Performance Evaluation of an Efficient Frequent Item sets-Based Text Clustering Approach

S.Murali Krishna ,S.Durga Bhavani

GJCST Classification (FOR)
H.2.8, I.7.m

***Abstract*-The vast amount of textual information available in electronic form is growing at a staggering rate in recent times. The task of mining useful or interesting frequent itemsets (words/terms) from very large text databases that are formed as a result of the increasing number of textual data still seems to be a quite challenging task. A great deal of attention in research community has been received by the use of such frequent itemsets for text clustering, because the dimensionality of the documents is drastically reduced by the mined frequent itemsets. Based on frequent itemsets, an efficient approach for text clustering has been devised. For mining the frequent itemsets, a renowned method, called Apriori algorithm has been used. Then, the documents are initially partitioned without overlapping by making use of mined frequent itemsets. Furthermore, by grouping the documents within the partition using derived keywords, the resultant clusters are obtained effectively. In this paper, we have presented an extensive analysis of frequent itemset-based text clustering approach for different real life datasets and the performance of the frequent itemset-based text clustering approach is evaluated with the help of evaluation measures such as, precision, recall and F-measure. The experimental results shows that the efficiency of the frequent itemset-based text clustering approach has been improved significantly for different real life datasets.**

Keywords-Text mining, Text clustering, Text documents, Frequent itemsets, Apriori, Reuter-21578, Webkb dataset, 20-newsgroups.

I. Introduction

Databases in every aspect of human actions progresses rapidly. This has led to an enormous demand for efficient tools that turn data into valuable knowledge. Researchers from numerous technological areas, namely, pattern recognition, machine learning, data visualization, statistical data analysis, neural networks, information retrieval, econometrics, information extraction etc have been searching for eminent approaches to fulfill this requirement. An effective research area known as data mining (DM) or Knowledge Discovery in Databases (KDD) has resulted as a result of their entire efforts [6]. Data represented in quantitative, multimedia or textual forms are commonly utilized for data mining [21]. Presently, in documents, news, email and manuals, large amount of information exists in the form of text. Because of the growth of digital libraries, web, technical documentation and medical data, access to a large quantity of textual documents turns out to be effectual. These textual data consists of resources which can be used in a better way. Text mining (TM) or in other words knowledge discovery from textual databases is a prominent and tough challenge in majority of the existing documents which employs natural language due to its value and ambiguity [1]. The need of acquiring knowledge fromlarge number of available text documents, particularly on the Web has made text mining as one of the major research fields [8].In some perspective, the information mining tasks such as, text mining and data mining are identical. Text can be mined by adapting the data mining techniques in a better way [8]. In both the knowledge-discovery variants, data is organized by tagging the document elements or by representing the numerical data in organized data structures. By identical application of statistical techniques, the complexity of the problem is lessened, correlations, linkages, clusters, and relationships are recognized, and predictive rules are made, which are also identified in few circles as knowledge [11, 12, and 13]. Text mining can be described as a knowledge-intensive process in which a user corresponds using a set of analysis tools in due course with a collection of documents. Similar to data mining by recognition and searching of interesting patterns, text mining anticipates valuable information from data sources. “Text mining” refers to the automated discovery of valuable or interesting information from unstructured text by the application of data mining techniques [4], [3], [2] and [10]. The requirement to attain knowledge from massive amount of text documents has led to a progressively more significant research field known as text mining [34]. Pre-processing of text documents and subsequent saving of data in a data structure that is more suitable for further processing than a plain text file are essential to mine large document collections [18]. Typically, tokenization, Part of Speech (PoS) Tagging [19], word stemming and the application of a stop words removal technique are included in text preprocessing. The process of splitting the text into words or terms is known as tokenization. According to the grammatical context of a word in a sentence, it can be tagged using Part of Speech Tagging and thereby words can be divided into nouns, verbs and more [20]. Extract of key information from the original text represented by filling predefined structured representation, or templates by mapping natural language texts (namely newswire reports, journal and newspaper articles, World Wide Web pages, electronic mail, any textual database and more.) is defined as Information Extraction [7]. Lately, significant interest has been achieved by extracting relationships from entities in text documents. The interesting associations and/or correlation relationships among large set of data items are discovered by association rule mining.In a wide range of application areas, mining association rules in transaction databases has been demonstrated to be valuable [15, 16]. Due to the difference in characteristics between transaction databases and text databases, application of association rules mining seems to be

About-Associate Professor Department of Computer Science and Engineering
smuralikrishnaphd@gmail.com

more promising in text databases [14]. In the case of text mining, extracted rules are able to return semantic relations among the terms because they are deduced as co-occurrences of terms in texts [17]. Text analysis, information retrieval, clustering, information extraction, categorization, visualization, machine learning, data mining and database technology are functions included in the multidisciplinary field of text mining [22]. Cluster analysis is the method of dividing data objects (e.g.: document and records) into significant clusters or groups such that contradictory to objects in other clusters, analogous characteristics are possessed by objects within a cluster [4], [5]. Navigation, summarization, and arrangement of text documents by users are assisted by text clustering which sorts several documents in an efficient way [31, 32, 9]. Document clustering, which arranges a huge number of documents into meaningful clusters, can be employed to browse a set of documents or to arrange the results given by a search engine in answer to a user's query. The accuracy and recall in information retrieval systems can be considerably enhanced and the nearest neighbors of a document can be determined in a proficient way by document clustering [33]. In our prior research [45], we have presented an effective frequent itemset-based document clustering approach. First, with the aid of stop words removal technique and stemming algorithm, the text documents in the text data are preprocessed. Then, from each document, the top-p frequent words are extracted and the binary mapped database is formed using the extracted words. The different length frequent itemsets are discovered by applying the Apriori algorithm. Based on the support level of the mined frequent itemsets, the itemsets are sorted in descending order for each length. Subsequently, using the sorted frequent itemsets, we split the documents into partition. Understandable description of the partitions can be obtained from these frequent itemsets. Furthermore, the derived keywords (words obtained by taking the absolute complement of familiar keywords with respect to the top-p frequent words) are used to form the resultant cluster within the partition. In this paper, we have used the different real life datasets such as, Reuter-21578, 20-newsgroups and Webkb datasets for analysing the frequent itemset-based document clustering approach. In addition with, for evaluating the performance, we make use of evaluation metrics namely, precision, recall and F-measure.The paper is organized as follows. The concise review of related researches of text clustering is presented in Section 2. The text clustering approach based on frequent itemset is described in section 3. The extensive analysis of the text clustering approach using different datasets is given in section 4. The conclusion is summed up in section 5.

II. Literature Review of Recent Related Researches

The literature presents a lot of text clustering approach, wherein the frequent itemset based text clustering has received considerable attention among the research community. Some recent researches related to frequent itemset-based text clustering is briefly reviewed in this section.Zhou Chong *et al.* [23] have presented a method for text clustering known as Frequent Itemset-based Clustering with Window (FICW), in which the semantic information has been employed with a window constraint. FICW has revealed better performance in terms of clustering accuracy and efficiency in the experimental results obtained from three (hypertext) text sets. Xiangwei Liu and Pilian [24] have introduced a text-clustering algorithm known as Frequent Term Set-based Clustering (FTSC) which clusters texts by employing frequent term sets. Initially, significant information from the documents are extracted by it and kept in databases. Later, the frequent item sets are mined by it employing the Apriori algorithm. Finally, the documents are clustered as per the frequent words in subsets of the frequent term sets. The dimension of the text data can be lessened by the algorithm for extremely large databases, so the accuracy and speed of the clustering algorithm can be enhanced. Experimental results have showed that the clustering performance efficiency of FTSC and FTSHC algorithms are superior to that of the K-Means algorithmLe Wang *et al.* [25] have presented a top-k frequent term sets and k-means based simple hybrid algorithm (SHDC) to overcome the main challenges of current web document clustering. K initial means regarded as initial clusters were provided by employing top-k frequent term sets, which were later refined by k-means. K-means returns the final optimal clustering whereas *k* frequent term sets provides the clear description of clustering. Efficiency and effectiveness of SHDC was proved to be superior to that of the other two representative clustering algorithms (the farthest first k-means and random initial k-means) by the experimental results conducted on two public datasets. Zhitong Su *et al.* [26] have introduced a maximal frequent itemsets based web-text clustering method for personalized e-learning. The Web documents were initially represented by vector space model. Maximal frequent word sets were determined subsequently. In the end, documents were clustered by employing maximal itemsets on the basis of a new similarity measure of itemsets. The method was proved to be efficient by the obtained Experimental results.Yongheng Wang *et al.* [27] have introduced a parallel clustering algorithm based on frequent term for clustering short documents in very large text database. To enhance the clustering accuracy, a semantic classification method has also been employed. The algorithm has been proved to be more precise and efficient than other clustering algorithms when clustering large scale short documents based on experimental analysis. In addition, the scalability of the algorithm is good and also huge data could be processed by employing it. W.L. Liu and X. S. Zheng have proposed the frequent term sets based documents clustering algorithm [29]. Initially, by means of the Vector Space Model (VSM), the documents were denoted and all the terms were sorted in accordance with their relative frequency. Then, frequent-pattern growth (FP growth) has been used to mine the frequent term sets. Lastly, on the basis frequent term sets the documents were clustered. The approach has been efficient in very large databases and also a clear explanation in terms of frequent terms about the determined clusters has been provided by the algorithm. With the aid of experimental results, the efficiency and suitability of the proposed algorithm has been demonstrated.Henry Anaya-Sanchez *et al.* [30] have proposed a text clustering algorithm which depending on the most probable term pairs of the collection and its estimate of related topic homogeneity, discovers and unfolds the topics, included in the text collection. From term pairs that have support sets with sufficient homogeneity for denoting collection topics, topics and their descriptions were produced. The efficacy and usefulness of the approach has been verified by the experimental results obtained over three benchmark text collections. Florian Beil *et al.* [28] have proposed an approach for text clustering which employs frequent item (term) sets. Utilizing algorithms for association rule mining such frequent sets were determined. They have gauged the mutual overlap of frequent sets with regard to the sets of supporting documents to cluster on the basis of frequent term sets. FTC and HFTC are the two algorithms, which they have provided for frequent term-based text clustering. Flat clustering is produced by FTC whereas HFTC is for obtaining hierarchical clustering. Clustering of the presented algorithm has been proved to be of comparable quality and appreciably more efficiency than modern text clustering algorithms based on an experimental assessment on classical text as well as web documents.

III. Proficient Approach For Text Clustering Based On Frequent Itemsets

The exploration for hidden knowledge in text collections has been provoked by the reputation of the Web and the huge quantity of documents existing in electronic form. Therefore, research concentration is increasing in the general topic of text mining. One of the popular techniques among the various data mining techniques that have been used by researchers for finding the meaningful information from the text documents is clustering. Grouping a collection of documents (unstructured texts) into different category groups so that the same subject is described by documents in the same category group is known as text clustering. Possible ways to improve the performance of text document clustering based on the popular clustering algorithms (partitional and hierarchical clustering) and frequent term based clustering has been investigated by many researches [23-26, 28]. An effectual approach for clustering a text corpus with the aid of frequent itemsets [45] is discussed in this section. The text clustering approach consists of the following major steps:

1) Text preprocessing
2) Mining of frequent itemsets
3) Partitioning the text documents based on frequent Itemsets
4) Clustering of text documents within the partition

1) Text Pre-processing

Let D be a set of text documents represented as $D=\{d_1\ d_2\ d_3....d_n\};1\le \mathrm{i}\le n$, where, n is the number documents in the text dataset D. The words or terms are extracted (tokenization) from the text document set D using the text preprocessing techniques and it is converted from unstructured format into some common representation. For preprocessing the input data set D (text documents), two techniques namely, removing stop words and stemming algorithm are used.

(a) Stop word Removal: Stop (linking) words like “have”, “then”, "it", "can”, “need", "but", "they", "from", "was", "the", "to", "also" are removed from the document [36].

(b) Stemming algorithm: Prefixes and suffixes of each word [35] are removed.

2) Mining of Frequent Itemsets

Mining of frequent itemsets from the preprocessed text documents D is described in this sub-section. After the preprocessing step, the frequency of the extracted words or terms is computed for every document d_i, and the top- p frequent words from each document d_i are taken out.

$$K_w = \{d_i \mid \mathrm{p}(\mathrm{d_i}) \quad ; \quad \forall\, d_i \subseteq D\}$$

where, $p(d_i) = T_{w_j} \quad ; \quad 1 \le j \le p$

By using the unique words of the set of top- p frequent words, the binary database B_T is formed. Let B_T be a binary database consisting of n number of transactions (documents) T and q number of attributes (unique words) $U=[u_1,u_2,....,u_q]$. Whether the unique words are present or not in the documents, is represented in the binary database B_T, which consists of binary data.

$$B_T = \begin{cases} 0 & if \quad u_j \notin d_i \\ 1 & if \quad u_j \in d_i \end{cases}$$

$$; \quad 1 \le j \le q\,, \quad 1 \le i \le n$$

Then, the frequent itemsets (words/terms) F_s is mined by inputting the binary database B_T to the Apriori algorithm.

a) Apriori algorithm

Apriori algorithm, first introduced in [37] is a conventional algorithm for mining association rules. Association rules mining involves two steps such as, (1) Identifying frequent itemsets (2) Generating association rules from the frequent itemsets. Two steps are involved in frequent itemsets mining. Generating the candidate itemsets is the first step. In the second step, assisted by these candidate itemsets, the frequent itemsets are mined. Frequent itemsets consists of itemsets whose support is greater than the user specified minimum support.In our proposed approach, we use only the frequent itemsets for further processing. The pseudo code corresponding to first step (generation of frequent itemsets) of the Apriori algorithm [38] is given below.

Pseudo code:

C_k : Candidate itemset of size k

I_k : Frequent itemset of size k.

```
$I_1 = \{l \arg e \ \ 1-itemsets\};$
$for\ (k=2;\ I_{k\text{-}1} \neq 0;\ k++)\ do\ begin$
        $C_k = apriori-gen(I_{k-1});$   // New candidates
        for all transactions $T \in D$ do begin
            $C_T = subset(C_k, T);$
            // Candidates contained in $T$
            for all candidates $c \in C_T$ do
                $c.count++;$
            end
        end
        $I_k = \{c \in C_k \mid c.count \geq \min \sup\}$
    end
    $Answer = \bigcup_k I_k;$
```

C. Partitioning the Text Documents Based on Frequent Itemsets

Partitioning of text documents (D) based on mined frequent itemsets (F) is described in this section. ***Definition1:****Frequent itemset* is a set of words that occur together in some minimum fraction of documents in a cluster. A set of frequent itemsets of varying length (l), from 1 to k , are generated by the Apriori algorithm. First, the set of frequent itemsets of each length (l) are sorted in descending order of their support level.

$$F_s = \{ f_1\ \ f_2\ \ f_3 \ldots f_k \}\ \ ;\ \ 1 \le l \le k$$

$$f_l = \{ \ f_{l(i)}\ \ ;\ \ 1 \le i \le t\}$$

where, $\sup(f_{l(1)}) \ge \sup(f_{l(2)}) \ge \ldots \ge \sup(f_{l(t)})$ and t denotes the number of frequent itemsets in the set f_l .At first, the first element ($f_{(k/2)}(1)$) is selected from the sorted list $f_{(k/2)}$, that contains a set of frequent itemsets. Then, an initial partition c_1 is constructed by grouping all the documents that contains the itemset $f_{(k/2)}(1)$. After that, a new partition c_2 is formed for the second element $f_{(k/2)}(2)$, the support of which is less than $f_{(k/2)}(1)$. This new partition c_2 is formed by grouping all the documents that have the frequent itemset $f_{(k/2)}(2)$ and also, the documents in the initial partition c_1 are taken away from this new partition. This process is done repeatedly until every text documents in the input dataset D are moved into partition $C_{(i)}$. In addition, if the above procedure is not terminated with the sorted list $f_{(k/2)}$, then the above discussed steps (inserting the documents into the partition) are performed by taking the subsequent sorted lists ($f_{((k/2)-1)}$, $f_{((k/2)-2)}$ etc..). This provides a set of partition c and each partition $C_{(i)}$ contains a collection documents $D^{(x)}_{c(i)}$.

$$c = \{ \ c_{(i)} \mid c_{(i)} \in f_{l(i)} \}\ \ \ ;\ \ \ \ 1 \le i \le m\ ,\ \ 1 \le l \le k$$

$$C_{(i)} = Doc[\ f_{l(i)}\ ]$$

$$C_{(i)} = \{ \ D^{(x)}_{c(i)}\ ;\ D^{(x)}_{c(i)} \in\ D\ \ ,\ 1 \le x \le r\ \ \}$$

where, m denotes the number of partitions and r indicates the number of documents in each partition.

Mined frequent itemset used for constructing initial partition (or cluster), significantly reduces the dimensionality of the text document set, and the reduced dimensionality considerably enhances the efficiency and scalability of clustering. Due to the use of frequent itemsets, overlapping of documents exists in the clustering results produced by the approaches presented in [41, 28] and the final results are obtained by removing these overlapping documents. In the proposed research, non-overlapping partitions are directly generated from the frequent itemsets. This makes the initial partitions disjoint, because the document is kept only within the best initial partition by this text clustering approach.

3) Clustering of Text Documents within the Partition

This sub-section, describes how clustering is done on the set of partitions obtained from the previous step. This step is necessary to form a sub cluster (describing sub-topic) of the partition (describing same topic) and the outlier documents can be significantly detected by the resulting cluster.Furthermore, a pre-specified number of clusters are not required by this text clustering approach. The devised procedure for clustering the text documents available in the set of partition c is discussed below.

In this phase, for each document $D^{(x)}_{c(i)}$, the familiar words $\mathrm{f}_{c(i)}$ (frequent itemset used for constructing the partition) of each partition $C_{(i)}$ are first identified. Then, by taking the absolute complement of familiar words $\mathrm{f}_{c(i)}$ with respect to the top- p frequent words of the document, the derived keywords $K_d[D^{(x)}_{c(i)}]$ of document $D^{(x)}_{c(i)}$ are obtained.

$$K_d[D^{(x)}_{c(i)}] = \{ T_{w_j} \setminus f_{c(i)} \}\ ;\ T_{w_j} \in D^{(x)}_{c(i)}\ \ ,$$

$$1 \le i \le m\ \ ,\ 1 \le j \le p\ \ , 1 \le x \le r$$

$$T_{w_j} \setminus f_{c(i)} = \{ x \in T_{w_j} \mid x \notin f_{c(i)} \}$$

For each partition $C_{(i)}$, the set of unique derived keywords and their support are computed within the partition. The representative words of the partition $C_{(i)}$ are formed by the set of keywords which satisfy the cluster support ($cl\_\sup$). Definition2: The percentage of the documents in $C_{(i)}$ that contains a keyword is the *cluster support* of that keyword in $C_{(i)}$.

$$R_w[\ c(i)\ ] = \{ \ x : p\ (x) \}$$

$$\text{where, } p(x) = [K_d[D_{c(i)}^{(x)}]] \geq cl\_\sup$$

Subsequently, with respect to the representative words $R_w[c(i)]$, the similarity of the documents $D_{c(i)}^{(x)}$ is computed. An important role is played by the definition of similarity measure in obtaining effective and meaningful clusters. The similarity between two text documents S_m is calculated as follows,

$$S\left(K_d[D_{c(i)}^{(x)}], R_w[c(i)]\right) = \left| K_d[D_{c(i)}^{(x)}] \cap R_w[c(i)] \right|$$

$$S_m\left(K_d[D_{c(i)}^{(x)}], R_w[c(i)]\right) = \frac{S\left(K_d[D_{c(i)}^{(x)}], R_w[c(i)]\right)}{|R_w[c(i)]|}$$

2. Experimentation And Performance Evaluation

This section details the experimentation and performance evaluation of the frequent itemset-based text clustering approach. We have implemented the frequent itemset-based text clustering approach using Java (JDK 1.6). The text clustering approach is evaluated based on the evaluation metrics given in sub-section 4.1 and in sub-section 4.2, the performance of the text clustering approach is analyzed with the different real life datasets.

1) Evaluation measures

Precision, Recall and F-measure described in [39, 40] are used for evaluating the performance of the frequent itemset-based text clustering approach. The definition of the evaluation metrics is given below,

$$\text{Precision}(i, j) = M_{ij} / M_j$$

$$\text{Recall}(i, j) = M_{ij} / M_i$$

$$\mathbf{F-measure}\,(i, j) = \frac{2 * \text{Recall}\,(i, j) * \text{Precision}\,(i, j)}{\text{Precision}\,(i, j) + \text{Recall}\,(i, j)}$$

where M_{ij} is the number of members of topic i in cluster j, M_j is the number of members of cluster j and M_i is the number of members of topic i.

2) Performance Evaluation

The experimentation is carried out on different datasets based on the steps described in the section 3. At first, the top- p frequent words are extracted from each document and binary database is constructed using these frequent words. Using Apriori algorithm, frequent itemsets are mined from the binary database and sorted it based on their support level. Then, we construct initial partition using these frequent itemsets. Subsequently, we compute representative words of each partition with the help of top- p frequent words and familiar words. For each document, we calculate the similarity measure with respect to the representative words. Finally, if the similarity value of the document within the partition is below 0.4, it forms as a separate cluster. The detailed evaluation of the frequent itemset-based text clustering approach for different real life datasets is described below.

Dataset 1: We have taken 100 documents manually from various topics namely, Content based video retrieval, Semantic web, Incremental clustering, Gene prediction, Human Resource, Sequential pattern mining, Adaptive e-learning, Multimodal Biometrics, Public key cryptography, Automatic text summarization and Grid Computing. These documents are fed as an input to the text clustering approach that provides 25 clusters. The performance of the resulted cluster is evaluated with three measures (Precision, Recall and F-measure) and the obtained result is given in table 1. The plotted graph for the dataset 1 is given in figure 1. By analyzing the graph shown in figure 1, some of the resulted cluster has achieved maximum precision.

Table 1. Precision, Recall and F-measure of dataset 1

Partition	Cluster	Precision	Recall	F-measure
P_1	C_1	1	0.666667	0.8
	C_2	0.454545	0.714286	0.555555
P_2	C_3	1	0.4	0.571429
	C_4	0.222222	0.2	0.210526
P_3	C_5	1	0.5	0.666667
	C_6	0.375	0.428571	0.4
P_4	C_7	0.666667	0.2	0.307692
	C_8	0.571429	0.4	0.470588
P_5	C_9	1	0.2	0.333333
	C_{10}	0.25	0.142857	0.181818
P_6	C_{11}	1	0.2	0.333333
	C_{12}	0.333333	0.125	0.181818
P_7	C_{13}	1	0.555556	0.714286
	C_{14}	0.4	0.2	0.266667
P_8	C_{15}	0.75	0.333333	0.461538
	C_{16}	0.333333	0.142857	0.2
P_9	C_{17}	1	0.25	0.4
	C_{18}	0.666667	0.2	0.307692
P_{10}	C_{19}	1	0.222222	0.363636
	C_{20}	0.5	0.125	0.2
P_{11}	C_{21}	1	0.2	0.333333
P_{12}	C_{22}	1	0.222222	0.363636
	C_{23}	1	0.111111	0.2
P_{13}	C_{24}	1	0.2	0.333333
P_{14}	C_{25}	1	0.25	0.4

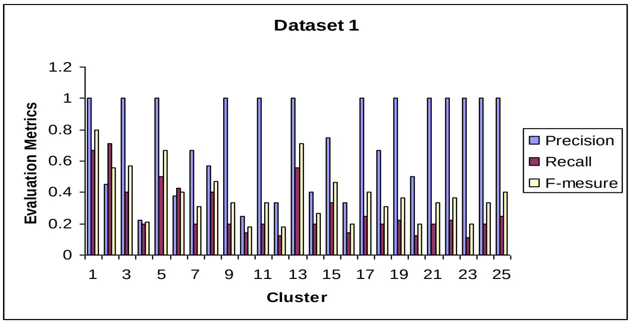

Fig.1. Performance of the frequent itemset-based text clustering approach on dataset 1

(2) Reuter 21578 dataset: The documents in the Reuters-21578 set [42] resembled on the Reuters newswire in 1987. The documents were accumulated and indexed with grouping, by personnel from Reuters Ltd. Additionally, formatting and data file production was achieved in 1991 and 1992 by David D. Lewis and Peter Shoemaker at the Center for Information and Language Studies, University of Chicago. For experimentation, we have taken 125 documents from 10 different topics (cpi, bop, cocoa, coffee, crude, earn, trade, acq, money-fx, oilseed) and these documents is given as input documents to the text clustering approach. It provides 24 clusters and for each cluster, the precision, Recall and F-measure is computed. The results obtained are given in table 2 and their corresponding graph is shown in figure 2. It ensures that some of the clusters obtained its maximum precision and recall measures.

Table 2. Precision, Recall and F-measure of Reuter 21578

Partition	Cluster	Precision	Recall	F-measure
P_1	C_1	0.8	0.5333	0.639976
	C_2	0.2857	0.5454	0.374975
P_2	C_3	1	0.5	0.666667
	C_4	0.9285	1	0.962925
P_3	C_5	1	0.4166	0.588169
	C_6	0.4444	0.2666	0.333269
P_4	C_7	1	0.2666	0.42097
	C_8	1	0.3333	0.499962
P_5	C_9	0.6666	0.1333	0.222172
P_6	C_{10}	1	0.2857	0.444427
	C_{11}	0.75	0.25	0.375
P_7	C_{12}	1	0.2857	0.444427
	C_{13}	0.5	0.0909	0.153833
P_8	C_{14}	1	0.2727	0.428538
	C_{15}	0.3333	0.0909	0.142843
P_9	C_{16}	1	0.0666	0.124883
P_{10}	C_{17}	0.75	0.2	0.315789
P_{11}	C_{18}	0.5	0.0833	0.142808
P_{12}	C_{19}	1	0.2727	0.428538
	C_{20}	0.6	0.2727	0.374974
P_{13}	C_{21}	1	0.5	0.666667
	C_{22}	0.5	0.25	0.333333
P_{14}	C_{23}	1	0.0714	0.133284
P_{15}	C_{24}	0.75	0.25	0.375

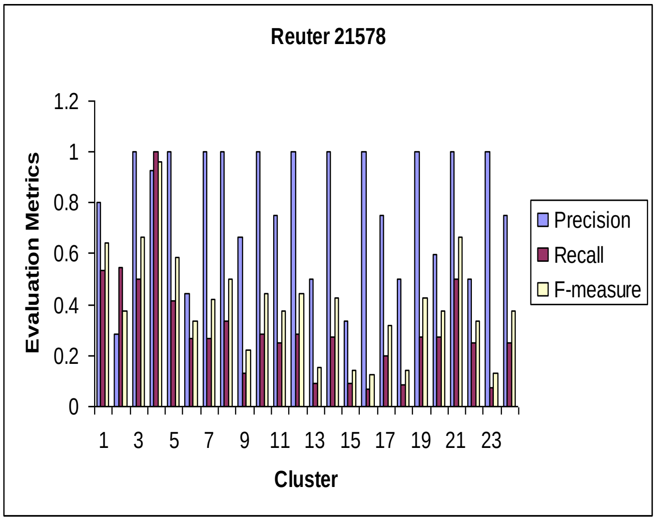

Fig.2. Performance of the frequent itemset-based text clustering approach on Reuter-21578

(3) 20 newsgroups dataset: This data set (20NG) [43] contains 20000 messages obtained from 20 newsgroups and 1000 messages are collected from each newsgroup. The various newsgroups prescribed in the dataset are alt.atheism, comp.graphics,comp.os.mswindows.misc,comp.sys.ibm.pc.hardware,comp.sys.mac.hardware,comp.windows.x, misc.forsale, rec.autos,rec.motorcycles, rec.sport.baseball, rec.sport.hockey, sci.crypt,sci.electronics,sci.med,sci.space,soc.religion.christian ,talk.politics.guns,talk.politics.mideast,talk.politics.misc,talk.religion.misc. For evaluation, the text clustering approach is applied on 201 documents taken from various newsgroups of 20NG dataset. Consequently, we have obtained 29 clusters and the resulted cluster is used to find the precision, recall and f-measure. The measured parameter is given in table 3 and the graph shown in figure 3 is plotted based on the measured parameters.

Table 3. Precision, Recall and F-measure of 20-newsgroups

Partition	Cluster	Precision	Recall	F-measure
P_1	C_1	1	0.565217	0.722222
P_2	C_2	1	0.578947	0.733333
P_3	C_3	0.875	0.233333	0.368421
P_4	C_4	0.5	0.333333	0.4
P_5	C_5	0.428571	0.130435	0.2
P_6	C_6	1	0.263158	0.416667
P_7	C_7	0.5	0.130435	0.206897
P_8	C_8	0.857143	0.24	0.375
P_9	C_9	1	0.466667	0.636364
P_{10}	C_{10}	0.857143	0.2	0.324324
P_{11}	C_{11}	0.666667	0.235294	0.347826
P_{12}	C_{12}	1	0.083333	0.153846
	C_{13}	1	0.166667	0.285714
P_{13}	C_{14}	0.5	0.181818	0.266667
P_{14}	C_{15}	1	0.352941	0.521739
P_{15}	C_{16}	1	0.24	0.387097
P_{16}	C_{17}	0.5	0.066667	0.117647
P_{17}	C_{18}	1	0.384615	0.555556
P_{18}	C_{19}	1	0.416667	0.588235
P_{19}	C_{20}	0.5	0.166667	0.25
P_{20}	C_{21}	1	0.166667	0.285714
	C_{22}	0.368421	0.233333	0.285714
P_{21}	C_{23}	0.333333	0.153846	0.210526
P_{22}	C_{24}	0.333333	0.307692	0.32
P_{23}	C_{25}	0.2	0.066667	0.1
P_{24}	C_{26}	0.25	0.25	0.25
P_{25}	C_{27}	0.4	0.5	0.44444
P_{26}	C_{28}	0.75	0.24	0.363636
P_{27}	C_{29}	0.25	0.1	0.142857

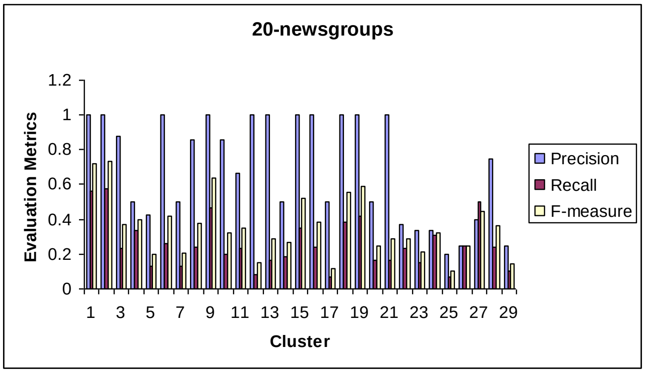

Fig.3. Performance of the frequent itemset-based text clustering approach for 20-newsgroups

(4) Webkb dataset: This data set [44] consists of WWW-pages collected from computer science departments of different universities namely, Cornell, Texas, Misc, Washington, Wisconsin in January 1997 by the World Wide Knowledge Base (Web->Kb) project of the CMU text learning group. The 8,282 pages were manually categorized into the following categories: student (1641), faculty (1124), staff (137), department (182), course (930), project (504) and other (3764). In order to evaluate the text clustering approach on Webkb dataset, we have taken 395 documents from various topics. These documents are used as input text documents and lastly, it results 27 clusters. We compute the precision, Recall and F-measure for each cluster and the attained parameters are shown in table 4. The graph for the results is given in figure 4. The obtained results show the efficiency of the text clustering approach.

Table 4. Precision, Recall and F-measure of WebKb dataset

Partition	Cluster	Precision	Recall	F-measure
P_1	C_1	0.84415584	0.77844311	0.80996885
P_2	C_2	0.67088608	0.37857143	0.48401826
P_3	C_3	0.60869565	0.1	0.17177914
	C_4	0.31818182	0.53846154	0.4
P_4	C_5	1	0.07857143	0.14569536
	C_6	0.6	0.04285714	0.08
P_5	C_7	0.82352941	0.1	0.17834395
P_6	C_8	1	0.18181818	0.30769231
	C_9	0.625	0.03571429	0.06756757
P_7	C_{10}	1	0.02142857	0.04195804
	C_{11}	0.5	0.06060606	0.10810811
P_8	C_{12}	0.75	0.01796407	0.03508772
	C_{13}	0.75	0.01796407	0.03508772
P_9	C_{14}	0.33333333	0.07692308	0.125
P_{10}	C_{15}	1	0.02142857	0.04195804
P_{11}	C_{16}	1	0.01197605	0.02366864
	C_{17}	0.5	0.15384615	0.23529412
P_{12}	C_{18}	1	0.01197605	0.02366864
	C_{19}	0.75	0.01796407	0.03508772
P_{13}	C_{20}	1	0.02142857	0.04195804
	C_{21}	1	0.07692308	0.14285714
P_{14}	C_{22}	0.5	0.00598802	0.01183432
P_{15}	C_{23}	0.5	0.00714286	0.01408451
P_{16}	C_{24}	0.33333333	0.03448276	0.0625
P_{17}	C_{25}	0.5	0.00598802	0.01183432
P_{18}	C_{26}	1	0.01197605	0.02366864
P_{19}	C_{27}	0.47058824	0.04790419	0.08695652

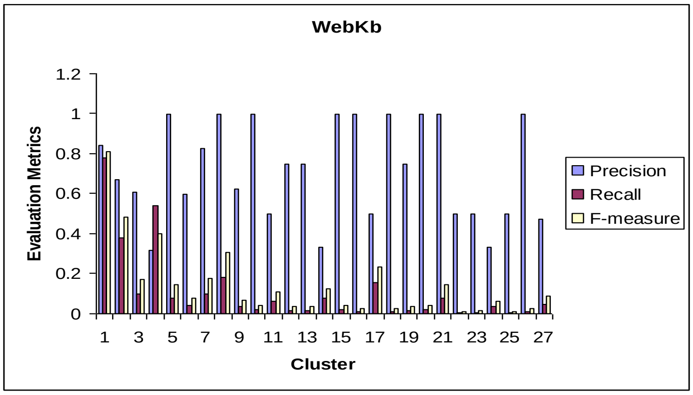

Fig.4. Performance of the frequent itemset-based text clustering approach on WebKb dataset

IV. Conclusion

Text clustering is a more specific method for unsupervised text grouping, automatic topic extraction and fast information retrieval or filtering. There has been a plenty of approaches available in the literature for clustering the text documents. In this paper, we have conducted an extensive analysis of frequent itemset-based text clustering approach with different text datasets. For different text datasets, the performance of frequent itemset-based text clustering approach has been evaluated with precision, recall and F-measure. The experimental results of the frequent itemset-based text clustering approach are given for Reuter-21578, 20-newsgroups and Webkb datasets. The performance study of the text clustering approach showed that it effectively groups the documents into cluster and mostly, it provides better precision for all datasets taken for experimentation.

V. References

1) Hany Mahgoub, Dietmar Rosner, Nabil Ismail and Fawzy Torkey, "A Text Mining Technique Using Association Rules Extraction", International Journal of Computational Intelligence, Vol. 4; No. 1, 2008.
2) Shenzhi Li, Tianhao Wu, William M. Pottenger, "Distributed Higher Order Association Rule Mining Using Information Extracted from Textual Data", ACM SIGKDD Explorations Newsletter, Natural language processing and text mining Vol. 7, No. 1 , pp. 26 - 35 , 2005.
3) R. Baeza-Yates, B. Ribeiro-Neto. “Modern Information Retrieval”, ACM Press, New York, 1999.
4) J. Han, M. Kamber, “Data Mining: Concepts and Techniques”, Morgan Kaufmann, San Francisco, 2000.
5) Jochen Dijrre, Peter Gerstl, Roland Seiffert, "Text Mining: Finding Nuggets in Mountains of Textual Data", Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining , San Diego, California, United States , pp: 398 - 401, 1999.
6) Haralampos Karanikas, Christos Tjortjis and Babis Theodoulidis, "An Approach to Text Mining using Information Extraction", Proc. Knowledge Management Theory Applications Workshop, (KMTA 2000), Lyon, France, pp: 165-178, September 2000.
7) Wilks Yorick, “Information Extraction as a Core Language Technology”, International Summer School, SCIE-97, 1997.
8) Ah-hwee Tan, "Text Mining: The state of the art and the challenges", In Proceedings of the PAKDD Workshop on Knowledge Discovery from Advanced Databases, pp. 65-70,1999.
9) Jain, A.K., Murty, M.N., Flynn, P.J., “Data Clustering: A Review”, ACM Computing Surveys, Vol: 31, No: 3, pp: 264-323. 1999.
10) Feldman, R., Sanger, J., “The Text Mining Handbook”, Cambridge University Press, 2007.
11) Seth Grimes, "The Developing Text Mining Market", White paper from Alta Plana Corporation, Text Mining Summit, 2005.
12) M. Grobelnik, D. Mladenic, and N. Milic-Frayling, “Text Mining as Integration of Several Related Research Areas: Report on KDD’2000 Workshop on Text Mining,” 2000.
13) M. Hearst, “Untangling Text Data Mining,” in the Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics, 1999.
14) Alisa Kongthon, "A Text Mining Framework for Discovering Technological Intelligence to Support Science and Technology Management", Technical Report, Georgia Institute of Technology, April 2004.
15) Brijs, Tom, Swinnen, Gilbert, Vanhoof, Koen and Wets, "Using Association Rules for Product Assortment Decisions: A Case Study", In proceedings of Knowledge Discovery and Data Mining, pp: 254–260, 1999.
16) Dong, Jianning, Perrizo, William, Ding, Qin and Zhou, "The Application of Association Rule Mining to Remotely Sensed Data", In proceedings of the ACM symposium on Applied computing, Vol.1, pp: 340–345, 2000.
17) Valentina Ceausu and Sylvie Despres, "Text Mining Supported Terminology Construction", In proceedings of the 5th International Conference on Knowledge Management, Graz, Austria, 2005.
18) Hotho, Nurnberger and Paass, "A Brief Survey of Text Mining Export", LDV Forum, Vol.20, No.2, pp.19-62, 2005.
19) Manning and Schütze, “Foundations of statistical natural language processing”, MIT Press, 1999.
20) Shatkay and Feldman, “Mining the Biomedical Literature in the Genomic Era: An Overview”, Journal of Computational Biology, Vol.10, No.6, pp.821-855, 2003.
21) Pegah Falinouss, "Stock Trend Prediction using News Articles", Technical Report, Lulea. University of Technology, 2007.
22) Nasukawa and Nagano, "Text Analysis and Knowledge Mining System", IBM Systems Journal, Vol.40, No.4, pp.967-984, October 2001.
23) Zhou Chong, Lu Yansheng, Zou Lei and Hu Rong, "FICW: Frequent itemset based text clustering with window constraint", Wuhan University Journal of Natural Sciences, Vol: 11, No: 5, pp: 1345-1351, 2006.
24) Xiangwei Liu and Pilian He, "A Study on Text Clustering Algorithms Based on Frequent Term Sets", Lecture Notes in Computer Science, Vol:3584,pp:347-354, 2005.
25) Le Wang, Li Tian, Yan Jia and Weihong Han, "A Hybrid Algorithm for Web Document Clustering Based on Frequent Term Sets and k-Means", Lecture Notes in Computer Science, Springer Berlin ,Vol: 4537, pp: 198-203, 2010.
26) Zhitong Su ,Wei Song ,Manshan Lin ,Jinhong Li, "Web Text Clustering for Personalized E-learning

Based on Maximal Frequent Itemsets", Proceedings of the 2008 International Conference on Computer Science and Software Engineering , Vol: 06, Pages: 452-455 , 2008.

27) Yongheng Wang , Yan Jia and Shuqiang Yang, "Short Documents Clustering in Very Large Text Databases", Lecture Notes in Computer Science, Springer Berlin ,Vol:4256, pp: 83-93, 2006.

28) Florian Beil, Martin Ester and Xiaowei Xu, " Frequent term-based text clustering", in Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, Edmonton, Alberta, Canada, pp. 436 - 442 , 2002.

29) W.-L. Liu and X.-S. Zheng, "Documents Clustering based on Frequent Term Sets", Intelligent Systems and Control, 2005.

30) Henry Anaya-Sánchez, Aurora Pons-Porrata, and 45) Rafael Berlanga-Llavori, "A document clustering algorithm for discovering and describing topics", Pattern Recognition Letters, Vol: 31, No: 6, pp: 502-510, April 2010.

31) Congnan Luo, Yanjun Li and Soon M. Chung, "Text document clustering based on neighbors", Data & Knowledge Engineering, Vol: 68, No: 11, pp: 1271-1288, November 2009.

32) Zamir O., Etzioni O., "Web Document Clustering: A Feasibility Demonstration", in Proceedings of ACM SIGIR 98, pp. 46-54, 1998.

33) M.H.C. Law, M.A.T. Figueiredo, A.K. Jain, "Simultaneous feature selection and clustering using mixture models", IEEE Transaction on Pattern Analysis and Machine Intelligence, 26(9), pp.1154-1166, 2004.

34) Un Yong Nahm and Raymond J. Mooney, "Text mining with information extraction", ACM, pp. 218, 2004.

35) Lovins, J.B. 1968: "Development of a stemming algorithm", Mechanical Translation and Computational Linguistics, vol. 11, pp. 22-31, 1968.

36) Pant. G., Srinivasan. P and Menczer, F., "Crawling the Web". Web Dynamics: Adapting to Change in Content, Size, Topology and Use, edited by M. Levene and A. Poulovassilis, Springer- verilog, pp: 153-178, November 2004.

37) R. Agrawal, T. Imielinski and A. Swami, "Mining association rules between sets of items in large databases", In proceedings of the international Conference on Management of Data, ACM SIGMOD, pp. 207–216, Washington, DC, May 1993.

38) R. Agrawal and R. Srikant, "Fast algorithms for mining association rules", In Proceedings of 20th International Conference on Very Large Data Bases, Santiago, Chile, pp. 487–499, September 1994.

39) Bjornar Larsen and Chinatsu Aone, "Fast and Effective Text Mining Using Linear-time Document Clustering", in Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, San Diego, California, United States , pp. 16 – 22, 1999.

40) Michael Steinbach, George Karypis and Vipin Kumar, "A Comparison of Document Clustering Techniques", in proceedings of the KDD-2000 Workshop on Text Mining, Boston, MA, pp. 109-111, 2000.

41) B.C.M. Fung, K. Wang and M. Ester, "Hierarchical document clustering using frequent itemsets", in Proceedings of SIAM International Conference on Data Mining, 2003.

42) Reuters-21578, Text Categorization Collection, UCI KDD Archive.

43) 20-newsgroups, "http://kdd.ics.uci.edu/databases/20newsgroups/20newsgroups.html"

44) Webkb dataset, "http://www.cs.cmu.edu/afs/cs/project/theo-20/www/data/"

S.Murali Krishna and S.Durga Bhavani, "An Efficient Approach for Text Clustering Based on Frequent Itemsets", European Journal of Scientific Research, vol. 42, no.3, 2010 (accepted for publication).