International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 7 (2018) pp. 4822-4833
 https://dx.doi.org/10.37622/IJAER/13.7.2018.4822-4833

Text Document Retrieval through Clustering using Meaningful Frequent Ordered Word Patterns

Pushpalatha K.P.[1], G.Raju[2]

[1] *School of Computer Sciences, Mahatma Gandhi University, Kottayam, Kerala, PIN 686560, India.*

[2] *School of Information Science and Technology, Kannur University, Kannur, Kerala, PIN 670567, India.*

Abstract

Agglomerative Hierarchical Clustering (AHC) algorithm has two major limitations: one is its rigid nature and the other is the resulting globular clusters. This is due to the constraint of closest pair selection of clusters to re-cluster in a single iteration. In the proposed algorithm, this rigid nature is removed by a greedy approach to select more than one object for clustering in the same iteration using k-Nearest Neighbours approach. Instead of k, a similarity threshold is used to find neighbours. There is a biasness towards globular clusters that are produced when Normalised Google Distance (NGD) is used as the similarity measure. This limiation is reduced to a great extent using a modified NGD measure named as **Score**, modified by considering the local weightage between the features of different clusters. Many of the algorithms for text document retrieval are based on bag-of-words (BoW) approach. The sequence of the words are not given much importance in such algorithms. The bag of words representation used for these clustering is often unsatisfactory as it ignores relationships between co-occuring terms. WordNet and Association mining is used to enable the algorithm for meaningful document clustering giving more importance to relationships between co-occuring terms. WordNet is used to enhance the common concepts among documents. Association mining is used to construct feature set named as Frequent Ordered Word Patterns (FOWPs) from WordNet-enriched document data sets. New documents, constructed on FOWPs, are used for mining clusters. This hybrid hierarchical clustering approach when applied on the data sets formed by the newly formed documents, is found to give better results in terms of F-measure than that is given in the work taken as a bench mark for comparison purpose.

Keywords: Agglomerative hierarchical clustering, Apriori algorithm, Frequent ordered word patterns, Frequent word patterns, Score, WordNet, Similarity measure.

INTRODUCTION

The abundance of digitized data and information, easy access and high availability of any information from internet attracts the vast population towards the retrieval of information using internet. Managing text databases is a nontrivial challenge in the areas of text mining and especially in information retrieval. While submitting a query for getting an information, the search engine retrieves thousands or lakhs of document links, by clicking which, the user can read and find out the required information. Generally after 10 to 20 clicks a user gives up reading the documents in the current retrieval either finds documents of relevance to him or end up in set of documents not relevant to his query [1]. This is why the precision of the retrieval for a given query/concept is important and highlights the need of efficient algorithms for retrieving limited number of relevant documents and present them in the order of relevancy ranking. Text document clustering is a process of unsupervised and automatic grouping of text documents into clusters, with high intra cluster similarity and low inter cluster similarity [2].

Clustering is not only a grouping process, but also a query processing method. It is common to consider the analogy of common concept among the objects of a cluster to query and the members of a cluster to the retrieved documents containing that query, is meaningful. When the concept is a query, the cluster members become the relevant retrieved documents. Also many documents share common concepts and also deal with multiple topics. So when a query is submitted and if the resulting documents are selected using some suitable clustering algorithms, one of the clusters will give maximum number of documents with the query as the common concept. This clustering process makes the search engine more efficient and accurate.

A novel algorithm named as Clustering based on Frequent Ordered Word Patterns (CFOWP) is proposed with the objective of developing an algorithm with higher performance than that of the existing Agglomerative Hierarchical Clustering (AHC) algorithm for clustering text documents using frequent item set mining. The rigid nature of the AHC which always gives rise to a set of disjoint clusters, not benefitting the query processing application is the main motivation behind the proposal. The algorithm is rigid in the sense that only two clusters with the maximum similarity are always selected at any state of the problem, for merging. Second is the non-overlapped clusters. Most of the documents in a data set, deal with multiple topics/concepts and so overlapping of clusters must be allowed to a certain extent, to have better clustering output. Another concern is the use of Generalised Search Tree (GST) for building clusters by the existing algorithm [1]. But the construction of GST is very expensive. Hence the need of a simple algorithm with higher performance without the construction of GST is felt.

In the proposed algorithm, Frequent Ordered Word Patterns

(FOWPs) are the feature set used for comparing clusters for merging. Features are extracted after enriching the documents by identifying synonyms and hyperoyms and replacing them by a common word, possibly a hyperonym, using WordNet, a popular Electronic Lexical Database [3]. This enriching process increases the capabilities of the feature set in generating better clusters compared to the clusters generated otherwise.

A new similarity measure named Score is designed for comparing documents for clustering. It is a hybrid method that combines the exponential of Normalised Google Distance (eNGD) [4] and the Average net frequency of patterns (AvgNetFrq) present in documents within each cluster. The motivation to develop this hybrid measure is that the NGD determines the distance using globally existing features. The limitation is that it is not providing a local information to determine the distance between the documents in two different clusters. This limitation is overcome with this hybrid similarity measure.

The rest of this paper is organized as follows: Related works on text document clustering and document retrieval, mining of frequent word patterns, and the application of WordNet to enrich text documents are described in Section 2. Section 3 describes the method to find frequent word sequences from an enriched text database, and describes the CFOWP clustering algorithm. The experimental results of CFOWP and the performance comparison with other clustering algorithms CFWS and CFWMS are presented in Section 4. Section 5 which is followed by conclusion.

RELATED WORKS

Han and Kamber in [5] describes five main different approaches of clustering such as Partitioning methods, Hierarchical methods, Density-based methods, Grid-Based Methods and Model-Based Methods. In hierarchical methods, agglomerative Heirarchical Clustering (AHC) is a prominent algorithm.

Overlapping of clusters is not allowed in the basic AHC [6]. Unweighted Pair Group Method with Arithmetic mean (UPGMA) of AHC is reported to be the most accurate one in its category. The Bisecting k-means (BKM) algorithm is a top down method which is making use of k-means algorithm to split the initial single data set into two and this bisecting step is repeated until the desired number of clusters is obtained [6]. It is a combined approach of both partitioning and hierarchical and is more efficient compared to both of them and has been used in many clustering applications. Generally hierarchical methods produce spherical shaped clusters. For eg. BIRCH [7] and Chameleon [8]. Li et. al [1] proposed a clustering algorithm CFWS and compared the performance with BKM and FIHC algorithm [6]. In terms of accuracy (F-measure), algorithm CFWS outperforms both algorithms. The second algorithm CFWMS proposed by the same authors outperforms CFWS algorithm. They used k-mismatch concept on candidate clusters produced using GST (Generalised Suffix Tree) to compute the similarity matrix in CFWS and they used degree of overlapping as the measure to compute the matrix in CFWMS algorithm.

In the above algorithms, the bag of words (BOW) approach is used in document representation, extraction and selection of features. The frequency of each term in each document is considered as a basic feature weight and the similarity or dissimilarity are computed using any of the distance functions or similarity functions [9]. Recently another methodology is evolved by the researchers to cluster text documents. The algorithms of this approach, make use of terms in the document themselves to determine the similarity between the clusters instead of term frequency or term weights. Since the words and group of words in sequence represent concepts or topics dealt within a document, such frequent words or word sets are selected as significant features for clustering. Frequent 1-word set and 2-words set are used in some of the efficient algorithms [1]. The cohesiveness of the clusters are measured directly by using frequent word sets in Hierarchical Clustering algorithm (FIHC) [10][11]. It is based on the idea that documents in the same cluster are expected to share more frequent word sets than those in different clusters. The frequent term sets are generated using Apriori algorithm [1].

Han & Kamber [5] suggests the use of maximal and closed frequent item sets for efficient usage of resources. Ahonen-Myka et al. [12] and Cilibrasi, et al. [4] pointed out that the information retrieval performance is increased when the sequential aspect of word occurrences are considered for feature selection. They used the maximal frequent word sequence, which is a frequent word sequence not contained in any longer frequent word sequence. They claimed that maximal frequent word sequences provide a rich computational representation of the document, which makes feature retrieval easy and gives a human-readable description of the document. The Apriori algorithm introduced by Agrawal et al [13] is the simplest and the basic algorithm for generating frequent itemsets. Most of the researchers have applied the Apriori algorithm in order to extract frequent itemsets, in spite of its weaknesses.

Zheng Yu et.al. [14] have designed an advanced method in their paper 'Understanding Short Texts through Semantic Enrichment and Hashing'. In this work, they described how the short texts can be enriched by using the standard Knowledge Corpuses such as WordNet and Wikipedia etc. M.W. Berry and Murray Browne [15] in their book 'Understanding Search Engines, Mathematical Modelling and Text Retrieval', have explained the use of inverted files in information retrieval process. They underline the fact that if the space overhead is not critical in an application using large data sets, with limited set of attributes, where the number of terms present in documents, is very low, it is always efficient to use inverted files.

Three most widely used methods for finding similarity between categorical data are the k-mismatch concept between two term sets [7][1], Normalized Google Distance (NGD) [4] and Jaccard coefficient [16].

Cilibrasi and Vitanyi in 2007 [4] proposed a statistical index, based on Google page counts, showing the logical distance between a pair of frequent word patterns. The value of NGD ranges from 0 (min. distance) to ∞ (max. distance). Since the

maximum value reaches to ∞, it is found better to consider exp(NGD) as a possible similarity metric which ranges from 0 to 1. They presented a new theory of similarity between words and phrases based on information distance and Kolmogorov complexity. The technique is based on Google search engine and this can be applied in hierarchical clustering, classification, and language translation [4]. Huang et. al. [18] have explained the significance of concept (or word sequences) based similarity measure. Another measure is proposed by Lin et. al. [19] to gauge the similarity between two sets of documents for text classification and clustering problems. The results show that the performance obtained by the proposed measure is higher compared to similar other measures.

Text documents usually contain lexical items having similar significances, like the words "a board" and "a plank". These words can be put into one group {board, plank}. But "a board" can also indicate a group of people e.g., 'board of directors' and to disambiguate these homonymic significances "a board" can be grouped into {board, committee}. These types of words or phrases of similar significance, varies from the very specific one to the very general [20]. If such similar groups of words or phrases are replaced by a common word or phrase, the frequency and in turn the weight of the particular topic/concept represented by them, is increased. This will be very influencing in selecting the documents that are dealing with same concept and belonging to the same cluster. This process of identifying the words or word patterns of similar significance and managing them to increase the potential of features so that such documents are grouped into one cluster, is termed as enriching the text documents. For implementing this, a lexical knowledge repository software called WordNet is used.

DOCUMENT CLUSTERING

In this section the underlying theory and concepts of preparing text documents, feature extraction method, the proposed new similarity measure and a modified hierarchical clustering technique are described in detail.

3.1 Preprocessing

A text document is modelled as a set of alphabetically arranged Frequent Ordered Word Patterns (FOWPs) [9].

- **Definitions**

i) **Ordered word pattern**: It is a single entity of pattern of one or more words taken in the same sequence as given in the original text document.

ii) **Support**: It is the number of documents in which an ordered word pattern occurs.

iii) **Minimum Support of a word pattern (minSup):** It is defined as the minimum number of documents that contain the word pattern. This is used as a threshold for the support with the assumption that all word pattern with support below this threshold are not potential terms.

iv) **Frequent Ordered Word Pattern (FOWP):** It is an ordered word pattern for which the support is above or equal to minSup.

For extracting the patterns, a dictionary of **FOWPs** is created first. Patterns are stored I n the decreasing order of lengths and in alphabetical order. Each text document is then represented as a vector of available Frequent Ordered Word Patterns of 3/2/1 word/s. Before doing FOWP mining, the contents of each text file are scanned and the terms which are having length greater than 3 and not a stop word, are selected and stored as two single text files unstemmed and stemmed [21]. The unstemmed or normal file is used for enriching the documents with the help of WordNet. The words are stored in the same order as given in the original documents. For a **FOWP**, all its sub-sequences must also be frequent according [22][23]. If the two sub sequences s2 and s3 are frequent then the super sequence s1 is considered as frequent, provided the support of s1 is greater than or equal to minSup.

- **Document Model**

A modified vector space model is used to represent the documents. In this approach, the sequence of words in each document is highly important [24]. Each document vector is constructed based on the dictionary vector. Let p_1, p_2… p_k are frequent 3-word patterns; q_1, q_2,…,q_s are the frequent 2-word patterns; r_1, r_2,…,r_t are the frequent 1-word patterns, then a document is represented as

$d1=\{ p_1,\ldots,p_i,\ldots, p_k, q_{1,\ldots,}q_j,...,q_s , r_{1\ldots,}r_k,.. r_t \}$

The patterns are stored in the decreasing order of lengths. This type of storage of features increases the probability of selection of most relevant documents having the specific topic represented by the longer patterns first. For example consider a typical sentence: 'The public must be careful about the contagious diseases in rainy season'. The word patterns are:

3-words:- public careful contagious, careful contagious diseases, contagious diseases rainy, diseases rainy season;

2-words:- public careful, careful contagious, contagious diseases, diseases rainy, rainy season;

1-word:- public, careful, contagious, diseases, rainy, season.

Enriching the Text Data Set using WordNet

The contents of each text file in the data set, consists of nouns and verbs and also words representing related concepts or synonym sets (synsets) such as i) Synonyms: person→individual, someone, somebody, mortal, human, drunk person), ii) Hyperonyms: dog → animal - the generic term used to designate a whole class of specific. If such words are kept as such in the data sets, while mining, they are considered as different words or different topics. Identifying such synsets in a text file and replacing all such sets with a common representative word or phrase, will enhance the potential (support) of frequent word patterns in the corpus.

Algorithm for enriching the text data set:

Steps

1. Dictionary Creation: Scan the unstemmed text file of the

corpus and generate a Term Dictionary (TermDict) of alphabetically sorted unique single words.

2. Generate Synsets/Concepts: Generate synonym vector with all synonyms and a concept vector with all hperonyms using WordNet, for each possible word (only nouns and verbs) in TermDict, without redundancy.

The above algorithm not only considers the direct occurrence of a term but the synonyms or concepts are also searched for and hence the total number of supporting documents will be greater than what otherwise computed. In this way text file is semantically enriched with synonym / hyperonym counting process.

Feature Extraction: Frequent Ordered Word Patterns (FOWPs) Mining

Features (FOWPs) are extracted from the enriched text documents. A raw text document contains a number of sentences. An ordered set of one or more potential words in a sentence is called an Ordered Word Pattern. For eg., consider two consecutive sentences: "**Allen** is **studying Physics. Most** of the **students studying Physics** are **well placed.**". Here "Allen studying", "Allen studying **Physics",** "students studying", "students studying Physics", and "well placed" are ordered patterns, but "Physics Most" is not. Also "**studying Physics"** and **"Physics studying"** are not the same pattern as per the assumption of ordered word pattern or sequence. Potential features are frequent ordered word patterns where the support of a word pattern is greater than or equal to minSup [6] The frequent patterns are mined using Apriori algorithm. Here the itemsets are the word patterns. The algorithm employs an iterative approach called as level-wise search where k-itemsets are used to explore (k+1) itemsets. The enriched documents of the corpus are scanned and frequent ordered 1-word patterns, 2-word patterns and 3-word patterns are generated and stored as vectors according to the Apriori property based on the document model described in section 3. It also ensures the maximal and closed frequent patterns under the constraint of ordered sequence. Since the terms are selected in the same order as given in the original text document, the number of patterns obtained will be very small compared to the case when they are selected in random order. The patterns generated are meaningful and equivalent to queries that are used with search engines for document search and retrieval. The proposed algorithm assumes 5%, 10% and 15% as the values for minSup in three different experiments. An upper limit is also set since a word pattern, occurring in a large number of documents, is not contributing any information for characterising the documents, instead, their presence degrades the overall potential of features that are selected for clustering. The anti-monotone nature of the algorithm due to the Apriori property, reduces the size of candidate sets, leading to good performance gain. But it suffers from the nontrivial cost of L (length of longest pattern) number of scans of the original transaction data set.

The proposed Similarity measure – Score

- **Computation of Normalised Google Distance (NGD) and eNGD**

Cilibrasi & Vitanyi, motivated by Kolmogorov complexity [25], proposed a page-count-based similarity measure based on Google search engine, called the Normalized Google Distance (NGD) given as equation (1).

$$NGD(p,q) = \frac{\max\{\log(f(p)),\log(f(q))\} - \log(f(p,q))}{\log(M) - \min\{(\log(f(p)), log f(q))\}} \quad (1)$$

Here M is the total number of web pages indexed through Google search done with p and q, f(p) is the number of pages containing term p, f(q) is the number of pages containing term q and f(p, q) is the number of pages containing both terms p and q among these M pages. NGD searches terms where the words need not necessarily co-occur together in a document or sequentially ordered. The NGD metric is a distance metric and is unbounded, i.e., the range is [0, ∞]. Hence it is converted to another form eNGD(p,q) for which the value ranges from 0 to 1 as given in equation (2). 0 represents minimum similarity (maximum distance) and 1 represents maximum similarity (minimum distance).

$$eNGD(p,q) = e^{-2NGD(p,q)} \quad (2)$$

eNGD(p,q) satisfies the following:

If f(p), f(q) > 0 and f(p,q) = 0 then eNGD(p,q) =0.

1. eNGD(p,q) is 0 for f (p) = f (q) = 0;

2. eNGD(p,q) = 0 for f (p,q) = 0 and either or both f (p) > 0 and f (q) > 0; and

3. eNGD(p,q) is 1 otherwise.

Based on the above facts, eNGD is considered as a metric and is used to measure similarity between clusters.

- **Computation of eNGD between two clusters**

The eNGD between two clusters CL1 and CL2 is calculated as follows.

Algorithm for computing eNGD(CL1,CL2)

Assume clusters CL1 = (C_1, TC_1); CL2 = (C_2, TC_2), where C_1 is the set of documents and TC_1 is the set of FOWPs. If the whole data set is grouped into such t clusters then the data set is denoted as {(CL_1, TC_1), (CL_2, TCL_2),…,(CL_t, TCL_t)}

1. For each cluster CLi, $1 \le i \le t$
2. Compute s1= $|TC_i|$; //TCi is the pattern vector of cluster CL_i.//
3. For each cluster CLj, $1 \le j \le t$
4. Compute s2=$|TC_j|$; // TC_j is the sequence vector of cluster C_j.//
5. Compute eNGD(CLi, CLj); //The following equation (3) is used here//

$$eNGD(CL_i, CL_j) = \frac{1}{s1*s2}\sum_{i1=1}^{s1}\sum_{j1=i1+1}^{s2} eNGD(wp_{i1}, wp_{j1}) \quad (3)$$

where wp_{i1} and wp_{j1} are FOWPs are the sequences from TC_i and TC_j respectively.

The time complexity of the algorithm is O(s1s2). It does not depend on number of clusters t. In the proposed algorithm, local importance of documents is defined using a measure called **AvgNetFrq** as defined below**:**

- **Computation of AvgNetFrq**

AvgNetFrq(C_1,C_2) gives the local average effectiveness of documents due to the net weight of all word patterns that are present in the two clusters. This in turn contributes to the inter cluster similarity. If the average effectiveness is high then the inter-cluster similarity is high with a maximum of 1 else the similarity is low with a minimum value of 0.

Algorithm: AvgNetFrq(CL1,CL2)

//Assumptions about the clusters are same as those given in the case of eNGD computation above. //

1. For each cluster CLi, $1 \le i \le t$
2. L1 = length(TC_i)

   // L1 is the number of word patterns in the sequence vector TC_i of cluster CL_i. //
3. For each q_{i1} in TC_i $1 \le i1 \le L1$
4. For each 1-word and 2-words and 3-words sequences s, in q_{i1}

   Compute $Fri(k) = \sum^{L1}_{\substack{i1=1 \\ for\ all\ k \in CLi \\ s==UAllSeqs(id)}} TDFr_{id,k}$;

//TDFr(id,k) is the frequency of the $(id)^{th}$ FOWP in k^{th} document in frequency matrix of FOWPs. Fri(k) is the total frequency of all FOWPs in the k^{th} document of cluster CLi. UallSeqs is the dictionary of word patterns for the data set//

5. For each cluster CL_j, $i + 1 \le j \le t$
6. L2 = length (TC_j) // number of FOWPs in TC_j.//
7. For each FOWP q_{j1} $1 \le j1 \le L2$
8. For each 1-word, 2-words and 3-words sequences s, in q_{j1}
9. Compute $Frj(k) = \sum^{L2}_{\substack{j1=1 \\ k \in CLj \\ s==UAllSeqs(id)}} TDFr_{id,k}$

// Explanation to Frj(k) is similar to that of Fri(k).//

10. Compute Dev = abs(Fri – Frj);

// creates a vector of absolute deviations or distances of frequency vectors.//

11. LFr = length (Dev); // to determine the average of the normalised deviations.//
12. Calculate AvgNetFrq using (4)

$$AvgNetFrq(CL_i, CL_j) = 1 - [\sum_{k=1}^{n}(\frac{Dev(k)}{\max(Dev)})]/LFr \quad (4)$$

Division by LFr gives the average net influence of word patterns in the range [0, 1]. The second component actually represents the effective distance between the clusters since deviations are considered for the calculation. Hence that value is subtracted from 1 to get the similarity value. 1 is the maximum possible distance value. A value of 0 indicates no inter-cluster similarity and a 1 indicates the maximum similarity. The time complexity of the algorithm is O(L1L2). It does not depend on number of clusters t.

Since this measure considers total similarity of all pair's p and q, `it is very effective in identifying locally and globally co-occurring word sequences and the documents which contain those sequences. This enhances the cluster quality. The value of eNGD is in the range [0, 1] and that of AvgNetFrq is [0, 1]. Hence the value of Score varies in the range [0,2]. Thus insufficiency of information in NGD is reduced to a great extent by using the new similarity measure Score.

- **<u>Score – a new similarity measure</u>**

eNGD measures similarity based on a global perspective. If local importance is also incorporated, a better similarity measure may result. Score is defined as the sum of eNGD and AvgNetFrq. For any two clusters C1 and C2, the equation is shown in (5).

$$Score(C_1, C_2) = eNGD(C_1, C_2) + AvgNetFrq(C_1, C_2) \quad (5)$$

Clustering: a new algorithm for document clustering CFWOP is described below:

It combines average linked Agglomerative Hierarchical Clustering (AHC) technique with k-nearest neighbours approach, with the constraint of similarity threshold value (simT). The feature set used are the frequent ordered word patterns. New document vectors are designed using the selected FOWPs. These new documents are the inputs for the clustering algorithm. A document is represented by the cell array of FOWPs. The length of each document depends upon the number of FOWPs of that document. Each document stores FOWPs in the descending order of length of FOWPs.

CFOWP is summarised below:

Steps

1. Create a vector of clusters (First level in the hierarchy) where the size of the cluster is less than or equal to the threshold value sizeT. Each cluster is the set of one or more documents containing one FOWP from the dictionary vector of FOWPs.
2. Construct similarity matrix for the above set of clusters using the similarity measure Score.

3.a. Select the first unselected cluster as a reference/seed cluster and find out other clusters for which the similarity value from similarity matrix, is higher than that of the threshold value simT (nearest neighbours). Merge them one by one into the seed cluster under the constraint of sizeT and mark the merged clusters as selected. After each merge, the similarity matrix is updated based on the merged clusters. ie., size of the matrix is reduced by 1 and new similarity values are computed for rows and columns of merged clusters.

3.b. Select the next unselected cluster as seed cluster and repeat step 3.a until all clusters are selected and merged.

The clusters formed in this iteration becomes the clusters of the next higher level in the hierarchy.

4. With the above set of clusters, repeat steps 3.a and 3.b until desired number of clusters with desired sizes are generated.

Algorithm CFOWP (Pseudo code)

Assumptions:

1. The pair (Tcl, cl) represents each cluster, where Tcl is the word patterns/sequence vector and cl is the vector of document Ids in the cluster.

2. simMat(i,j) is the similarity matrix of current level clusters. Both i and j represents the various clusters, generated in the lower level of hierarchy. SM1 and SM2 are instances of simMat.

Steps:

Initialise k to a suitable number where k is the number of clusters required (optional);

Initialise minSup to 5% or 10% or 15% as per the requirement;

Initialise simT, to any value in [0.5, 1.5] and sizeT to a maximum cluster size threshold value; generally it is around n/k (by trial and error method)

1. Scan the data set to construct unstemmed and stemmed text data files after preprocessing (stop words removal and stemming).

2. Construct Synset and Concept vectors (both stemmed and unstemmed) by scanning the data set prepared in step 1.

3. Compute UAllSeqs and Sup;

// UAllSeqs is the dictionary vector of FOWPs, combining 1-word, 2-words and 3-words FOWPs and Sup is the corresponding support vector, constructed by merging the three support vectors Sup1,Sup2 and Sup3 generated in step 3. FOWPs are stored in UAllSeqs, from longest to shortest.//

4. Compute Doc;

// Doc is the set of all document vectors constructed using only the FOWPs. The length of each vector in Doc depends on the number of FOWPs in it.//

5. Compute TDFr.

// TDFr is a mxn term-document matrix which stores the frequency of i^{th} FOWP in j^{th} document for all $1 \leq i \leq m, 1 \leq j \leq n$ where m is the length of UAllSeqs and n is number of documents in the data set. Since the sparsity (=1- density) of this matrix is very high, it is converted into a sparse matrix and then stored in the memory.//

6. Initialise CL1 = ϕ; CL2= ϕ; TCL1= UAllSeqs; TCL2= ϕ;

7. For each p_i in UAllSeqs, $1 \leq i \leq m$, do steps a, b and c
// p_i is a FOWP, selected in decreasing order of length.//

{

a. C_i ={the set of all documents containing p_i}; //a cluster//

b. CL1 = [CL1, C_i]; // Inserting all non-empty clusters C_i in CL1.//

c. If $|C_i|$ <= sizeT then CL2= [CL2, C_i]; TCL2= [TCL2, p_i];

} // (CL2, TCL2) is used as initial set of clusters for merging. //

// step 7 is used to identify clusters in level 1 (lowest) of the hierarchy.//

8. Compute similarity matrix SM1 using CL1 and TC.

// SM1 is the static m x m similarity matrix where m=|TCL1|. //

9. Compute similarity matrix SM2 using CL2 and TCL2 and set SM2(i, i) = -1;

// SM2 is m1 x m1 similarity matrix where m1 = |CL2|. This is derived out of SM1. The -1 value is set to the similarity between the same clusters in SM1 and SM2, because they must not be selected as neighbours. //

10. Repeat steps a, b and c until the number of clusters generated is equal to k

{

while i < m

a. Find J = cluster_ indices (SM2(i, :) >= simT);

// i means i^{th} reference cluster to which all similar clusters j are merged//

b. Sort J to sJ in the descending order of similarity.

c. For each cluster j in sJ, do {

i) Merge cluster j to cluster i of CL2 and set cluster j to empty.

ii) Modify SM2 with respect to i and j as given below in steps a1 and b1.

a1) Move all rows after j^{th} row backward once and all columns after j^{th} column backward once and reduce size by 1.

b1) Recompute the values of i^{th} row and i^{th} column of SM2;

// Compute the average of similarities using (7) //

iii) If size(merged cluster i) >= sizeT then exit loop in step c else continue step c; }

// end For j //

d. Compute m1 = |TCL2|;

/ / m1 is updated with new size of TCL2.//

} // go to step 10.//

If number of resultant clusters is not limited to k, then the degree of overlap between any two clusters is computed using

(6). The value of Overlap varies from 0 to 1. If Overlap ≥ ovlpT (overlap threshold; typical value = 0.5) then merge the two clusters, otherwise repeat the process of calculating Overlap and merging, with remaining clusters until desired number of clusters is obtained. This results to a reduced set of clusters in CL2 (and TCL2).

$$Overlap(C1, C2) = \frac{|C1 \cap C2|}{|C1 \cup C2|} \quad (6)$$

- **Updating of Similarity Matrix**

Each cell of a similarity matrix (simMat) stores the inter cluster similarity value between any two clusters. The similarity is computed using (1) as described above. For a reference (seed) cluster, a set of other clusters with Score >= simT are identified first Using nearest neighbour approach. From this set, each cluster is merged with the reference cluster to form a new cluster. The cells corresponding to the row and column of the matrix represented by the row Id (Id of the cluster taken for merging), are then modified by computing the average similarities between the paired clusters. For example if the cluster pair taken for merging is 4 and 6 (i.e., 4th row and 6th column) then 6th cluster is merged into 4th cluster. The 6th row and 6th column of the matrix is eliminated by moving all further rows and columns backward once and reducing the size by 1. The cells corresponding to the intersection of 4th row and all other columns are now re-computed using AvgSim using (7). Similarly the cells corresponding to the intersection of 4th column and all other rows are re-computed using (7).

Assume $CL_i = (Ci, TCi)$ and $CL_j = (C_j, TC_j)$

$$AvgSim(CL_i, CL_j) = \frac{1}{|TC_i| * |TC_j|} \sum_{i1=1}^{|TC_i|} \sum_{j1=1}^{|TC_j|} SM1(i1, j1) \quad (7)$$

where CL_i and CL_j are the clusters to be merged, i and j represents i^{th} row and j^{th} column of the matrix SM1. TC_i and TC_j are the sequence vectors of CL_i and CL_j.

The time required for updating simMat after each merge is O(|TCi||TCj|).

The Time Complexity of the Clustering Algorithm CFOWP:

The construction of simMat has a time complexity of $O(m^2)$ where m is the length of simMat. The time complexity of algorithm of simMat construction (eNGD and AvgNetFrq) and simMat updating together is (O(s1s2)+O(L1L2)) + (O(|Tci||Tcj|)) which does not depend on input data size, since s1, s2, L1, L2, |TCi| and |TCj| are the lengths of TCi, the sequence vector of clusters. Since simMat is an upper triangular matrix, only half of the values are to be computed. i.e., the time actually is $m^2/2$ and the asymptotic time complexity of the clustering algorithm is $O(m^2)$.

EXPERIMENTS AND ANALYSIS OF RESULTS

Data Sets

We considered the work reported by Li et.al, [1], as bench mark for comparison purpose. They have proposed two clustering algorithms: CFWS and CFWMS. They experimentally proved that CFWMS outperforms CFWS. Hence the data set used for this work is almost same as that used by Li et. al. 11 data sets were chosen which includes: 5 data sets Re1 to Re5 are selected from Reuters21578 newswire corpus [26]. The documents for them are selected from the classes EXCHANGES, ORGS, PEOPLE and TOPICS of the Reuters21578 newswire corpus. 3 text data sets Ce1, Ce2 and Ce3 are selected based on the Cacm, Cran and Cisi classes of Classic text data corpus [27]. Remaining three data sets Se1, Se2 and Se3, are based on three query terms used in TREC 2004 conference [28][29]. They are extracted using Google search engine providing the query terms used by Li et. al.. The data set characteristics are given in Table 1.

Table 1. Characteristics of Data Sets

Data set	No.of Docs	No.of Class	Min.Class Size	Max.Class size	Avg class size	No.of unique terms	Avg doc length	No.of total terms
Re1	340	7	28	97	49	4075	185	63008
Re2	807	9	20	349	90	4949	102	82328
Re3	797	15	20	168	53	5686	117	92899
Re4	1761	31	20	349	57	8542	102	179938
Re5	4443	39	20	801	114	13021	87	387945
Ce1	262	4	56	144	66	1612	67	17499
Ce2	355	4	70	146	89	2125	54	19240
Ce3	178	3	35	117	59	1495	75	13377
Se1	200	3	44	92	67	2457	89	17886
Se2	200	3	20	98	67	2288	86	17142
Se3	200	3	18	152	67	2048	87	17393

All data sets are different in their size, number of classes, the class size etc. Documents for Re1 to Re5 were taken from Reuters-21578 Distribution 1.0 [30] and those for Ce1, Ce2, and Ce3 from the CISI, CRAN, and CACM abstracts of Classic data corpus Se1, Se2, and Se3 with 200 documents each, were collected using Google search engine by giving 3 queries 'marineVegitation', 'southAsianDeaths' and 'rapMusicViolence' specified in TREC 2004 conference proceedings.

- **System Configuration**

The coding for the experiments are done using Matlab R2009a programming features. All source data sets and intermediate data files generated are stored using flat text files. A desktop system with Intel Core 2 Duo Processor @2.20GHz each, 2GB Ram, and 64bit Windows 8.1 Pro- is used in experimenting the algorithms.

Clustering Experiments and Results

The proposed CFOWP clustering algorithm is run using the

11 data sets of which the characteristics are described in Table 1. Various performance measures are used to evaluate the performance of the new clustering algorithm CFOWP. A detailed description of various measures used for the evaluation of clustering algorithm is given in the following section.

Performance Evaluation Measures

The clustering algorithm is considered as an information retrieval problem solving method. In such a process, the precision and recall are given more importance than the rate of correctly classified objects. Each cluster obtained is considered as the result of a document retrieval process corresponding to a query. Precision and recall are the two important measures which indicates the quality of an information retrieval algorithm. So F-measure is used to determine the accuracy of the algorithm. It is the harmonic mean of precision and recall values of the process. So during testing, the precision and recall values are calculated for each cluster separately in separate class and that is used to compute the F-Measure of i^{th} class for j^{th} cluster [4]. It is calculated using (8)

$$F(i,j) = \frac{2*P(i,j)*R(i,j)}{P(i,j)+R(i,j)} \quad (8)$$

where the Precision P(i, j) is calculated using (9).

$$P(i,j) = \frac{n_{ij}}{n_j} \quad (9)$$

and the Recall, R(i, j) is calculated as

$$R(i,j) = \frac{n_{ij}}{n_i} \quad (10)$$

where n_i is the number of members of class i; n_j is the number of documents of cluster j and n_{ij} is the number of documents of class i in cluster j.

The F-measure F is given as

$$F = \sum_i \frac{n_i}{n} \max_j(F(i,j)) \quad (11)$$

Another measure used to express the performance of an algorithm is the Purity of clusters generated. The purity of a cluster is the fraction of the cluster corresponding to the largest class of documents assigned to that cluster. The purity of a cluster j is defined as in (12).

$$Purity(j) = \frac{1}{n_j} \max_i(n_{ij}) \quad (12)$$

where i denotes the class.

The cluster purity of the data set is computed using (13).

$$ClusterPurity = \frac{1}{n}\sum_{j=1}^{cl} \frac{1}{n_j} \max_i n_{ij} \quad (13)$$

Many researchers use entropy as a performance indicator of cluster quality. Entropy of a clustering, measures the homogeneity of the clusters generated. It quantifies how much disorder is remaining in clustering. Measuring entropy, is an efficient way to evaluate the goodness of a cluster set generated and it is computed using (14). Entropy of a good clustering must be very small. When there is no disorder in clustering, then entropy will be zero.

$$Entropy = -\sum_{i=1}^{k} \frac{n_{ij}}{n_j} \log(\frac{n_{ij}}{n_j}) \quad (14)$$

where k is number of classes. i denotes the class of a cluster, j denotes a cluster; n_{ij} is the number of members in the j^{th} class of i^{th} cluster.

Performance Based on F-measure

F–measure is used as a scale for measuring the performance of the clustering algorithm. Because the other measures such as percentage of correctly clustered members etc are specifying only how many relevant members are retrieved out of total relevant, that is the recall value only. When we consider the clustering problem in the context of information retrieval process, it is better to consider both precision and recall and F-measure is the harmonic mean of both precision and recall. This can reflect the ratio of maximum relevant objects retrieved out of minimum number of total retrieved objects. In order to determine the performance of the clustering system, the output data for F- measure and Cluster Purity and Entropy are collected and recorded at the end of each test run of the program.

- **Comparison of F-measures of CFOWP with CFWMS and CFWS**

Table 2 shows the computed output of F-measure and Cluster Purity for all the 11 data sets. The testing is also done using varying scales of minimum support (minSup). The values of F-measure obtained after scaling, are tabulated and shown in Table 3. The graph drawn based on the F-measure data from table 2 is shown as figure 1. The blue colored line corresponds to CFWS, the green colored line corresponds to CFWMS and the orange colored line corresponds to CFOWP algorithms. Except for one data set, the proposed algorithm gives a better performance than the CFWMS. It is clear from the graph that the algorithm CFOWP outperforms the algorithm CFWS for all data sets. This underlines the positive effect of adding local information to the value of eNGD in computing similarity between clusters for merging using the new measure Score.

Table 2. F-measures: CFWMS & CFWS vs CFOWP

Data sets	F-measure		
	CFWMS	CFWS	CFOWP
Re1	0.78	0.65	0.77
Re2	0.83	0.80	0.85
Re3	0.69	0.72	0.76
Re4	0.70	0.56	0.75
Re5	0.64	0.50	0.67
Ce1	0.66	0.54	0.80
Ce2	0.72	0.48	0.81
Ce3	0.83	0.55	0.79
Se1	0.80	0.69	0.72
Se2	0.81	0.73	0.81
Se3	0.86	0.81	0.85

International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 7 (2018) pp. 4822-4833
© Research India Publications. https://dx.doi.org/10.37622/IJAER/13.7.2018.4822-4833

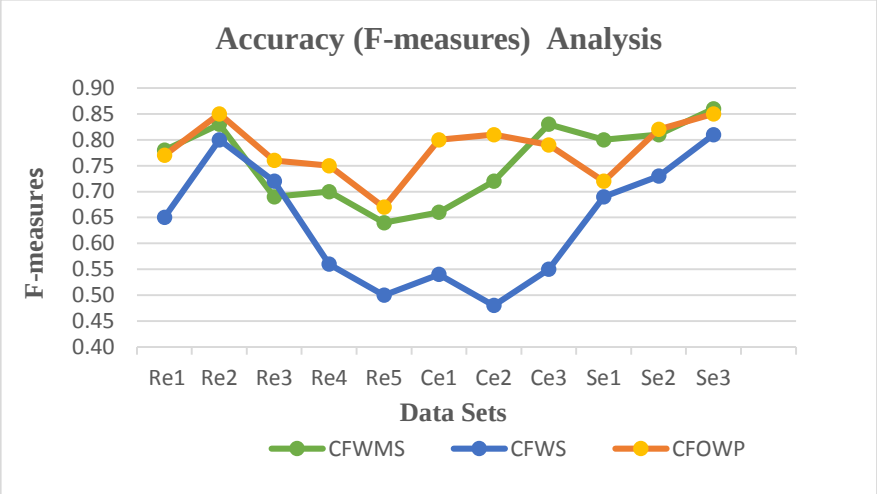

Figure 1. Accuracy analysis between the three algorithms: CFWMS, CFWS, CFOWP

- **The effect of scaling of data set on the performance**

Experiments are also conducted to understand the effect of clustering at varying levels of minimum support for generating the feature set. At three different levels (05%, 10% and 15%) of minSup values, features are generated using one data set from Reuters, one from classic and one from Se and the clustering is done. The output of the experiments are shown in Table 3. When the F-measure values are compared with that obtained for the algorithm CFWS, CFOWP with the new approach of AHC with k-nearest neighbours plus the new similarity measure Score shows better results. Figure 2 shows 3 bar graphs, one for each of the three data sets re1, Ce1 and Se1.

Table 3. F-Measure Vs Minimum Support showing Scaling Effects

Data sets with 3 different minimum supports	F - measure	
	CFWS	CFOWP
Re1 05%	0.551	0.70
10%	0.708	0.78
15%	0.641	0.56
Ce1 05%	0.571	0.69
10%	0.516	0.77
15%	0.508	0.80
Se1 05%	0.629	0.72
10%	0.754	0.78
15%	0.691	0.74

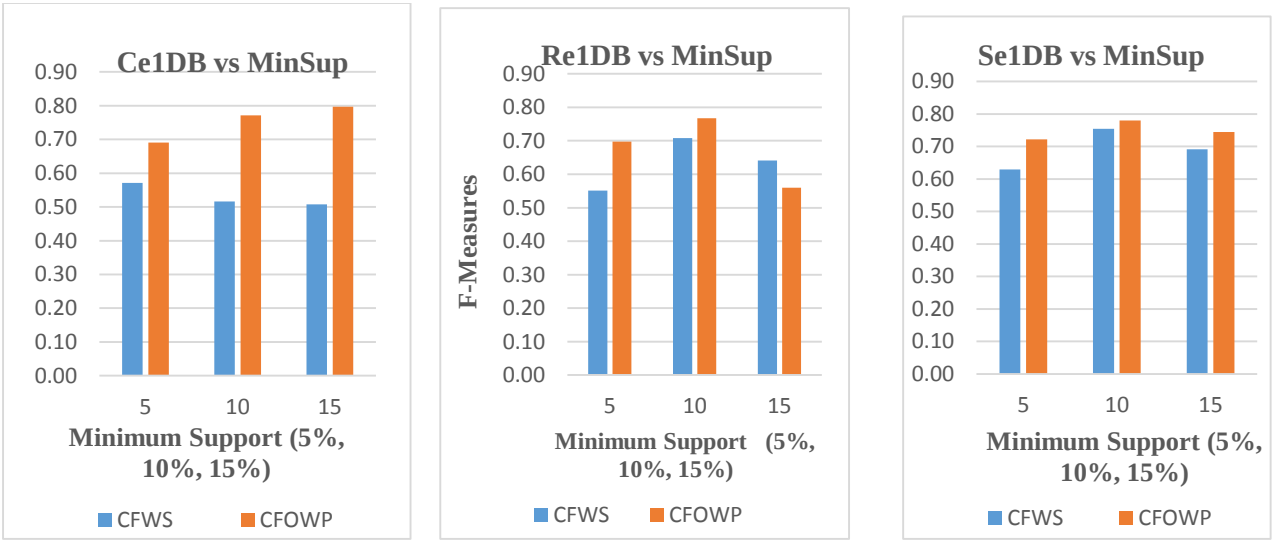

Figure 2. Analysis of scaling effect on F-measure values for the two algorithms on three data sets with different values of minimum supports (MinSup)

International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 7 (2018) pp. 4822-4833
© Research India Publications. https://dx.doi.org/10.37622/IJAER/13.7.2018.4822-4833

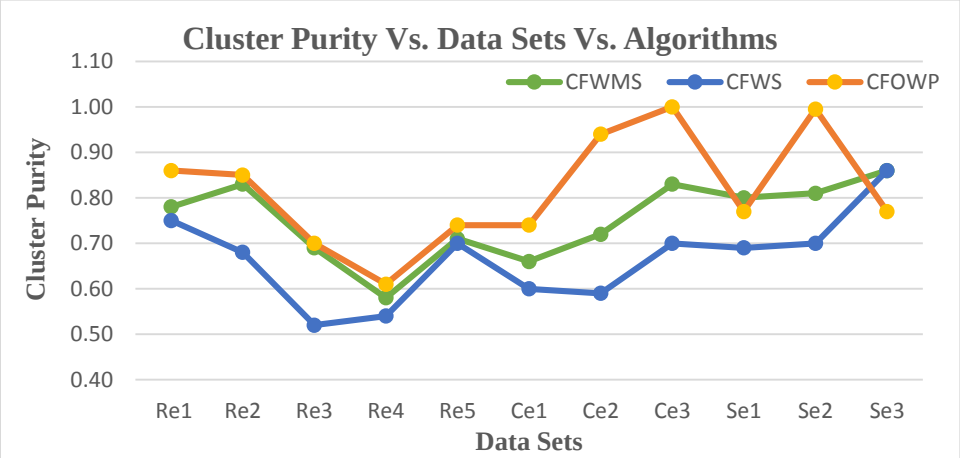

Figure 3. The Cluster Purity attained by the three algorithms

Table 4. Cluster Purity: CFWMS & CFWS vs CFOWP

Data sets	Cluster Purity		
	CFWMS	CFWS	CFOWP
Re1	0.78	0.78	0.86
Re2	0.83	0.83	0.85
Re3	0.69	0.69	0.84
Re4	0.58	0.58	0.61
Re5	0.71	0.71	0.74
Ce1	0.66	0.66	0.74
Ce2	0.72	0.72	0.94
Ce3	0.83	0.83	1.00
Se1	0.80	0.8	0.77
Se2	0.81	0.81	1.00
Se3	0.86	0.86	0.77

Cluster Purity Analysis

The purity of a cluster is the measure of the representation of the cluster corresponding to the largest class of documents assigned to that cluster. The purity output of the execution of the clustering algorithm on all the 11 data sets are tabulated in Table 4. The purity values vary from 0 to 1. Figure 3 shows the graph of Cluster Purities achieved by the three algorithms, using the data from Table 4. From the graph it is clear that the proposed algorithm CFOWP is performing better compared to the other two algorithms. This is achieved by adding the local feature weights AvgNetFrq of FOWPs with respect to each document to the global weight eNGD. For the data sets Se1 and Se3, CFWMS algorithm shows higher performance.

The Entropy Analysis

Entropy of a clustering algorithm measures the homogeneity of the clusters generated. It quantifies how much disorder is remaining in clustering. Measuring entropy, is an efficient way to evaluate the goodness of a cluster set generated. As the number of categories increases, the entropy also increases [31]. This is clear from the graph of entropy values shown in figure 4, collected during the execution of the proposed algorithm CFOWP. Re4 and Re5 are having 31 and 35 categories and their entropy values are very high. This means that the output clusters are not efficient for these data sets.

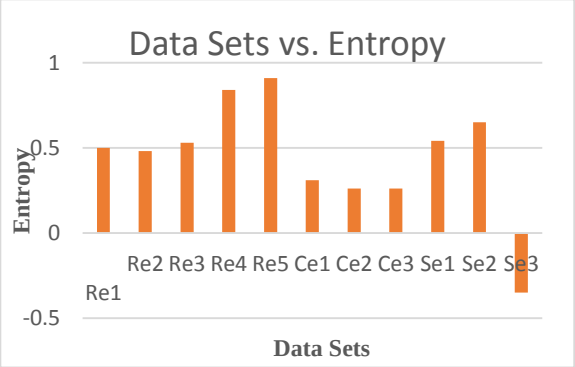

Figure 4. The Variation in Entropy Produced by CFOWP on Various Data Sets

CONCLUSION AND FUTURE WORKS

Average Link Agglomerative Hierarchical Clustering (AHC), combined with k-Nearest Neighbor approach is used in this work for clustering text documents. The greedy approach of nearest neighbor improved the efficiency of AHC by allowing a limited number of nearer clusters to be merged with the same seed cluster in the same iteration. This in turn reduced the number of accessing the similarity matrix to find the closest clusters. The Apriori algorithm is used to generate the Frequent Ordered Word Patterns (FOWP) which are the features used in this work to determine the similarity between the clusters. A new similarity measure, called Score (eNGD + AvgNetFrq) is used to construct the similarity matrix (simMat) of the clusters at each level of hierarchy. The simMat is used to select a set of clusters nearer to the seed cluster in the current level of hierarchy, to be merged and

form new clusters. The Normalised Google Distance (NGD) is a very good distance measure for clustering text data but with the limitation that it does not consider the information local to documents for searching out the patterns and that no specific order of words is kept for forming the patterns. In the context of query processing and information retrieval using clustering, this reduces the quality of clusters generated. The proposed clustering algorithm CFOWP eliminates these limitations. The use of AvgNetFrq provided additional information on the local importance of FOWPs within each document towards the clusters' similarity and the use of this combined similarity measure is found to increase the quality of clustering through experiments.

The measure of closeness is limited by a minimum threshold value named as simT. After each merge operation, the simMat is updated according to the new set of clusters.

The algorithm CFOWP is implemented and experimented with standard data sets. The results are compared with a popular algorithm. Thus the results of experiments and the merits of the proposed method are established.

Some enhancements to the present algorithms can be done in the future to increase the quality of the clusters. When the size of the data set is very high (not able to load the data into RAM fully at a time) or the input data is available only dynamically, a dynamic approach by doing incremental clustering can be applied. Also fuzzy logic can be used in order to enrich the documents using WordNet or Encyclopedia like knowledge resources to make the document clustering more meaningful and to increase the accuracy further.

REFERENCES

[1] Yanjun Li , Soon M. Chung , John D. Holt, 2007, "Text document clustering based on frequent word meaning sequences," Knowledge and Data Engineering,, Elsevier B.V, 64 (2008) 381–404, Elsevier.

[2] O. Zamir, O. Etzioni, O. Madani, R.M. Karp, 1997, "Fast and intuitive clustering of web documents," in: Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining, pp 287–290.

[3] C. Fellbaum (Ed.), 1998, "WordNet: An Electronic Lexical Database," Cambridge, MA: MIT Press.

[4] Cilibrasi, R. L., & Vitanyi, P.M.B, 2007, "The Google Similarity Distance, "Transactions on Knowledge and Data," IEEE Engineering, 19(3), 370–383.

[5] Jiawei Han and Micheline Kamber , 2012, "Data Mining Theory and Concepts," ELSEVIER, III edition.

[6] M. Steinbach, G. Karypis, V. Kumar, 2000, "A Comparison Of Document Clustering Techniques," KDD-Workshop on Text Mining.

[7] T Zhang, R.Ramakrishnan, M.Livny, 1996, "BIRCH – an Efficient Data Clustering Method for Very Large Data Bases," In Proc. 1996 ACM SIGMOD, Int. Conf. Management of Data (SIGMOD '96), pp103-114, Montreal, Quebec, Canada.

[8] G.Karypis, E-H Han and V.Kumar, 1999, "Chameleon – a Hierarchical Clustering Algorithm using Dynamic Modelling," COMPUTER, 32:68-75.

[9] Yung-Shen Lin, Jung-Yi Jiang, and Shie-Jue Lee, 2014, "Similarity Measure for Text Classification and Clustering," IEEE Transactions On Knowledge And DataEngineering, Vol. 26, No.7.

[10] B.C.M. Fung, K. Wang, M. Ester, 2003 "Hierarchical Document Clustering Using Frequent Itemsets," in: Proceedings of SIAM International Conference on Data Mining.

[11] B.C.M. Fung, K. Wang, M. Ester, 2005, "Hierarchical document clustering," in: John Wang (Ed.), The Encyclopedia of Data Warehousing and Mining, Idea Group.

[12] H. Ahonen-Myka, 2002, "Discovery of frequent word sequences in text," in: Proceedings of the ESF Exploratory Workshop on Pattern Detection and Discovery in Data Mining, pp. 16–19.

[13] R. Agrawal, R. Srikant, 1994, "Fast algorithms for mining association rules," in: Proceedings of the 20th VLDB Conference, pp. 487–499.

[14] Zheng Yu, Haixun Wang, Xuemin Lin, 2016, "MinWang, Understanding Short Texts through Semantic Enrichment and Hashing," IEEE Transactions On Knowledge And Data Engineering, Vol. 28, No.2.

[15] Michael W. Berry, Murray Browne, 2005, "Understanding Search Engines," Mathematical Modeling and Text Retrieval, SIAM, Society for Industrial and Applied Mathematics, Philadelphia, Second Edition.

[16] Bjørn Kjos-Hanssen, Alberto J. Evangelista, 2006, "Google distance between words**,"** Frontiers in Undergraduate Research, University of Connecticut. (or arXiv:0901.4180v2 [cs.CL]).

[17] Huang, Anna, 2008, "Similarity Measures for Text Document Clustering," Proceedings of the sixth New Zealand, Computer Science research Student NZCSRSC 2008.

[18] Lan Huang, David Milne, Eibe Frank and Ian H. Witten, 2012, "Learning a Concept-based Document Similarity Measure**,"** Journal of American Society for Information Science and Technology Vol. 63 Issue 8, pages 1593-1608, August 2012.

[19] Yung-Shen Lin, Jung-Yi Jiang, and Shie-Jue Lee, 2014, "A Similarity Measure for Text Classification and Clustering," IEEE.

[20] Zakaria Elberrichi, Abdelattif Rahmoun, and Mohamed Amine Bentaalah, 2008, "Using WordNet for Text Categorization," The International Arab Journal of Information Technology, Vol. 5, No. 1.

[21] Porter M, 1980, "An Algorithm for Suffix stripping. Program," 14(3):130-137.

[22] R. Srikant and R. Agrawal, 1996, "Mining Sequential Patterns: Generalisations and Performance Improvements," In. Proc. 5th International Conference, Extending Database Technology (EDBT'96), pp. 3-17, Avingnon, France, March.

[23] R. Srikant, Q. Vu, and R. Agrawal. 1997, "Mining Association Rules With Item Constraints," .KDD'97, 67-73, Newport Beach, California.

[24] G. Salton, A. Wong, and C. S. Yang, 1975, "A vector space model for automatic indexing," *Commun. ACM*, vol. 18, no. 11, pp. 613–620.

[25] Cilibrasi, Rudi. L and Paul MB Vitanyi, 2006, "Automatic Meaning Discovery Using Google," Daggstuhl Seminar Proceedings, CWI, University of Amsterdam.

[26] Available:http://www.daviddlewis.com/resources/ Test collections/Reuters21578/.

[27] [Classic Data Set] [Online] Available: ftp://ftp.cs.cornell.edu/pub/smart/.

[28] 2004, "High Accuracy Retrieval from Documents (HARD) Track of Text Retrieval Conference".

[29] J. Allan, 2003, "HARD track overview in TREC 2003 high accuracy retrieval from documents," in: Proceedings of the 12th Text Retrieval Conference, pp.24-37.

[30] David.D.Lewis,[online] Available at **Error! Hyperlink reference not valid.**resources/testcollections/ reuters21578/.

[31] Gang Kou, Yi Peng, Guoxun Wang,, 2014, "Evaluation of Clustering Algorithms for Financial Risk Analysis using MCDM methods," Information Sciences, vol 275,Pages1-12, Science Direct, Elsevier.