ORIENTAL JOURNAL OF COMPUTER SCIENCE & TECHNOLOGY
An International Open Free Access, Peer Reviewed Research Journal
www.computerscijournal.org

ISSN: 0974-6471
December 2011,
Vol. 4, No. (2):
Pgs. 379-385

An Approach to Improve Quality of Document Clustering by Word Set Based Documenting Clustering Algorithm

SANDEEP SHARMA, RUCHI DAVE and NAVEEN HEMRAJANI

Department of Computer Science and Engineering, Suresh Gyan Vihar University, Jaipur (India).

(Received: September 19, 2011; Accepted: September 24, 2011)

ABSTRACT

This paper presents a technique to improve the quality of Document Clustering based on Word Set Concept. The proposed Technique WDC (word set based document clustering), a clustering algorithm work with to obtain clustering of comparable quality significantly more efficiently more than the state of the art text clustering algorithm. The proposed WDC algorithms utilize the semantic relation ship between words to create concepts. The Word sets based Document Clustering (WDC) obtains clustering of comparable quality significantly more efficiently than state-of-art approach is efficient and give more accurate clustering result than the other methods.

Key words: Document clustering, Frequent concept, Word set.

INTRODUCTION

A large amount of text information has been generated and stored in text databases or digital data warehouses for years because the most natural form to store information is text.

The main purpose of this paper to proposed a document clustering algorithm named WDC (Word set based clustering), is designed to meet the above requirements for good text clustering algorithm.

The special feature of proposed WDC Algorithm is to Cluster the documents by using the word that co occur in sufficient number of documents. Each document in this approach corresponds to a transaction and each corresponds to an item.

Clustering is an unsupervised discovery process for separating unrelated data and grouping related data into clusters in a way to increase intra-cluster similarity and to decrease inter cluster similarity. A clustering of a data set is a splitting of the data set into a collection of subsets. These subsets are called clusters.

Data mining, also known as Knowledge Discovery in Databases (KDD), Data mining is a method of extracting interesting knowledge, such as rules, patterns, regularities, or constraints, from data in large databases.

Literature survey

Many clustering techniques have been proposed in the literature. Clustering algorithms are mainly categorized into hierarchical and partitioning methods. A hierarchical clustering method works by grouping data objects into a tree of clusters.

These methods can further be classified into agglomerative and divisive. Hierarchical clustering depending on whether the hierarchical decomposition is formed in a bottom-up or top-down fashion. K means and its variants are the well known partitioning methods which are used in several clustering applications.

Both hierarchical and partitioning methods do not really address the problem of high **Dimensionality in document clustering**

Frequent item set-based clustering method is shown to be a promising approach for high dimensionality clustering in recent literature.

It has been found that most of existing text clustering algorithms uses the vector space model which treats documents as bags of words.

Bisecting K-Means and FIHC algorithms are evaluated on the performance of text clustering.

Proposed methodology and performance evaluation metrics

This proposed methodology used to implement proposed Document Clustering method. Proposed method is cluster centered in that. The Cohesiveness of a cluster is measured directly using frequent closed word sets. Proposed Algorithm (Figure 1) first creates a normal document vector word based after creating the feature vector based on concepts, we utilize Apriori paradigm, designed originally for finding frequent item sets in market basket datasets, to find the frequent concepts from the feature vector. Then we formed the initial clusters by assigning one frequent concept to each cluster. WDC created a cluster for each closed word set. The algorithm process the initial clusters makes final clusters arranged in hierarchical structure.

Document Preprocessing

Preprocessing is a very important step since it can affect the result of a clustering algorithm. Preprocessing take a plain text document as input and produce output as a set of tokens to be used in the vector model. These steps typically consist of:

Filtering

The process which removes special characters and punctuation, which are not hold any discriminative power under the vector model.

Fig 1: Overview of Algorithm architecture

Tokenization

This step splits sentences into individual tokens, typically words.

Stop word Removal

A stop word is defined as a term which is not thought to convey any meaning as a dimension in the vector space..

Document Representation

The various clustering algorithm use the vector space model to represent each document. In this model, each document "d" is considered to be a vector in the term-space. In its simplest form, each document is represented by the term frequency (TF) vector

$$dtf = (tf_1, tf_2, \ldots\ldots, tf_m) \quad \ldots(1)$$

Where, tfi is the frequency of the i^{th} term in the document. A widely used refinement to this model is to weight each term based on its Inverse Document Frequency (IDF) in the document collection. This is commonly done by multiplying the frequency of each term i by log(N/df i), where N is the total number of documents in the collection, and df_i is the number of documents that contain the i^{th} term. This leads to the term Frequency Inverse Document Frequency (TF-IDF) representation of the document i.e.

$$dtfidf = (tf1\ \log(N/df1), tf_2 \log(N/df2), \ldots\ tfm\ \log(N/dfm)) \quad \ldots(2)$$

To account for documents of different lengths, the length of each document vector is normalized so that it is of unit length ((||dtfidf|| = 1), that is each document is vector in the unit hyper sphere.

To cluster similar documents together,

most of the traditional clustering algorithms require a similarity measure between two documents d1 and d2.Many possible measures are proposed in the literature, but the most common one is the cosine measure and it is defined below:

Similarity (d1, d2) =cosine (d1, d2) = (d1.d2)/ |d1|.|d2| ...(3)

Where **.** Represents the vector dot product and II represent the length of a vector.

Document Clustering

The proposed clustering method consists of the following phases: finding frequent closed word sets, creating initial clusters for each closed word set, making clusters disjoint using score function, building cluster tree, and tree pruning. The entire algorithm is described as following in step wise order.

Step 1

Create normal document vectors (word-based); each document is represented by vector of frequencies of remaining words after preprocessing within the document.

Step 2

Generate frequent closed word sets based on the threshold global support defined by the user.

Step 3

Construct initial clusters; construct a cluster for each global frequent closed word set. All documents containing this closed word set are included in the same cluster.

Step 4

Make clusters disjoint; assign a document to the best initial cluster. If there are several best clusters, the document is kept only in the cluster identified by longest label (in terms of the number of items). A cluster Ci is good for a document docj if there are many global frequent words in docj that appear in many documents in Ci. Assign each docj to the initial cluster Ci that has the highest score i as shown.

$$Score(C_i \bullet doc_j) = [\acute{}_x n(x) * \text{cluster support}(x)] - [\acute{}_x n(x') * \text{global support}(x') \quad ...(4)$$

Where

- x represents a global frequent word in docj and the word is also cluster frequent in Ci .
- x' represents a global frequent word in docj but the word is not cluster frequent in Ci.
- n(x) is the frequency of word x in the feature vector of docj.
- n(x') is the frequency of word x' in the feature vector of docj.

Step 5

Construct the tree; build a tree from bottom-up by choosing a parent for each cluster start from the cluster with the largestNumber of items in its cluster label Choosing a parent for cluster Ci in level *K* can be done by looking for all the clusters in level.

K-1 that have the cluster label being a subset of Ci's cluster label, then to determine the best parent of Ci, all the documents in the subtree of Ci are merged into a single Conceptual document doc(Ci) and then compute the score of doc(Ci) against each Potential parent. The potential parent with the highest score would become the parent of Ci. Finally remove any leaf cluster that does not contain any document.

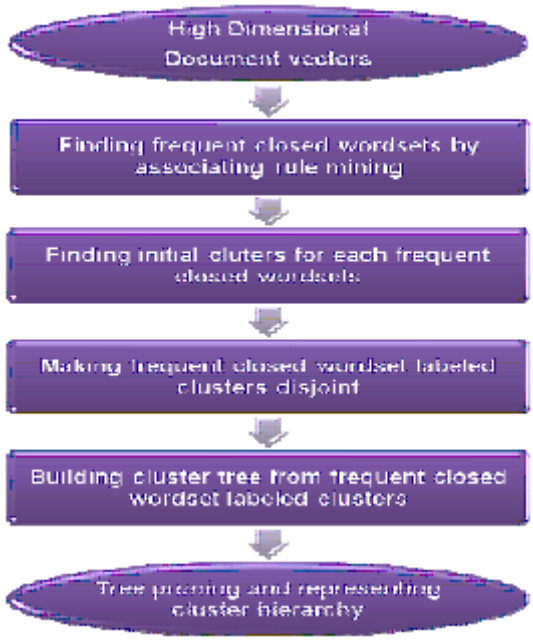

Fig. 2: Shows the algorithm as a flow chart

Step 6

Prune the tree, the aim of tree pruning is to merge similar clusters in order to produce a natural topic hierarchy for browsing and to increase the clustering accuracy. This step is divided into two phases: Child Pruning and Sibling Merging. But, Inter-cluster similarity between two clusters Ca and Cb is calculated by measuring the Similarity of *C*a to Cb. it is done by treating one cluster as a conceptual document (by Combining all documents in the cluster) and by calculating its score against the other cluster by using the following score equation:

$$Sim(Ca\ Cb) = \frac{score(c_a\ doc(c_b))}{\Sigma_x n(x) + \Sigma_{x'} n(x')} + 1 \quad ...(5)$$

Where:

- x represents a global frequent item in doc(Cb) and the item is also cluster frequent in Ca.
- x' represents a global frequent item in doc(Cb) but the item is not cluster frequent in Ca.
- n(x) is the frequency of item x in the feature vector of doc(Cb).
- n(x') is the frequency of x' in the feature vector of doc(Cb).

The inter similarity is defined

$$Inter\_Sim(Ca \leftrightarrow C_b) = [Sim(Ca\ Cb * Sim(C_b\ C_a)]^{1/2} \quad ...(6)$$

Where C_a and C_b are two clusters including their descendants: Sim(Ca Cb) is the similarity of Cb against Ca; Sim(Cb Ca) is the similarity of Ca against Cb.

The two phases of tree pruning are described as follow:

Child Pruning

It starts by scanning the tree in bottom-up order. During this scan, for any non-leaf node calculates inter-similarity between this node and its children; and each child with inter-similarity greater than 1 is pruned.

Sibling Merging

It merges similar clusters at level 1; the inter-similarity is calculated for each pair of clusters at level 1 and the cluster pair that has the highest inter-similarity is merged. The children of the two clusters become the children of the merged cluster. Sibling merging stops when, all inter-similarity between each pair are less than or equal to 1.

Performance Evaluation Metrics

The quality measure is the F-Measure a measure that combines the precision and recall ideas from information retrieval. It is a commonly used external measurement, which is employed to evaluate the accuracy of the produced clustering solutions. It is a standard evaluation method for both flat and hierarchical clustering structures. It produces a balanced measure of precision and recall. The recall, precision, and F-Measure for natural class Ki and cluster Cj are calculated as follows:

$$Recall(K_i, C_j) = \frac{nij}{|K|} \quad ...(7)$$

$$F(C) = \sum_{K_{1 \in k}} \frac{|K_i|}{|D|} \max_{C_j \in C}\{F(K_i, C_j)\} \quad ...(8)$$

Where nij is the number of member of class Kiinthe cluster Cj.The corresponding F-Measure F (Ki,Cj) is defined as:

$$F(K_i C_j) = \frac{2 * Recall(K_i, C_j) * Precesion(K_i, C_j)}{Recall(K_i, C_j) + Preision(K_i, C_j)} \quad ...(9)$$

F(Ki,Cj) represents the quality of cluster Cj in describing class Ki.While computing F (Ki,Cj) in a hierarchical structure all the documents in the subtree of Cj are considered as the documents in Cj.The overall F-measure,F(C) is the weighted sum of the maximum F measure of all the classes as defined below:

...(10)

Where, K denotes the set of natural classes;C denotes all clusters at all levels;|Ki|

Table 1. Summary Descriptions of Data sets

Date Set	Number of Documents	Number of Classes	Class Size	Average Class size	Number of Terms
Classic	7094	4	1033-3203	1774	12009
Hitech	2301	6	116-603	384	13170
Reo	1504	13	11-608	116	2886

denotes the number of documents in the class Ki; and |D| denotes the total number of documents in the data set. Taking the maximum of F(Ki; Cj) can be viewed as selecting the cluster that can best describe a given class, and F(C) is the weighted sum of the F-Measure of these best clusters. The range of F(C) is [0, 1]. A larger F(C) value indicates a higher accuracy of clustering.

Experimental Result

The experimental evaluation of proposed method. The comparison has been performed among proposed method and several other popular document clustering algorithms like agglomerative UPGMA, bisecting k-means and FIHC. The CLUTO-2.0 Clustering Toolkit has been used to generate the results of UPGMA and bisecting k-means. The gCLUTO-1.0 Graphical Clustering Toolkit has been used for visualizing the resulting clustering solution using tree, matrix, and an OpenGL-based mountain visualization.

Data Sets

All datasets used for evaluation in this thesis work are real life document data sets which have been widely used in document clustering research. They are heterogeneous in terms of document size, cluster size, number of classes, and document distribution. Their general characteristics are summarized in Fig. 3.1. The smallest of these data sets contained 1,504 documents and the largest contained 7,094 documents. The Classic data set was combined from the four classes CACM, CISI, CRAN, and MED of computer science, information science, and aerodynamics and medical articles abstracts.

Hitech data set was derived from the San Jose Mercury newspaper articles that are distributed as part of the TREC collection.

Data set Re() was extracted from newspaper article.

Experimental Results

All Experiments have been performed on Pentium 2.8 GHz Processor and 1 GB RAM based personal computer with MicrosoftWindows XP operating systems with service Pack 2. The proposed algorithm, i.e., Wordset based Document Clustering (WDC), and its competitors are evaluated in terms of accuracy, sensitivity to parameters, efficiency and scalability.Proposed method has also compared with another frequent itemset-based algorithm in term of the clustering quality measured by F-Measure. Fig. 3.2 shows the F-measure values for all the five algorithms with different user specified Number of clusters.

The highlighted results show the best algorithm for the specified document dataset. It can be observed that WDC has worked better on all the datasets. F-Measure Results of Classic, Hitech and Re0 dataset with different numbers of clusters. Proposed clustering method is robust enough to produce consistently high quality clusters for a wide range of number of clusters.

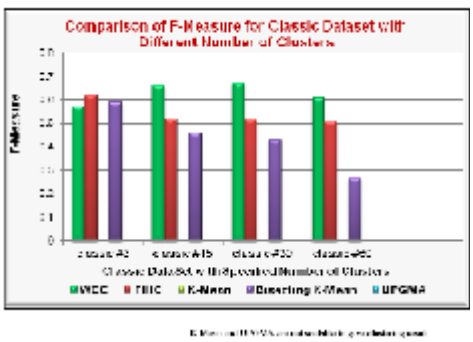

Fig. 3(1): F-measure results comparisons with classic data sets

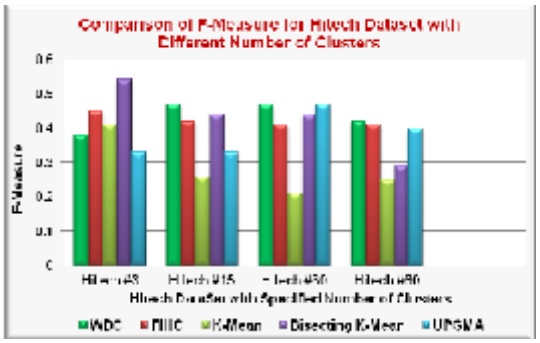

Fig. 3(2): F-measure results comparisons with Hitech datasets

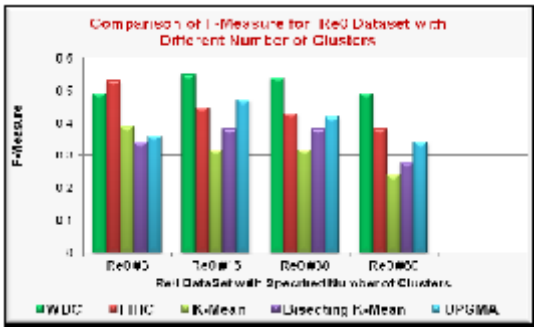

Fig. 3(3): F-measure results comparisons with Reo datasets

Data set	Number of clusters	Over all F Measure				
		WDC	FIHC	K Means	Bi - kmeans	UPGMA
Classic	3	0.57	0.62	NA	0.59	NA
	15	0.66	0.52	NA	0.46	NA
	30	0.67	0.52	NA	0.43	NA
	60	0.61	0.52	NA	0.27	NA
Hitech	3	0.38	0.45	0.41	0.54	0.33
	15	0.47	0.42	0.26	0.44	0.33
	30	0.47	0.41	0.21	0.44	0.47
	60	0.42	0.41	0.25	0.29	0.4
Reo	3	0.49	0.53	0.39	0.34	0.36
	15	0.55	0.45	0.32	0.38	0.47
	30	0.54	0.43	0.32	0.38	0.41
	60	0.49	0.38	0.24	0.28	0.34

UPGMA is not scalable for large data sets like Classic UPGMA fails to provide a clustering solution even after it has consumed all of the main memory. Hence, some experimental results could not be generated for UPGMA.

CONCLUSION

Cluster analysis examines unlabeled data, by either constructing a hierarchical structure, or by forming a set of groups according to a prespecified number. This process includes a series of steps, ranging from preprocessing and algorithm development, to solution validity and evaluation. Each of them is tightly related to each other and exerts great challenges to the scientific disciplines. Here, the focus on the clustering algorithms is placed and a wide variety of approaches appearing in the literature are analyzed.

Future Scope and Recommendations

Future study on document clustering using frequent closed word sets has the following possible avenues:

1. The proposed algorithm can be modified for getting the clustering results of documents other than English language.
2. The proposed algorithm may incorporate the modern natural language processing technique like Latent Semantic Indexing, Independent Component Analysis etc to improve the accuracy.

REFERENCES

1. J. Han and M. Kimber., Data Mining: Concepts and Techniques. Morgan Kaufmann (2000).
2. Jain, A.K, Murty, M.N., and Flynn P.J., Data clustering: a review. ACM Computing Surveys, **31**(3): 264-323 (1999).
3. M. Steinbach, G. Karypis, and V. Kumar. A comparison of document clustering techniques. KDD Workshop on Text Mining 00 (2000).
4. P. Berkhin., Survey of clustering data mining techniques [Online]. Available: http://www.accrue.com/products/rp_cluster_review.pdf (2004).
5. Xu Rui., Survey of Clustering Algorithms. IEEE Transactions on Neural Networks, **16**(3): 634-678 (2005).
6. Miller G., Wordnet: A lexical database for English. CACM, **38**(11): 39-41 (1995).
7. L. Zhuang, and H. Dai., A Maximal Frequent Itemset Approach for Document Clustering. Computer and Information Technology, CIT. The Fourth International Conference, 970-977 (2004).
8. R. C. Dubes and A. K. Jain., Algorithms for Clustering Data. Prentice Hall college Div, Englewood Cliffs, NJ, March (1998).
9. "A Frequent Concepts based document clustering" algo by Dr.Renu Dhir and Rekha Baghel International journal of comp application (2010).