to automatically generate hierarchical clusters of documents (Koller and Sahami, 1997). A somewhat different approach (Aggarwal *et al.*, 1999) finds the natural clusters in already existing document taxonomy and then uses these clusters to produce an effective document classifier for new documents.

Jain *et al.* (1999) cover the topic very well from the point of view of cluster analysis theory and they break down the methodologies mainly into partitional and hierarchical clustering methods. In this work the clustering algorithm used falls under the partitional category. Chakrabarti (2003) also discusses various types of clustering methods and categorizes them into partitioning, geometric embedding and probabilistic approaches.

Text mining research in general relies on a vector space model, first proposed by Salton (1971) to model text documents as vectors in the feature space. Features are considered to be the words in the document collection and feature values come from different term weighting schemes, the most popular of which is the Term Frequency-Inverse Document Frequency (tfidf) term weighting scheme. This model is simple but assumes independence between words in a document, which is not a major problem for statistical-based methods, but poses difficulty in phrase-based analysis.

Many clustering techniques have been applied to clustering documents. Willett (1988) provided a survey on applying hierarchical clustering algorithms into clustering documents. Cutting *et al.* (1992) adapted various partition-based clustering algorithms to clustering documents. Two of the techniques are Buckshot and Fractionation. Buckshot selects a small sample of documents to pre-cluster them using a standard clustering algorithm and assigns the rest of the documents to the clusters formed. Fractionation splits the N documents into m buckets where each bucket contains N/m documents. Fractionation takes an input parameter ρ, which indicates the reduction factor for each bucket. The standard clustering algorithm is applied so that if there are n documents in each bucket, they are clustered into n/ρ clusters. Now each of these clusters is treated as if they were individual documents and the whole process is repeated until there are only K clusters.

A relatively new technique was proposed by Zamir and Etzioni (1998). They introduced the notion of phrase-based document clustering. They used a generalized suffix-tree to obtain information about the phrases and used them to cluster the documents.

As far as its application to IR is concerned, cluster analysis has been used both for term (or keyword) clustering and for document clustering. Term clustering (Wulfekuhler and Punch, 1997) is performed on the basis of the documents in which terms co-occur and it allows each term in a document, or query, to be replaced by the representation describing the cluster to which this term belongs. Application areas for term clustering include query expansion (Rijsbergen *et al.*, 1981), automatic thesaurus construction (Crouch and Yang, 1992) and thesaurus linking (Amba *et al.*, 1996). Peat and Willett (1991) have raised questions regarding the effectiveness of the use of keyword co-occurrence data.

Early experimentation showed that the effectiveness of searches based on document partitions is significantly inferior to that based on searches of the unclustered file (Salton, 1971). Most recent applications of partitioning methods to IR (Zamir and Etzioni, 1998) have also focused on efficiency aspects for on-line browsing tasks, rather than on the effectiveness of the methods.

Huang (1997) introduced k-modes, an extension to the well-known K-Means algorithm for clustering numerical data. By defining the mode notion for categorical clusters and introducing an incremental update rule for cluster modes, the algorithm preserved the scaling properties of K-Means. Naturally, it also inherited its disadvantages, such as dependence on the seed clusters and the inability to automatically detect the number of clusters.

Gibson *et al.* (1998) presented a method that encoded datasets into a weighted graph structure where the individual attribute values correspond to weighted vertices. It iterated multiple instances of these graphs

using a user defined combination operator to eventually converge to a fix point. The authors argued that upon reaching this fixed point, the weights of the basins can be used to partition the data points, yielding the final clusters. The dynamical systems approach underlying this method was problematic with regards to the type of detected clusters; the separation of attribute values by their weights was non-intuitive. Moreover, the number of basins required to attain a sufficiently large probability of convergence can be significant.

Guha *et al.* (1999) presented a clustering algorithm based on the number of links between tuples. The number of links intuitively captures the number of records that two records are both sufficiently similar to. This approach yields satisfactory results with respect to comparing attribute values that never co-occur in a single tuple. It heuristically optimizes a cluster quality function with respect to the number of links in an agglomerative hierarchical fashion. The base algorithm exhibits cubic complexity in the number of records, which makes it unsuitable for large datasets.

Ganti *et al.* (1999) introduced a combinatorial search based algorithm utilizing summary information of the dataset. Unlike earlier algorithms it characterized the detected categorical clusters. The algorithm relied on inter and intra attribute summaries that are assumed to fit into main memory for most categorical datasets. It first computed cluster projections onto the individual attributes. To reduce the complexity of this step, the authors assumed the existence of a distinguishing number K that represents the minimum size of the distinguishing sets which are attribute value sets that uniquely occur within only one cluster. The distinguished sets are then extended to cluster projections. Finally, cluster projections could be combined to clusters candidates over multiple attributes which are validated against the original dataset. The distinguished sets rely on the assumption that clusters are uniquely identified by a core set of attribute values that occur in no other cluster. While this assumption may hold true for many real world datasets, it is unnatural and unnecessary for the clustering process. Moreover, it is desirable to choose K as low as computationally possible in order to detect all clusters. A small K, however, entails a large number of candidate cluster projections on the individual attributes that lead to a combinatorial explosion in the number of final clusters.

The cluster projections on single attributes that generated are used in its extension phase to generate cluster candidates of higher dimensionality that are then validated on the actual dataset. In this approach to this end selected initial one dimensional candidate C^1 all cluster projections c_1 on the first attribute. Candidates in subsequent C^{k+1} are generated by combining each $(c_1,...,c_K) \in C^K$ with all cluster projections C^{K+1} on attribute A^{K+1}. If for all $1 \le i \le K$ (c_i, c_{K+1}), is a cluster projection on (A_i, A_{k+1}), $(c_1, \ldots, c_{K+1})$ is added to the candidate set C^{K+1}. The candidates are validated by scanning the original dataset and counting the support of each candidate.

Zhang *et al.* (2000) pointed out that the lack of a definite convergence is one of shortcomings and proposed a similar method that is guaranteed to converge. However, for both methods, the combination operator, as well as local modification operations are left to the user to find depending on the concrete data. Finally, the post-processing required to generate the actual clusters from the basin weights upon reaching the fix point is non-trivial and impacts the detected clusters. The clusters identified were shown to be incomplete in cases of overlapping cluster projections.

The algorithm introduced by Barbara *et al.* (2002) was based on the idea of entropy reduction within the generated clusters. It first bootstraps itself using a sample of maximally dissimilar points from the dataset to create initial clusters. The remaining points are then added incrementally. This approach was highly dependent on the order of selection. To mitigate this dependency, the authors propose to remove the worst fitting points at defined times during the execution and re-clustering them.

Cristofor and Simovici (2002) presented another approach based on cluster entropy measures for categorical attributes. Starting from a seed clustering, it uses GAs with crossover and mutation operators to heuristically improve the purity of the generated clusters. The quality of the resulting clusters depends on a prior knowledge of the importance of the individual attributes toward the natural clustering.

There are earlier works that apply GA and evolutionary programming to clustering. Some of them deal with clustering a set of objects by assuming that the appropriate value of K is known (Mertz and Zell, 2002). Sarkar *et al.* (1997) proposed an evolutionary programming-based clustering algorithm that groups a set of data into an optimum number of clusters. It is based on the well known K-Means algorithm. They use two objective functions that are minimized simultaneously: one gives the optimum number of clusters, whereas the other leads to proper identification of each cluster's centroids. Casillas *et al.* (2000) used only one objective function at the same time both aspects of the solution are calculated: an approximation to the optimum. K value and the best grouping of the objects into these K clusters. Makagonov *et al.* (2002) discussed other heuristics to split the dendrite in an optimal way without fixing the number of clusters.

Cui and Potok (2005) proposed a PSO based hybrid document clustering algorithm. The PSO clustering algorithm performs a globalized search in the entire solution space. In the experiments, they applied the PSO, K-Means and a hybrid PSO+K-Means clustering algorithm on four different text document datasets. The results illustrated that the hybrid PSO algorithm can generate more compact clustering results than the K-Means algorithm.

Fun and Chen (2005) introduced an evolutionary PSO learning-based method to optimally cluster N data points into K clusters. The hybrid PSO and K-Means, with a novel alternative metric algorithm are called Alternative KPSO-clustering method. This is developed to automatically detect the cluster centers of geometrical structure data sets. In this algorithm, the special alternative metric is considered to improve the traditional K-Means clustering algorithm to deal with various sets of data.

Csorba and Vajk (2006) presented a novel topic based document clustering technique where there is no need to assign all the documents to the clusters. Under such conditions the clustering system can provide a much cleaner result by rejecting the classification of documents with ambiguous topic. This is achieved by applying a confidence measurement for every classification result and by discarding documents with a confidence value less than a predefined lower limit. This means that the system returns the classification for a document only if it feels sure about it. If not, the document is marked as unsure. Beside this ability the confidence measurement allows the use of a much stronger term filtering, performed by a novel, supervised term cluster creation and term filtering algorithm, which is presented in this paper as well.

Jing *et al.* (2007) presented a new k-means type algorithm for clustering high-dimensional objects in sub-spaces. In high-dimensional data, clusters of objects often exist in subspaces rather than in the entire space. For example, in text clustering, clusters of documents of different topics are categorized by different subsets of terms or keywords. The keywords for one cluster may not occur in the documents of other clusters. This is a data sparsity problem faced in clustering high-dimensional data. In the new algorithm, The proposed k-means clustering calculates a weight for each dimension in each cluster and use the weight values to identify the subsets of important dimensions that categorize different clusters. This is achieved by including the weight entropy in the objective function that is minimized in the k-means clustering process. An additional step is added to the k-means clustering process to automatically compute the weights of all dimensions in each cluster.

Sun *et al.* (2008) developed a novel hierarchal algorithm for document clustering. They used cluster overlapping phenomenon to design cluster merging criteria. The system computes the overlap rate in order to improve time efficiency and the veracity and the line passed through the two cluster's center instead of the ridge curve. Based on the hierarchical clustering method it used the Expectation-Maximization (EM) algorithm in the Gaussian mixture model to count the parameters and make the two sub-clusters combined when their overlap is the largest.

Cao *et al.* (2008) presented a document clustering based on named entities as objectives into fuzzy document clustering, which are the key elements defining document semantics and in many cases are of user concerns. Traditional keyword-based document clustering techniques have limitations due to simple treatment of words and hard separation of clusters. First, the traditional keyword-based vector space model is adapted with vectors defined over spaces of entity names, types, name-type pairs and identifiers, instead of keywords. Then, hierarchical fuzzy document clustering can be performed using a similarity measure of the vectors representing documents. For evaluating fuzzy clustering quality, they proposed a fuzzy information variation measure to compare two fuzzy partitions.

Premalatha and Natarajan (2008a, b) introduced a hybrid method using PSO and GA for document clustering. In this algorithm, both PSO and GA are run in parallel. If the gbest particle stagnates, it can be replaced by a new particle. The new particle is generated by performing crossover operation on chromosome with gbest particle. It improves the diversity of the population. Premalatha and Natarajan (2008a) proposed Binary PSO with Local Search for Document Clustering. For goodness of fitness m% of particles are chosen by roulette wheel selection and the local search K-means algorithm is applied on those particle.

Premalatha and Natarajan (2009a, b) presented a procreant PSO algorithm for document clustering. This algorithm is a hybrid of Particle Swarm Optimization and Genetic Algorithm, a population-based heuristic search technique, which can be used to solve combinatorial optimization problems, modeled on the concepts of cultural and social rules derived from the analysis of the swarm intelligence (PSO) and also based on crossover and evolution (GA). In standard PSO the non-oscillatory route can quickly cause a particle to stagnate and also, it may prematurely converge on suboptimal solutions that are not even guaranteed to local optimal solution. They proposed modification strategy for PSO algorithm and applied to the document corpus. The strategy adds

reproduction using crossover when stagnation in the movement of the particle is identified and carries out local search to improve the goodness of fit. Reproduction has the capability to achieve faster convergence and better solution. Premalatha and Natarajan (2009c) proposed a document clustering based on Genetic Algorithm with Simultaneous and ranked mutation. In this study, more than one mutation operator is applied. The ratio of the mutation operators is based on the rank of the offspring generated by the mutation operator.

Muflikhah and Baharudin (2009) proposed a method that integrates the information retrieval method and document clustering as concept space approach. It used the Latent Semantic Index (LSI) approach which used Singular Vector Decomposition (SVD) or Principle Component Analysis (PCA). LSI method is used to reduce the matrix dimension by finding the pattern in document collection with refers to concurrent of the terms. Each method is implemented to weight of term-document in vector space model (VSM) for document clustering using fuzzy c-means algorithm.

CONCLUSIONS

Clustering is the ability of finding groups in data. Document clustering finds overall similarity among groups of documents. It is often applied to the huge document corpus in order to make a partition based on their similarity. It can lead to more effective retrieval than linear search which ignores the relationships that exist between documents It is used to improve the precision and recall from query. Document clustering is very useful to retrieve information application in order to reduce the consuming time and get high precision and recall.

REFERENCES

Aggarwal, C.C., C.S. Gates and P.S. Yu, 1999. On the merits of building categorization systems by supervised clustering. Proceedings of the 5th Conference on ACM Special Interest Group on Knowledge Discovery and Datamining, Aug. 15-18, San Diego, California, United States, pp: 352-356.

Allan, J., A. Leuski, R. Swan and D. Byrd, 2001. Evaluating combinations of ranked lists and visualizations of inter-document similarity. Int. J. Inform. Process. Manage., 37: 435-458.

Amba, S., N. Narasimhamurthi, K.C. O'Kane and P.M. Turner, 1996. Automatic linking of thesauri. Proceedings of the 19th Annual ACM Special Interest Group on Information Retrieval Conference, Aug. 18-22, Zurich, Switzerland, pp: 181-186.

Arabie, P. and L.J. Hubert, 1996. An Overview of Combinatorial Data Analysis. In: Clustering and Classification, Arabie, P., L.J. Hubert and G.D. Soete (Eds.). World Scientific Publishing Co., New Jersey.

Barbara, D., Y. Li and J. Couto, 2002. COOLCAT: An entropy-based algorithm for categorical clustering. Proceedings of the 11th International Conference on Information and Knowledge Management, Nov. 4-9, McLean, Virginia, USA., pp: 582-589.

Ben-Dor, A. and Z. Yakhini, 1999. Clustering gene expression patterns. Proceedings of the 3rd Annual International Conference on Computational Molecular Biology, 1999, Lyon, France, pp: 11-14.

Berkhin, P., 2005. A survey on page rank computing. Internet Mathematics, 2: 73-120.

Buckley, C. and A.F. Lewit, 1985. Optimizations of inverted vector searches. Proceedings of the ACM Special Interest Group on Information Retrieval Conference, June 5-7, Montreal, Quebec, Canada, pp: 97-110.

Cadez, I., P. Smyth and H. Mannila, 2001. Probabilistic modeling of transaction data with applications to profiling, visualization and prediction. Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Aug. 26-29, San Francisco, California, pp: 37-46.

Cao, T.H., H.T. Do, D.T. Hong and T.T. Quan, 2008. Fuzzy named entity-based document clustering. Proceedings of IEEE International Conference on Fuzzy Systems, June 1-6, Hong Kong, pp: 2028-2034.

Casillas, M.T., G. de Lena and R. Martinez, 2000. Document clustering into an unknown number of clusters using a genetic algorithm. Trans. Neural Networks, 11: 586-600.

Chakrabarti, S., 2003. Mining the Web: Discovering Knowledge from Hypertext Data. Morgan Kaufmann Publishers, California.

Cooley, R., B. Mobasher and J. Srivastava, 1999. Data preparation for mining world wide web browsing. J. Knowledge Inform. Syst., 1: 5-32.

Cristofor, D. and D. Simovici, 2002. An information-theoretical approach to clustering categorical databases using genetic algorithms. Proceedings of 2nd Society for Industrial and Applied Mathematics Industrial Data Mining, Workshop on Clustering High Dimensional Data, (SAIM IDMWCHDD'02), USA., pp: 37-46.

Croft, W.B., 1978. Organizing and searching large files of document descriptions. Ph.D. Thesis, Churchill College, University of Cambridge.

Csorba and Vajk, 2006. Term clustering and confidence measurement in document clustering. Proceedings of IEEE International Conference on Computational Cybernetics, Aug. 20-22, Budapest, pp: 1-6.

Cui, X. and T.E. Potok, 2005. Document clustering analysis based on hybrid PSO+K-means algorithm. J. Comput. Sci., 1: 27-33.

Cutting, D., J. Carger, J. Pedersen and J. Tukey, 1992. Scatter/gather: A cluster-based approach to browsing large document collections. Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, June 21-24, Copenhagen, Denmark, pp: 318-329.

Dempster, A.P., N.M. Laird and D.B. Rubin, 1977. Maximum likelihood from incomplete data via the EM algorithm. J. Stat. Soc., 39: 1-38.

Dhillon, I., J. Fan and Y. Guan, 2001. Efficient Clustering of Very Large Document Collections. In: Data Mining for Scientific and Engineering Applications, Grossman, R.L., C. Kamath, P. Kegelmeyer, V. Kumar and R.R. Namburu (Eds.). Kluwer Academic Publishers, USA.

Duda, R. and P. Hart, 1973. Pattern Classification and Scene Analysis. John Wiley and Sons, New York.

Everitt, B., 1993. Cluster Analysis. 3rd Edn., Edward Arnold, London, UK.

Fasulo, D., 1999. An analysis of recent work on clustering algorithms. Technical Report, University of Washington.

Frakes, W.B. and R. Baeza-Yates, 1992. Information Retrieval Data Structures and Algorithms. Prentice-Hall Inc., Englewood Cliffs, New Jercy.

Fukunaga, K., 1990. Introduction to Statistical Pattern Recognition. Academic Press, San Diego, CA.

Fun, Y. and C.Y. Chen, 2005. Alternative KPSO-clustering algorithm. J. Sci. Eng., 8: 165-174.

Ganti, V., J. Gehrke and R. Ramakrishnan, 1999. CACTUS: Clustering categorical data using summaries. Proceedings of the International Conference on ACM Special Interest Group on Knowledge Discovery and Data Mining, Aug. 15-18, San Diego, California, United States, pp: 73-83.

Gersho, A. and R.M. Grey, 1992. Vector Quantization and Signal Compression. Kluwer Academic Publishers, Norwell, MA., ISBN: 0792391810, pp: 760.

Ghosh, J., 2002. Scalable Clustering Methods for Data Mining. In: Handbook of Data Mining, Nong, Y. (Ed.). Erlbaum, Lawrence.

Gibson, D., J. Kleinberg and P. Raghavan, 1998. Clustering categorical data: An approach based on dynamical systems. Proceedings of the 24th International Conference on Very Large Databases, Aug. 24-27, New York, USA., pp: 311-322.

Guha, S., R. Rastogi and K. Shim, 1999. ROCK: A robust clustering algorithm for categorical attributes. Proceedings of the 15th International Conference on Data Engineering, Mar. 23-26, Sydney, Australia, pp: 512-521.

Han, J., M. Kamber and A.K.H. Tung, 2001. Spatial Clustering Methods in Data Mining: A Survey. In: Geographic Data Mining and Knowledge Discovery, Miller, H. and J. Han (Eds.). Taylor and Francis, USA.

Hartigan, J.A., 1975. Clustering Algorithms. 4th Edn., John Wiley, New York.

Hearst, M.A. and J.O. Pedersen, 1996. Re-examining the cluster hypothesis: scatter/gather on retrieval results. Proceedings of the 19th Annual Conference on ACM Special Interest Group on Information Retrieval, Aug. 18-22, Zurich, Switzerland, pp: 76-84.

Huang, Z., 1997. A fast clustering algorithm to cluster very large categorical data sets in data mining. Proceedings of ACM Workshop on Research Issues on Data Mining and Knowledge Discovery, (WRIDMKD'97), Tucson, AZ., pp: 526-529.

Jain, A.K. and P.J. Flynn, 1966. Image Segmentation Using Clustering. IEEE Press, USA., pp: 65-83.

Jain, A.K. and R.C. Dubes, 1988. Algorithms for Clustering Data. Prentice-Hall Inc., Englewood Cliffs, NJ., USA.

Jain, A.K., M.N. Murty and P.J. Flynn, 1999. Data clustering: A review. ACM Comput. Surveys, 31: 264-323.

Jardine, N. and C.J. van Rijsbergen, 1971. The use of hierarchical clustering in information retrieval. Int. J. Inform. Storage Retrieval, 7: 217-240.

Jing, L., M.K. Ng and J.Z. Huang, 2007. An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data. IEEE Trans. Knowledge Data Eng., 19: 1026-1041.

Kaufman, L. and P.J. Rousseeuw, 1990. Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley and Sons, New York.

Koller, D and M. Sahami, 1997. Hierarchically classifying documents using very few words. Proceedings of the 14th International Conference on Machine Learning, July 8-12, Nashville, Tennessee, pp: 170-178.

Larsen, B. and C. Aone, 1999. Fast and effective text mining using linear-time document clustering. Proceedings of the 5th International Conference on ACM Special Interest Group on Knowledge Discovery and Data Mining, Aug. 15-18, San Diego, CA, pp: 16-22.

Makagonov, P., M. Alexandrov and A. Gelbukh, 2002. Selection of Typical Documents in a Document Flow. WSEAS Press, USA., pp: 197-202.

Massart, D. and L. Kaufman, 1983. The Interpretation of Analytical Chemical Data by the Use of Cluster Analysis. John Wiley and Sons, New York.

Mertz, P. and A. Zell, 2002. Clustering gene expresion profiles with memetic algorithms. Lecture Notes Comput. Sci, 2439: 811-820.

Mirkin, B., 1996. Mathematic Classification and Clustering. Kluwer Academic Publishers, USA.

Muflikhah, L. and B. Baharudin, 2009. Document clustering using concept space and cosine similarity measurement. Proceedings of International Conference on Computer Technology and Development, Nov. 13-15, ICMC, pp: 58-62.

Peat, H.J. and P. Willett, 1991. The limitations of term co-occurrence data for query expansion in document retrieval systems. J. Am. Soc. Inform. Sci., 42: 378-383.

Porter, M.F., 1980. An algorithm for suffix stripping. Program, 14: 130-137.

Preece, S.E., 1973. Clustering as an output option. Proc. Conf. Am. Soc. Inform. Sci., 10: 189-190.

Premalatha, K. and A.M. Natarajan, 2008a. Hybrid PSO-GA with crossover for document clustering. Int. J. Mathematical Sci. Eng. Appl., 2: 185-198.

Premalatha, K. and A.M. Natarajan, 2008b. Binary PSO for document clustering with local search. Int. J. Eng. Res. Ind. Appl., 1: 209-224.

Premalatha, K. and A.M. Natarajan, 2009a. Genetic algorithm for document clustering based on simultaneous and ranked mutation. J. Modern Applied Sci., 3: 35-42.

Premalatha, K. and A.M. Natarajan, 2009b. Dimension reduction for indexing structure of documents based on concept hierarchy. J. Arts Commerce Comput. Sci. Technol., 7.

Premalatha, K. and A.M. Natarajan, 2009c. Procreant PSO for fastening theconvergence to optimal solution in the application of document clustering. Curr. Sci., 96: 137-143.

Rijsbergen, C.J., D.J. Harper and M.F. Porter, 1981. The selection of good search terms. IP M, 17: 77-91.

Salton, G., 1971. The SMART Retrieval System-Experiment in Automatic Document Processing. Prentice-Hall, Englewood Cliffs, New Jersey.

Sander, J., M. Ester, H.P. Kriegel and X. Xu, 1998. Density-based clustering in spatial databases: The algorithm GDBSCAN and its applications. Int. J. Data Mining Knowledge Discovery, 2: 169-194.

Sarkar, M., B. Yegnanarayana and D. Khemani, 1997. A clustering algorithm using an evolutionary programming-based approach. Pattern Recognition Lett., 18: 975-986.

Sparck, K.J., 1972. A statistical interpretation of term specificity and its application in retrieval. J. Documentation, 28: 11-21.

Steinbach, M., G. Karypis and V. Kumar, 2000. A comparison of document clustering techniques. Proceedings of the 6th ACM SIGKDD World Text Mining Conference, (TMW'00), Boston, MA., pp: 1-2.

Sun, H., Z. Liu and L. Kong, 2008. Document clustering method based on hierarchical algorithm with model clustering. Proceedings of International Conference on Advance Information Networking and Applications, March 25-28, Shantou University, pp: 1229-1233.

Willett, P., 1988. Recent trends in hierarchical document clustering: A critical review. J. Inform. Process. Manage., 24: 577-597.

Wulfekuhler, M.R. and W.F. Punch, 1997. Finding salient features for personal web page categories. Comput. Networks ISDN Syst., 29: 1147-1156.

Zamir, O. and O. Etzioni, 1998. Web document clustering: A feasibility demonstration. Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Aug. 24-28, ACM Melbourne, Australia, pp: 46-54.

Zhang, Y., A. Fu, C. Cai and P. Heng, 2000. Clustering categorical data. Proceedings of the International Conference on Data Engineering, (ICDE'00), IEEE Computer Society, Washington, DC., USA., pp: 305-305.