Maximum Frequent Item Set based Clustering Algorithm for Big Text Data

K. V. Kanimozhi, Rajakumarkrishnan, M.Venkatesan

Abstract**:** ***Due to fast growth of internet and continuous expansion of World Wide Web like digital libraries, online news contributes to massive amount of electronic unstructured text documents on the web. Although lot traditional techniques are available to extract the knowledge from large collection of text documents, still to improve precision of the web search retrieval and to find most appropriate documents from huge text collections proficiently is a big challenge. Clustering techniques helps the search engine to retrieve the documents. The proposed system overcomes existing problems using bivariate n-gram frequent item clustering algorithm by concept of maximum frequent set which maintain the sequence and meaning of sentence in order to reduce huge dimension and and frequent item sets finds similarity. Then based on maximum document occurrence we cluster the documents. Thus our method obtains quality of clusters when compared with existing methodologies and improves the efficiency. The experiment is shown for sample Newsgroup dataset for existing K-Mean and FICMDO (Frequent item clustering method based on maximum document occurrence) and proved the f-measure is higher for our algorithm. Since the f-measure increases, obtains efficient clusters. Hence it is faster and efficient big data method which improves the performance when compared with vector space model like K-Means algorithm.***

Keywords **:** ***Text documents, frequent item set, Similarity, Clustering, Map Reduce.***

I. INTRODUCTION

This Today we live in the world of big data, handling those massive datasets like online news and online digital library is difficult because of its unstructured nature, and nowadays everyone uses web search engine to retrieve related information by submitting a query with word or phrase, while there are millions and billions of web pages, the retrieved pages may or may not return the exact pages, To provide the necessary exact result we need to increase the precision rate of retrieval. Hence clustering documents have been becoming an active re-search topic during the past decade. Many clustering techniques were proposed to place best search results using the ranking and inverted indexing. Clustering groups similar objects (Jiawei Han al., 2012). Nowadays frequent item set based algorithm have been used in text clustering enhances efficiency and accuracy of retrieval for researchers to extract the meaningful information. The paper is planned into different sections where section 2 shows back ground work and recent literature. Section 3 includes problem statement and proposed solution Section 4 shows implementation and evaluates output of the algorithm. Section 5 describes about the explanations of proposed algorithm result compared with existing methodologies. Finally, the conclusion is discussed in Section 6.

Revised Version Manuscript Received on 16 September, 2019.

***** Correspondence Author

K.V.Kanimozhi*,Computer Sscience and Engineering, VIT University Vellore, India. kanimozi.kv214@vit.ac.in.

Dr.Rajakumarkrishnan**,** Computer Science and Engineering, VIT University, Vellore, India. rajakumar.krishnan@vit.ac.in

Dr.M.Venkatesan Computer Science and Engineering, NIT Surathkal Vellore, India. venkisakthi77@gmail.com.

II. BACK GROUND AND LITERATURE REVIEW

Frequent item set is an interesting branch in data mining plays main role in mining the frequent pattern or the common set of items present in the dataset, defined as those item sets must satisfy the minimum support count. The very basic existing algorithms are Apriori algorithm and frequent pattern growth algorithm. In Apriori algorithm using candidate generation, the frequent items are mined at different levels in an iterative approach using prior knowledge. Next using frequent pattern growth algorithm there is no requirement of candidate generation but we need to generate the prefix tree and then the tree is traced to mine the patterns.

(O. Zamir et al., 1998, Hsinchun Chen et.al., 2003) describes about mining from web. Categorizing huge documents into useful clusters, efficient document clustering algorithms plays an important role. Clustering Frequent item based method is one of the recent research topic and essential challenge for researchers. The various problem like social network analysis, bioinformatics, clustering customer behaviours, machine learning document clustering, and sentiment analysis, news collections, text categorization has been analyzed.

Some of the recent literature surveys includes frequent item set based text clustering (K.V.Kanimozhi and M.Venkatesan, 2015, 2016, M. Steinbach et al., 2000), (Yanjun Li et al., 2008) proposed frequent word meaning sequence based clustering the text documents. (Congnan Luo et al., 2009) proposed text document clustering based on neighbours. (Wen Zhang et al., 2010,) implemented Text clustering using frequent itemsets. (F.Beil et.al., 2002, B.Fung et.al., 2003) used clustering technique based on frequent item. (H.Edith et al., 2006) suggested Document clustering based on maximal frequent sequences. All these implementation shows that it works efficiently for selective datasets with limited scalability and not suitable for big data.

Using the Big Data environment few algorithms like K-Mean algorithm implemented by (R.C.Saritha and M.Usha Rani. 2014) using map-reduce text clustering using vector space model. (David C.Anastasiu et al., 2014) suggested using big data (Haoyuan Li et al., 2008) proposed a Parallel FP-Growth for Query Recommendation' (Hongjian Qiu et al., 2014) implemented a new algorithm using spark and showed faster execution than apriori algorithm implemented in map reduce framework.

(Wenhao Wang, Bin Wu, 2011) uses k means clustering algorithm for comparing Twitter and Chinese

Native Micro blog. (LiHong Xu et al., 2016) uses VSM-Cilin method for text similarity in vector space model. (Baoshan Sun et al., 2017 and Lei-lei Shi et al., 2017) uses similarity methods for discovering user interest on social media data's. (Chuanping Hu 2014) uses clustering and searching method for multimedia data.

(Guangyou Zhou and Jimmy Xiangji Huang, 2017) proposed clustering approach for retrieving questions. Even though these algorithms implemented uses map-reduce technique it does not reduce the computation cost, due to huge memory consumption and also does not maintain the meaning and sequence of document sentences hence affects the cluster quality.

III. PROBLEM STATEMENTS AND PROPOSED SOLUTION

The recent problems faced by text clustering method are specifically the massive amount of documents and the huge volume of text document features. A new clustering methodology using map-reduce paradigm is proposed to solve the problem. The proposed system is first to decrease very large number of document terms through new FICMDO method to address the high dimensionality. Then to cluster the similar text documents in an efficient way.

A. *Map-Reduce Framework:*

Due to unprecedented increase of data generated worldwide, conventional techniques are not appropriate to extract, store, manage and analyze due to restricted scalability and also processes only structured data, since most of the data to be analyzed are unstructured in nature, big data map-reduce paradigm has gained important interest in recent years. To process the large scale web documents the Map-Reduce programming has been implemented which does the parallel processing at different levels using frequent items based on bivariate n-gram method and clustering is done based on maximum document occurrence. Thus solves the scalability problem and analyze the data in efficient manner.

B. Preprocessing:

This step is implemented using map-reduce coding for text documents to optimize algorithm by removing the irrelevant words using tokenization and stop words removal. The first step is called tokenization which breaks the every sentence of text document into individual words. Next step is to remove the stop words (a, an, are, at, like ...) which do not posses significant information or occurs very often.

C. Proposed Methodology:

Algorithm

Input: Large Text documents are given as a input to Hadoop file system.

Output: Efficient clusters.

Method:

Step 1: Sentence Break: All the text documents containing the paragraph is converted to sentence.

Step 2: Tokenization: By using Map Reduce pass for all files containing the sentence are converted to key value pairs. Using mapper the tokenization is performed to emit the values and iterative count tokens. And the reducer takes the text key and the iterative values as input from the mapper and calculates the sum of values as result along with the key.

Step 3: Stop word Removal: Load the stop words into a separate file and compare all the document files containing the tokens with the stop word list, and those stop words are eliminated.

Step 4: Finding the Min count: From all the sentence file tokens in every document calculate the minimum count using the mapper and reducer.

Step 5: Item set Generation: The item sets are generated based on the bivariate ngram algorithm iteratively from two words to till minimum count and output all the text with counts.

Step 6: Finding the frequent item set: Mapper tokenizes the item set generation into the key and values and reducer iterate all the values and emit the sum as the maximum occurred frequent items as key and values.

Step 7: Document Similarity Matrix: From the each list of files the words are compared with the frequent item list file and emit the output as reduced document similarity matrix as Item set as key and associated text document files as values. Then the document occurrence is calculated by taking the matrix input and emits the document name and the values as occurrences.

Step 8: Clustering: Maximum occurred text documents is chosen as centre point and all the associated documents are clustered iteratively and output as a separate file from the closed maximum.

Our proposed algorithm works in parallel manner using map reduce approach by taking the large input text documents from the local system to hadoop system to produce high quality clusters. Initially the proposed algorithm takes the text document as input and converts each document into separate sentences and then pre-processing steps like stop words are removed.

Then tokenization is applied for all documents and the minimum count is obtained. Then for each pre-processed document generate the item sets using bivariate n-gram mechanism by pairing minimum two words for generating item set and the pairing is continued till minimum count.

After that the occurrence of each word pair is calculated and the model finds out the item set which have occurred maximum times from the generated item sets and outputs frequent item set. Once frequent items are generated we have to calculate similarity between the documents by number of item set they have in common and construct document matrix as per occurrence by generating the frequent item set and corresponding documents. Next step is to calculate document occurrence by finding number of times each document occurs and select centre point based on maximum document occurrence and perform clustering as per centre point and output the final set of clusters.

D. Case Study: Clustering the sample Text documents using proposed Algorithm.

1. Input Text files:

The input Datasets is 20 Newsgroup data sets, all these datasets are changed to original text format and pre-processing techniques like tokenization, stop words removal are applied. Then it is copied to hadoop file system from local file system.

2 Finding the Minimum count:

After counting the number of words in each sentences of every document the minimum count is obtained and shown in Table1.

Table1: Obtain minimum count by counting the number of words.

Content of document D1	No. of Words
I am going to office.	5
Rama killed Ravana	3
The baby is sleeping	4
Good evening everyone	3

The minimum count in above table is 3.

3. Item sets generation:

Frequent item sets are obtained by means of bivariate n-gram method. Minimum two words are considered for obtaining frequent item and 'variate' because it depends on n=2 and n=minimum count the minimum two words are taken till it reaches the minimum support count. And n is the number of words to be paired together,

Example: the baby is very cute.
1st pairing the baby; baby is; is very; very cute
2nd pairing the baby is; baby is very; is very cute
3rd pairing the baby is very; baby is very cute
4th pairing the baby is very cute

Likewise the pairings are done and the item sets are obtained, and then number of times each pair occurred is calculated for sample two items is shown in Table 2.

Table 2: Obtain the occurrence of each item pair.

Item set	Occurrence
The baby	50
Eat healthy	55
Where are	20
Buy fruits	15
Hi everyone	45
Good evening	65

4. Finding the Frequent item sets:

The most occurred item sets from the generated item sets are selected given in Table 3.

Table 3. Maximum occurred frequent item set from Table 2.

Item set	Occurrence
The baby	50
Eat healthy	55
Where are	20
Hi everyone	45
Good evening	65

5. Calculate similarity as per item set and construct document matrix as per frequent item set occurrence.

The similarity is computed by the number of frequent item set they have similar between two documents. And Table 4 contains document matrix as per frequent item set.

Table 4. Reduced document similarity matrix obtained from Table 3.

Item set	Text documents
The baby	1.txt, 5.txt, 8.txt, 12.txt
Eat healthy	1.txt, 4.txt, 14.txt, 2.txt, 3.txt
Where are	3.txt, 1.txt, 6.txt, 10.txt
Hi everyone	1.txt, 5.txt, 9.txt, 12.txt
Good Evening	4.txt, 7.txt, 11.txt,13.txt,14.txt

6. Calculate document occurrence.

Once the document matrix is found, the document occurrence is calculated and shown in Table 5.

Table 5.Document occurrence obtained from Table 4.

Document	Occurrence
1.txt	4
3.txt	2
5.txt	2
4.txt	2
12.txt	2
14.txt	2
2.txt	1
6.txt	1
8.txt	1
7.txt	1
11.txt	1
10.txt	1
9.txt	1
13.txt	1

7. Select pivot points based on

Retrieval Number: B15390982S1119/2019©BEIESP
DOI: 10.35940/ijrte.B1539.0982S1119

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

maximum occurred documents and perform Clustering as per pivot.

Based on maximum occurred documents the pivot points are selected and clustering are obtained and shown in Table 6.

Table 6.Final Clusters obtained from Table 5.

Cluster	Pivot	Documents
1	1.txt	1, 8, 11, 12
2	3.txt	3, 4, 14
3	4.txt	4, 9

IV. EXPERIMENT AND RESULTS ANALYSIS

The algorithm is implemented using the Map reduce paradigm and performance are estimated on a Ubuntu 14.04 PC with 4GB RAM and Intel Core i3 processor in hadoop 2.6.0 for sample example and the accuracy of clustering is shown by comparing the proposed algorithm with existing methodologies.

1. Evaluation for Computing Frequent Item Sets

The algorithm is tested for scalability by evaluating frequent item sets. Algorithm uses bivariate ngram mechanism for calculating frequent 2-pair items. This method proves by eliminating infrequent words, whole database size decreases dramatically. From the sample 20_Newsgroup data set containing the 100 documents sample has been taken. The bivariate ngram algorithm generates 36,996 2-pair item sets and from these item sets extracts 293 frequent 2-pair item sets when the minimum support is 2. Scalability is tested by increasing more datasets, and output is given in Fig.1. The proposed approach takes less execution time than FP-growth algorithm. It takes more execution time than Apriori algorithm for smaller dataset, and for increased dataset, our approach takes less time than Apriori. Hence Apriori and FP-growth algorithms are not suitable for big data and the scalability is shown in Fig.2.

Fig. 1 Execution time when compared with our Apriori, FP growth and bivariate n-gram method.

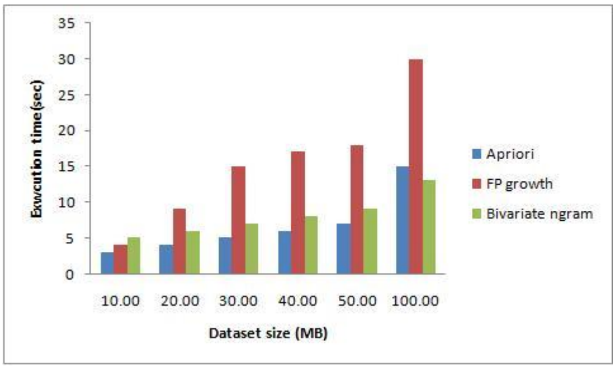


Fig. 2 Execution time of frequent item set in terms of increasing Dataset.

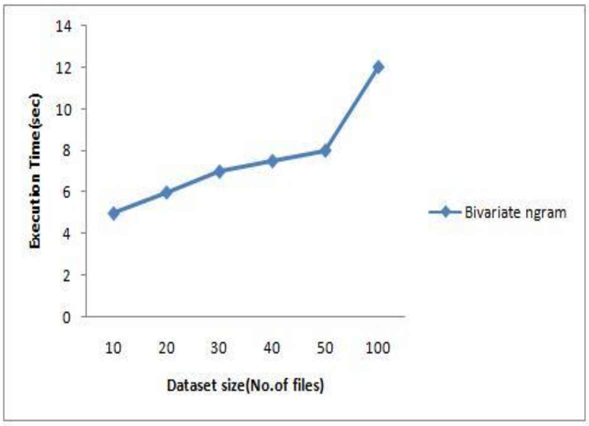
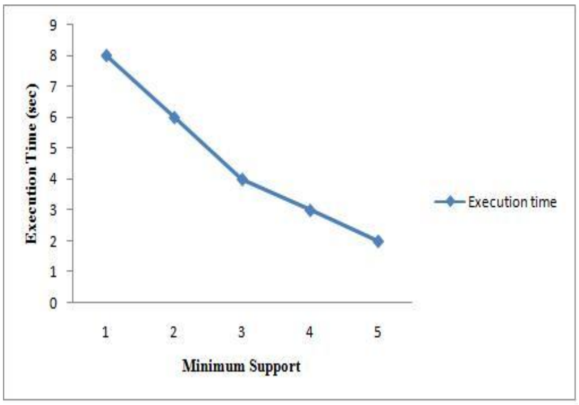


Fig. 3 Execution time of finding frequent item sets versus Minimum Support.



The algorithm is analyzed with different minimum support counts for 20_Newsgroup dataset, and the result obtained is shown in Fig. 3.

Whenever minimum support count increases, the computation time decreases. Hence the output neatly shows the new implemented algorithm is highly scalable.

2. Comparison of our approach with K-means algorithm.

We extract subsets D1, D2 and D3 from the 20Newsgroup dataset which contains around 50, 100 and 500 documents respectively. Table 7, Table 8 and Table 9 shows the results of (Rajesh Malviya and Pranita Jain, 2015) method called clustering using K-means and our approach FICMDO for for Fig.4 Precision, Fig.5 Recall and Fig.6 F-measure. Since f-measure is high it proves better clustering when compared to K-means.

Table 7. Precision Output

Dataset	Precision	
	K-Means	FICMDO
D1	0.4646	0.61
D2	0.4659	0.62
D3	0.6258	0.68

Table 8. Recall Output

Dataset	Recall	
	K-Means	**FICMDO**
D1	0.381	0.43
D2	0.5273	0.48
D3	0.551	0.54

Table 9. F-Measure Output

Dataset	F-Measure	
	K-Means	**FICMDO**
D1	0.4187	0.5
D2	0.4947	0.54
D3	0.586	0.62

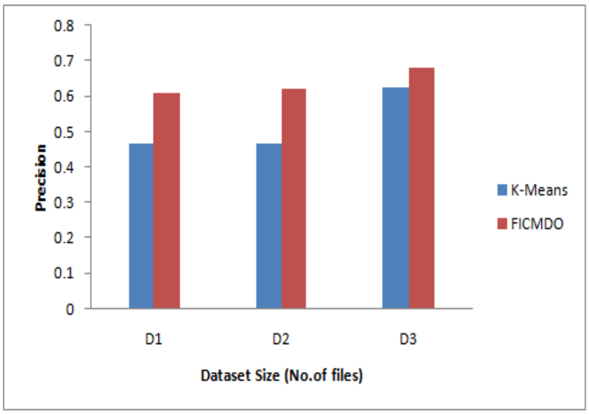


Fig 4: Precision of K-Means and FICMDO Approach

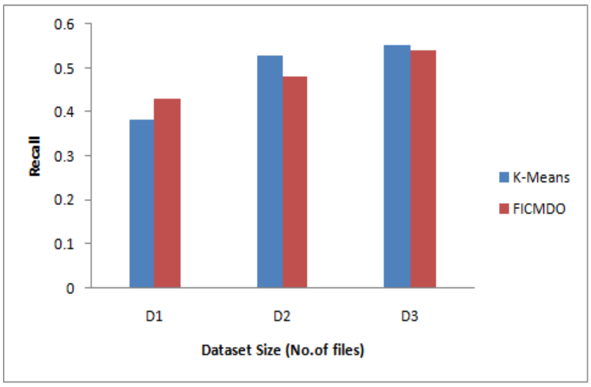


Fig 5: Recall of K-Means and FICMDO Approach

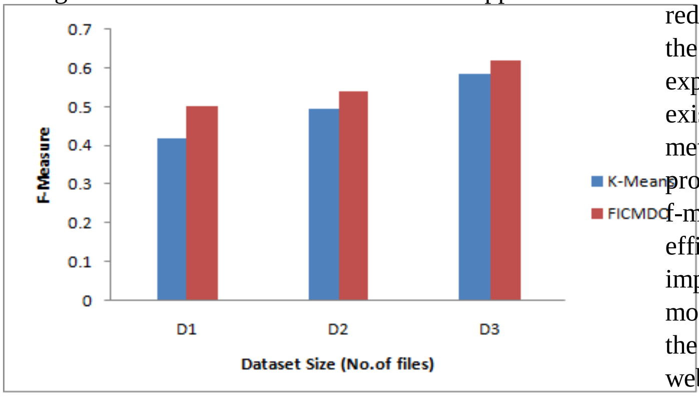


Fig 6: F-measure of K-Means & FICMDO Approach

V. EXPLANATIONS

1. Why bivariate n-gram frequent item based clustering works better than existing Apriori, frequent pattern growth and vector based Approach.

Since clustering is significant technique for information retrieval of web documents from world wide web text database, there are so many traditional techniques like partitioning based clustering, hierarchal based clustering has been used and called as vector based approach where all documents are considered as collection of words which leads to huge dimensionality and does not bother about the sequence and meaning of sentence, hence it effects the quality of clusters.

Hence recently frequent item based clustering has been used to overcome the problem of vector space model, In Apriori method, generation of larger number of candidate sets and matching those patterns in whole database leads to memory consumption and repeated scan of whole database is a major drawback and hence cost is high.

Next even though frequent pattern growth algorithm do not generate candidate item set, generating frequent pattern tree for those huge dataset is a problem and mining recursively requires more memory usage and takes longer computation time.

A main challenge is whenever minimum support count set to low a huge number of item sets are generated. Hence our proposed system addresses all above mentioned problem using bivariate ngram frequent item clustering algorithm uses the concept of maximum frequent set in order to reduce huge dimension and maintain the sequence and meaning of sentence and as per frequent item set similarity is calculated. Then based on the maximum document occurrence we cluster the documents. Thus our method obtains quality of clusters when compared with existing methodologies and improves the efficiency.

VI. CONCLUSION

This work tests the result by comparing our proposed frequent item based new clustering algorithm based on maximum document occurrence for web documents using map-reduce paradigm with existing methods like Apriori, frequent pattern growth and proves improved result by reducing the huge dimension , computation cost and solves the scalability problem and also provides quality clusters. The experiment is shown for sample Newsgroup dataset for existing K-MEAN and FICMDO (frequent item clustering method based on maximum document occurrence) and proved the f-measure is higher for our algorithm. Since f-measure increases, proves the clusters obtained are highly efficient. Hence its faster and efficient big data method which improves the performance when compared with vector space model like k-means algorithm. The future work will include the ranking of documents and will be tested for other type of web documents and analyzing more details including topics.

REFERENCES

1. Edith, E., Rene, A.J., J.A.Carrasco-Ochoa, J.A. and Martinez-Trinidad., J.F. , "Document clustering based on maximal frequent sequences", *in: Proceedings of FinTAL* 2006, LNAI, vol. 4139, 2006, pp. 257-267.

2. Beil, F., Ester, M. and Xu, X. (2002), "Frequent term based text clustering". *in: Proceedings of the ACM SIGKDD International Conference on knowledge Discovery and Data Mining*, 2002, pp.436-442.

3. Fung, B., Wang, K. and Ester, M. (2003), "Hierarchal document clustering using frequent item sets". *in: Proceedings of the 3rd SIAM International Conference on Data Mining*, 2003.

4. Haoyuan Li, Yi Wang and Dong Zhang., "Parallel FP-Growth for Query Recommendation", in RecSys 08. *In: Proceedings of the 2008 ACM conference on Recommender Systems*.pp107-114.

5. Hongjian Qiu, Rong Gu, Chungfeng Yuan, Yihua Huang and YAFIM. , 'A Parallel Frequent Item set Mining Algorithm with Spark." *In: 28th International Parallel & Distributed Processing Symposium Workshops. 2014. IEEE.*

6. Yanjun Li, Soon M.Chung and JohnB.Dolt. "Text document clustering based on frequent word meaning sequences", *Data and Knowledge Engineering*,64 (2008), 381-404, Elsevier.

7. Congnan Luo, Yanjun Li, and Soon M.Chung. "Text document clustering based on neighbors". *Data and Knowledge Engineering*, 68(2009), 1271-1288, Elsevier.

8. Wen Zhang, Taketoshi Yoshida. Xijin Tang, Qing Wang., "Text Clustering using Frequent item sets", *Knowledge-based Systems*, 23(2010), 379-388, Elsevier.

9. Kanimozhi, K.V. and Venkatesan, M., "Survey on Text Clustering Techniques", *Advanced Research in Electrical and Electronic Engineering*, Vol 2, Issue 12, 2015, 55-58.

10. Kanimozhi, K.V. and Venkatesan, M. "Big Text Datasets Clustering based on Frequent Item Sets - A Survey*", International Journal of Innovative Research in science and Engineering*, Vol.No.2, Issue 5.May 2016.ISSN: 2454-9665.

11. Jiawei Han, Micheline Kamber and Jian Pei. *Data Mining Concepts and Techniques*, Third Edition, Elsevier.

12. Saritha, R.C and Usha Rani, M "Map Reduce Text Clustering Using Vector Space model" , *International Journal of Emerging Technology and Advanced Engineering*. Vol 4, Issue 9, September 2014.

13. M. Steinbach, M., Karypis, G. and Kumar, V. "A comparison of document clustering techniques", *KDD-2000 Workshop on Text Mining*, 2000.

14. Rajesh Malviya and Pranita Jain. "A Novel Text Categorization Approach based on K-means and Support Vector Machine*", Samrat Ashok Technological Institute', Department of Information Technology*, Vidisha, M.P., India - November 2015

15. David C.Anastasiu., Jeemy Iverson., Shaden Smith and George Karypis. "Big data Frequent Pattern Mining", in C.C.Aggarwal,J.Hans(eds.) *Frequent Pattern Mining*, Springer .pp.225-259

16. Zamir, O. and Etzioni, O. "Web Document Clustering: A Feasibility Demonstration", *Research and Development in Information Retrieval*, pp.46-54.

17. Hsinchun Chen and Michael Chau. "Web Mining: Machine learning for web applications", *Annual Review of Information science and technology* , 2003.

18. Wen hao Wang, Bin Wu., "Comparing Twitter and Chinese Native Micro blog" *in IEEE Conference 2011: Proceedings of the Cyber security Summit (WCS)*, London, UK.

19. LiHong Xu, ShuTao Sun, Qi Wang, "Text Similarity Algorithm based on SemantiC vector Space Model" *in IEEE 2016: Proceedings of the Computer and Information Science (ICIS),2016 IEEE/ACIS 15th International Conference*. Doi: 10.1109/ICIS.2016.7550928.

20. Baoshan Sun, Aoshan Sun and Lingyu Dong. "Dynamic Model Adaptive to User Interest Drift Based on Cluster and Nearest Neighbors", *IEEE Access*. Vol.5, pp. 1682-1691.

21. Lei-lei Shi, Lu Liu, Yan Wu, Liang Jiang and James Hardy. "Event Detection and User Interest Discovering in Social Media Data Streams" *IEEE Access*.DOI 10.1109/ACCESS.2017.2675839,

22. Zhenwen Dai and Jorg Lucke. "Autonomous Document Cleaning— A Generative Approach to Reconstruct Strongly Corrupted Scanned Texts", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 36, No. 10.

23. Chuanping Hu, Zheng Xu, Yunhuai Liu, Lin Mei, Lan Chen, and Xianfeng Luo. "Semantic Link Network-Based Model for Organizing Multimedia Big Data", *IEEE Transactions on Emerging Topics in Computing*. Doi.10.1109/TETC.2014.2316525

24. Guangyou Zhou and Jimmy Xiangji Huang. "Modeling and Learning Distributed Word Representation with Metadata for Question Retrieval", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 29. No. 6, June 2017.

AUTHORS PROFILE

K.V.Kanimozhi Research Scholar VIT University, Department of Computer Science and Engineering, have published various papers in national and international journals. Her area of interest includes Data mining, Big data Analytics, Data Science.

Dr.Rajakumar Krishnan Associate Professor, School of Computing Science and Engineering have published various papers in national and international journals. His area of interest includes image processing, Computational Intelligence.

.

M.Venkatesan Assistant Professor, Department of Computer Science and Engineering, National Institute of Technology have published various papers in national and international journals. His area of interest includes Data mining, Big data Analytics, Data Science, Database and Geo Spatial.