Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 4, April 2014, pg.91 – 96

REVIEW ARTICLE

Review on Text Clustering Based on Frequent Itemset

Prajakta Jaswante[1], Dr. P.R. Deshmukh[2]

[1]Student, Sipna COET, Amravati

jaswante.p@gmail.com

p_rdeshmukh@yahoo.com

ABSTRACT

Recently the vast amount of textual information available in electronic form is growing at staggering rate. This increasing number of textual data has led to the task of mining useful or interesting frequent itemsets (words/terms) from very large text databases and still it seems to be quite challenging. The use of such frequent itemsets for text clustering has received a great deal of attention in research community since the mined frequent itemsets reduce the dimensionality of the documents drastically. In the proposed research, we have considered an efficient approach for text clustering based on the frequent itemsets. A renowned method, called Apriori algorithm is used for mining the frequent itemsets. The mined frequent itemsets are then used for obtaining the partition, where the documents are initially clustered without overlapping. Furthermore, the resultant clusters are effectively obtained by grouping the documents within the partition by means of derived keywords. Finally, for experimentation, any of the dataset can be used and thus the obtained outputs can ensure that the performance of the proposed approach has been improved effectively.

Keywords: Text mining, Text clustering, Text documents, Frequent itemsets, Apriori, Reuter-21578

1. INTRODUCTION

The rapid progress of databases in every aspect of human actions has resulted in enormous demand for efficient tools for turning data into valuable knowledge. The entire efforts have resulted in an effective research area known as data mining (DM) or Knowledge Discovery in Databases (KDD) [1]. Text mining is a major research field due to the need of acquiring knowledge from the large number of available text documents, particularly on the Web [2].

Both text mining and data mining are part of information mining and identical in some perspective. Similar to data mining, text mining anticipates mining valuable information from data sources by recognition and searching of interesting patterns. “Text mining” refers to the application of data mining techniques to automated discovery of valuable or interesting information from unstructured text [3,4, 5,6] .

Text mining is a progressively more significant research field since the requirement of attaining knowledge from the massive amount of text documents [7]. It is vital to pre-process the text documents and save the data in a data structure. Typically, text pre-processing includes tokenization, Part of Speech (PoS) Tagging [8], word stemming and the application of a stop words removal technique. Information Extraction is defined as the mapping of natural language texts into predefined structured representation, or templates [9]. Of late, extracting relationships from entities in text documents has achieved significant interest.

In the case of text mining, extracted rules are deduced as co-occurrences of terms in texts and therefore are able to return semantic relations among the terms [10]. Text mining is a multidisciplinary field, which includes these functions: text analysis, information retrieval, clustering, information extraction, categorization, visualization, machine learning, data mining and database technology [11]. The method of dividing data objects (e.g.: document and records) into significant clusters or groups such that objects within a cluster possess analogous characteristics but are contradictory to objects in other clusters is known as Cluster analysis [12], [13]. By arranging a huge number of documents into meaningful clusters, document clustering 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[14].

In this paper, we have presented an effective frequent itemset-based document clustering approach. First, the text documents in the text data are preprocessed with the aid of stop words removal technique and stemming algorithm. Then, top- *p* frequent words are extracted from each document and hence, we form the binary mapped database through the use of extracted words. We apply the Apriori algorithm to discover the frequent itemsets having different length. The mined frequent itemsets are sorted in descending order based on their support level for every length of itemsets. Subsequently, we split the documents into partition using the sorted frequent itemsets. These frequent itemsets can be viewed as understandable description of the obtained partitions. Furthermore, the resultant cluster is formed within the partition using the derived keywords.

2. AN EFFICIENT APPROACH FOR TEXT CLUSTERING BASED ON FREQUENT ITEMSETS

Text clustering is to group a collection of documents (unstructured texts) into different category groups so that documents in the same category group describe the same subject. Many researches [15-16, 17] have investigated 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. Here, we have devised an effective approach for clustering a text corpus with the aid of frequent itemsets. The devised 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.

2.1. Text Preprocessing

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

a) *Stop word Removal:* Removes the stop (linking) words like "have", "then", "it", "can", "need","but", "they", "from", "was", "the", "to", "also" from the document [18].

b) *Stemming algorithm:* Removes the prefixes and suffixes of each word [19].

2.2. Mining of Frequent Itemsets

This sub-section describes the mining of frequent itemsets from the preprocessed text documents *D*. For every document d_i , the frequency of the extracted words or terms from the preprocessing step is computed and the top- *p* frequent words from each document d_i are taken out.

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

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

From the set of top- *p* frequent words, the binary database *B* is formed by obtaining the unique words. Let B_T be a binary database consisting of *n* number of transactions (documents) *T* and *q* number of attributes (unique words) $U= [u_1\ u_2....u_q]$ Binary database B_T consists of binary data that represents whether the unique words are presented or not in the documents d_i .

$B_T=0$ when

$$u_j \notin d_i$$

$B_T=1$ when

$$u_j \in d_i$$

Where 1<=j<=q and1<=i<=n

Then, the binary database B_T is given to the Apriori algorithm for mining the frequent itemsets (words/terms) F_s.

2.2.1 Apriori Algorithm

Apriori is a conventional algorithm that was first introduced in [20] for mining association rules. The two steps used for mining association rules are as follows. (1) Identifying frequent itemsets (2) Generating association rules from the frequent itemsets. Frequent itemsets can be mined in two steps. At first, candidate itemsets are generated and afterwards frequent itemsets are mined with the help of these candidate itemsets. Frequent itemsets are nothing but the itemsets whose support is greater than the minimum support specified by the user. In the proposed approach, we have used only the frequent itemsets for further processing so that, we undergone only the first step (generation of frequent itemsets) of the Apriori algorithm. The pseudo code corresponding to the Apriori algorithm [21] is,

Pseudo code:

C_k **:** Candidate itemset of size k
I_k **:** Frequent itemset of size k.

$I_1 = \{large\ 1\text{-}itemsets\}$;

for $(k = 2;\ I_{k-1} \neq 0;\ k++)$ *do begin*

$C_k = apriori\text{-}gen(I_{k-1})$;

//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;

$I_k = \{c \in C_k \mid c.count >= min\ sup\}$

end

Answer= $\bigcup_k I_k$;

2.3. Partitioning the Text Documents Based on Frequent Itemsets

This section describes the partitioning of text documents *D* based on the mined frequent itemsets *F* .

Definition1:*Frequent itemset* is a set of words that occur together in some minimum fraction of documents in a cluster. The Apriori algorithm generates a set of frequent itemsets with varying length(*l*) from 1 to *k* . First, the set of frequent itemsets of each length (*l*) are sorted in descending order in accordance with their support level.

$F_s = \{F_1 F_2 ... F_k\}$; $1 <= l <= k$

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

where, sup(f*l* (1)) sup(f *l* (2)) ... sup(f*l* (*t*) *)* and *t* denotes the number of frequent itemsets in the set f_l.
Initially, the first element ($f_{(k/2)}(1)$) from the sorted list $f_{(k/2)}$, which is a set of frequent itemsets is selected. Subsequently, an initial partition c_1 , which contains all the documents having the itemset $f_{(k/2)}(1)$, is constructed. Then, we take the second element $f_{(k/2)}(2)$, whose support is less than $f_{(k/2)}(1)$ to form a new partition c_2 . This new partition 2 *c* is formed by identifying all the documents having frequent itemset $f_{(k/2)}(2)$ and takes away the documents that are in the initial partition c_1 . This procedure is repeated until every text documents in the input dataset *D* are moved into partition $C_{(i)}$.
Furthermore, if the above procedure is not terminated with the sorted list $f_{(k/2)}$, then the subsequent sorted lists ($f_{((k/2)-1)}$, $f_{((k/2)-2)}$ etc..) are taken for performing the above discussed step (inserting the documents into partition).This results a set of partition c and each partition $C_{(i)}$ contains a collection documents $D_{c_{(i)}^{(x)}}$

For constructing initial partition (or cluster), we make use of mined frequent itemset which significantly reduces the dimensionality of the text document set and clustering with reduced dimensionality is considerably more efficient and scalable. The clustering results produced by the approaches presented in [22, 23] consist of the overlapping of documents due to the use of frequent itemsets and these overlapping documents have been removed to obtain the final results. In the proposed research, we directly generate the non-overlapping partitions

from the frequent itemsets. This makes the initial partitions disjoint, because the proposed approach keeps the document only within the best initial partition.

2.4. Clustering of Text Documents within the Partition

In this sub-section, we consider how to cluster 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 resulting cluster can detect the outlier documents significantly. Furthermore, the proposed approach does not require a pre-specified number of clusters. The devised procedure for clustering the text documents available in the set of partition *c* is discussed below.

The set of unique derived keywords of each partition $C_{(i)}$ are obtained and the support of each unique derived keyword is computed within the partition. The set of keywords satisfying the cluster support (cl_sup) are formed as representative words of the partition $C_{(i)}$. ***Definition2:*** The *cluster support* of a keyword in $C_{(i)}$ is the percentage of the documents in $C_{(i)}$ that contains the keyword.

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

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

Subsequently, we find the similarity of the documents $D^{(x)}$ with respect to the representative words $R_w[\, c(i)\,]$. The definition of similarity measure plays an importance role in obtaining effective and meaningful clusters. The similarity between two text documents S_m is computed as follows,

$$S \rightarrow K_d\, [\, D_{c_{(i)}}^{(x)}\,]\,, R_w\, [\, c(i)\,] \rightarrow \rightarrow K_d\, [\, D_{c_{(i)}}^{(x)}\,]\ \mathrm{I}\ R_w\, [\, c\,(i)]$$

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

The set of unique derived keywords of each partition $C_{(i)}$ are obtained and the support of each unique derived keyword is computed within the partition. The set of keywords satisfying the cluster support (cl_sup) are formed as representative words of the partition $C_{(i)}$. ***Definition2:*** The *cluster support* of a keyword in $C_{(i)}$ is the percentage of the documents in $C_{(i)}$ that contains the keyword.

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

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

Subsequently, we find the similarity of the documents $D^{(x)}$ with respect to the representative words $R_w[\, c(i)\,]$. The definition of similarity measure plays an importance role in obtaining effective and meaningful clusters. The similarity between two text documents S_m is computed as follows,

$$S \rightarrow K_d\, [\, D_{c_{(i)}}^{(x)}\,]\,, R_w\, [\, c(i)\,] \rightarrow \rightarrow K_d\, [\, D_{c_{(i)}}^{(x)}\,]\ \mathrm{I}\ R_w\, [\, c\,(i)]$$

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

The documents within the partition are sorted according to their similarity measure and a new cluster is formed when the similarity measure exceeds the minimum threshold.

3. EXPERIMENTATION AND PERFORMANCE EVALUATION

For experimentation purpose various datasets can be considered for e.g. documents from different topics or Reuter 21578 dataset.

3.1. Experimental Results

A dataset is considered containing “n” no of documents; they are separated on the basis of Association Rule mining (ARM). Initially, the top frequent words are extracted from each document and the binary database with attributes is constructed. The frequent itemsets are mined from the binary database and the itemsets are sorted based on their support level. Subsequently, initial partition is constructed using frequent itemset. After that, representative words of the each partition are computed based on both the top and familiar words of the partition. The similarity measure is calculated for each document in the partition. The resultant cluster is formed, only if the similarity value of the documents within the partition is below a specific value. So, finally

we get clusters from partitions.

3.2. Performance Evaluation

The following metrics namely, Precision, Recall and F-measure described in [24, 25] can be used for evaluating the performance of the proposed approach. Thus depending on the values of Precision, Recall and F-measure different clusters is formed.

4. Conclusion

Due to the exponential increase in the volume of text document collections and the need for analyzing text documents, several techniques have been developed for mining the frequent associations from text documents. Within the text mining environment, text clustering signifies one of the most effective approaches to group documents in an unsupervised manner. In this paper, we have proposed an effective approach for text clustering in accordance with the frequent itemsets that provides significant dimensionality reduction. We can obtain a set of non-overlapping partitions using these frequent itemsets and the resultant cluster can be generated within the partition for the document collections. We can use the Reuter 21578 dataset for experimentation and the clustering performance of the proposed approach was effectively analyzed.

REFERENCES

[1] 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.

[2] 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.

[3] J. Han, M. Kamber, "Data Mining: Concepts and Techniques", Morgan Kaufmann, San Francisco, 2000.

[4] R. Baeza-Yates, B. Ribeiro-Neto. "Modern Information Retrieval", *ACM Press*, New York, 1999.

[5] 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.

[6] Feldman, R., Sanger, J., "The Text Mining Handbook", *Cambridge University Press*, 2007.

[7] Un Yong Nahm and Raymond J. Mooney, "Text mining with information extraction", *ACM*, pp. 218, 2004.

[8] Manning and Schütze, "Foundations of statistical natural language processing", *MIT Press*, 1999.

[9] Wilks Yorick, "Information Extraction as a Core Language Technology", International Summer School, *SCIE-97*, 1997.

[10] Valentina Ceausu and Sylvie Despres, "Text Mining Supported Terminology Construction", *In proceedings of the 5th International Conference on Knowledge Management*, Graz, Austria, 2005.

[11] Nasukawa and Nagano, "Text Analysis and Knowledge Mining System", *IBM Systems Journal*, Vol.40, No.4, pp.967-984, October 2001.

[12] J. Han, M. Kamber, "Data Mining: Concepts and Techniques", Morgan Kaufmann, San Francisco, 2000.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

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

[20] 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.

[21] 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.

[22] 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.

[23] 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

[24] 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.

[25] 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.

