ISSN: 0974-5823 Vol. 7 No. 1 January, 2022

International Journal of Mechanical Engineering

FEATURE EXTRACTION FOR UNSTRUCTURED TEXT DOCUMENTS USING MEEM MODEL

Ponmani K[1] Thangaraj M[2]

[1]*Research Scholar, Department of Computer Science, Bharathiar University, Coimbatore, India*

[2]*Professor, Department of Computer Science, Madurai Kamaraj University, Madurai, India*

Abstract

Designs for text analysis fuse an assortment of procedures like text arrangement, classification, and Clustering. Every one of them expects to uncover stored away connections, patterns, and examples which are a strong base for business decision-making.This research paper examines the problem of finding features in an unstructured text document using tf-idf values in clustering. Analyse this problem using different techniques. K-Means clustering and Hierarchical clustering (HC) are widely accepted clustering methods. These two clustering algorithms cannot be the best choice in all the applications. The proposed methodology Modified Efficient Expectation Maximization (MEEM) clustering model helps to extract the features in an efficient manner. When compared proposed MEEM model with standard approaches such as KMeans and HC clustering that shows remarkable improvement in the performance metrics.

Keywords: Clustering, MEEM, KMeans, HC, Unstructured, Document

1. Introduction

Text mining aims to identify (Aljaber, et al. 2010) non-trivial, implicit, previously unknown and potentially useful patterns. Information can be extracted to derive summaries of the words contained in the documents or to compute summaries for the documents based on the words contained them. Given a set of documents, each with a label called the class label for that document. The text-clustering task (Chen, et al. 2011) is to arrange a set of the unstructured text document into clusters such that the documents within each cluster are similar to each other. Various techniques for text clustering have been developed.

(Consoli, et al. 2009) clustering approaches can be categorized as agglomerative or partitioned clustering based on the underlying methodology of the algorithm, or as hierarchical or non-hierarchical clustering based on the structure of the final solution. In recent years, there has been considerable interest in formulating the feature selection problem in unsupervised settings using mixture models learned using different clustering algorithms. The primary objective is the identification and the reduction of the influence of extraneous features that do not contribute information about the true cluster structure. In text clustering, (Fan, Bouguila and Ziou 2012) text is generally mapped to a vector space, and each document becomes a feature vector using a weighting scheme. Clustering is then performed by measuring distance between vectors. The vector space mapping raises problems: the high dimensionality of the (Dai and Lücke 2014) feature space and the data sparsity. The vector space model may provide high performance clustering, but an additional process is required to get keywords or a summarized description from obtained clusters. To tackle these problems MEEM clustering algorithm was proposed. It can provide brief summaries of large documents (Hu, Park and Zhang 2009). It can be used to find the features in text document using tf-idf values in an efficient way. MEEM model is a probability based clustering and iterative model. Sensitivity to noise and outliers depend on the distribution. The rest of the paper organized as follows: Section 2 describes the details of the related work. In Section 3 presents the details of unsupervised feature selection model and describes it as a probabilistic Bayesian model. In section 4 describes the data sets used in MEEM model and the results of MEEM model. Section 5 presents the performance analysis of MEEM. Section 6 closes with conclusions, discussions, and future directions.

2. Literature Review

KMeans (Han, et al. 2016) developed the procedure follows a straightforward and simple approach to order a given informational index through a specific number of groups fixed apriori. In 2010, Jiang, Liou and Lee showed the result, it produce each point suitable to a given data set and inferior it to the nearest center. When none of the point is spending, the first step is accomplished. At this point need to re-work out k new centroids.

In 2013, Lin, Jiang and Lee 2013 implemented document clustering algorithm using K-Means. It is sensitive to the initial cluster centers, followed with an iterative loop to update fitness value until a certain stopping condition is met. In 2013, Xiong, Azimi and Fern developed clinical disease prevention, this allows individuals identified at different risk tiers benefit from further investigation and intervention. In 2017, Xu, et al. proved simultaneously integrates the limited supervised information and the size constraints to screen the high-risk population based on similarity measurement, and get a feasible and balanced stratification solution to avoid cluster with few points.

In 2010, Guan, et al. developed hierarchical clustering algorithm works by grouping the data one by one on the basis of the nearest distance measure of all the pair wise distance between the data point. In 2013, Shabtai, et al. implemented Hierarchical clustering algorithms proved that it was not scalable for very large data due to their non-linear complexity. In 2012, Sim, et al.

used hierarchical clustering algorithm, improved the purity of the clustering algorithm, and reduce the chaining factor. In 2011, Skabar and Abdalgader used the hierarchical data for groupings in text corpus.

3. Proposed Work

3.1 Problem Identification

Clustering is an important part of data mining and a fundamental means of knowledge discovery in data exploration. Fast and high-quality document clustering algorithms (Park, et al. 2019) play an important role in given that natural triangulation and browsing mechanisms as well as in enabling knowledge management. In modern years, it has perceived a great growth in the volume of text documents available on the Internet, digital libraries, news sources, and companywide intranets. This has managed to an augmented attentiveness in developing (Xu, et al. 2017) methods that can help users effectually traverse, encapsulate, and establish this information with the goal of helping them find what they are looking for. Fast and high-quality document agglomeration algorithms play a very important role toward this goal as they need been shown to supply each an intuitive navigation/browsing mechanism by organizing massive amounts of information into a tiny low variety of purposeful clusters additionally on greatly improve the retrieval performance either via cluster-driven spatial property reduction, term-weighting, or query development.

3.2 Problem Description

To find the features in unstructured data using tf-idf is the difficult task in the field of text mining. In this research examine the features (Yan, Chen and Tjhi 2013) in an unstructured data using tf-idf in an efficient way. The proposed MEEM algorithm is used to find the features in an unstructured text data. Information can be (Kao, et al. 2010) extracted to derive summaries of the words contained in the documents or to compute summaries for the documents based on the words contained them. Given a set of documents each with a label called the class label for that document.

3.3 MEEM Architecture

Figure 1 shows the proposed MEEM architecture. This is a layered architecture. Presentation Layer is used to present data to the application layer in an accurate, well-defined

and standardized format. It contains the component of Graphical User Interface (GUI) and Applications. GUI is a type of user interface that allows users to interact with through graphical icons instead of text-based user interfaces, typed command labels or text navigation. When creating an application, many object-oriented tools exist that facilitate writing the graphical user interface. Each GUI element is defined as a class widget from which can create object instances for an application. Business Layer serves as an intermediary for data exchange between the Presentation Layer and the Data Layer. This layer contains the component of Data Cleansing, Classification, (Huang, et al. 2012) and Clustering. Before using the data it is necessary to pre-process the data. To pre-process the data the following methods such as Tokenization, Stemming, Prefix, and Stop words are used. Tokenization: Tokenization is the activity of infringement up a grouping of text into contentions, maxims, expressions, images, and different components called tokens. Stemming: the term used to describe the process for reducing inflected words to their words stem, base or root. The stem need not be identical, it is usually sufficient that related words map to the same stem. Prefix: A prefix is a group of letters placed before the root of a word. Stop Words: Stop words are words which are filtered out before or after processing of data. Stop words usually refer to the most common words. In order to save space and time, these words are needed to be dropped. Classification is related to categorization. The task is to assign a document to one or more classes or categories. Classes are selected from a previously established taxonomy. The Text Classification takes care of all (Janani and Vijayarani 2019) pre-processing tasks required for automatic classification. Documents may be classified according to their attributes. In machine learning, (Iam-On, et al. 2010) naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong independence assumptions between the features. It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, (Song, et al. 2012) a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. Naive Bayes model is easy to build and particularly useful for very large datasets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

Clustering is the process (Wu, et al. 2014) of making a group of abstract objects into classes of similar objects. MEEM clustering is a model-based clustering. In this method, a model is hypothesized for each cluster to find the best fit of data for a given model. This method locates the clusters by clustering the compactness function. It replicates the spatial dissemination of the data points. This technique also affords a way to repeatedly regulate the

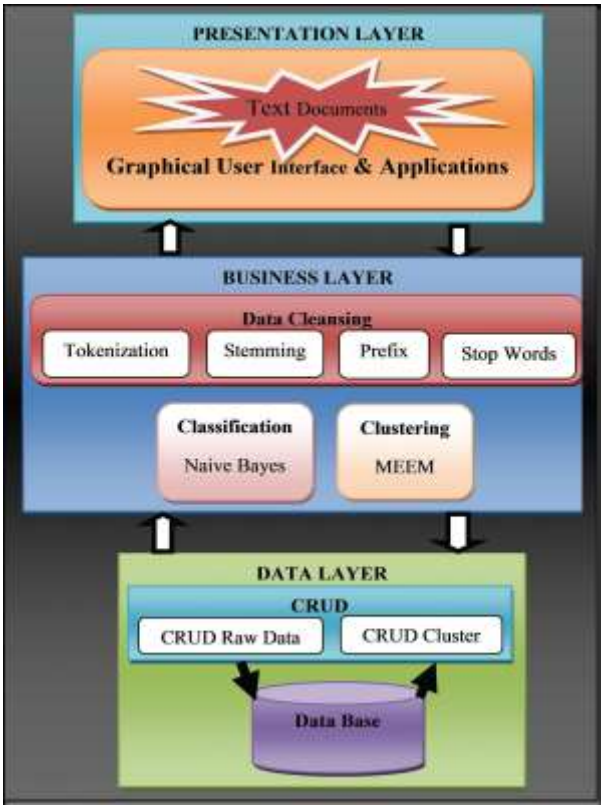

Fig. 1. Proposed MEEM Architecture.

number of clusters based on standard statistics, taking outlier or noise into justification. It, therefore, yields tough clustering method. Data Layer (DL) separates the data access logic from the presentation logic. However, while the DL cleanly separates the data access details from the presentation layer, it does not enforce any business rules that may apply. This layer contains the component of CRUD (Create, Retrieve, Update, and Delete).The CRUD cycle describes the elemental functions of a persistent database.

3.4 Algorithm

MEEM is well-known with KMeans in that it substitutes between an assumption step (E-step), identical to movement, and an enlargement step (M-step), comparable to recompilation of the restrictions of the model. The margins of KMeans are the centroids. The maximization step recomputed the conditional parameters. Document d is generated according to probability distribution. Components generate the document using its own parameters.

Generate a document selecting a cluster with probability. Generating the terms of the document according to the parameter. These maximum likelihood estimates maximize the likelihood of the data given the model. The E-step is responsible to estimate the parameters of the probability distribution of each cluster. This algorithm stops when the distribution parameters are converges or reach the maximum number of iterations.

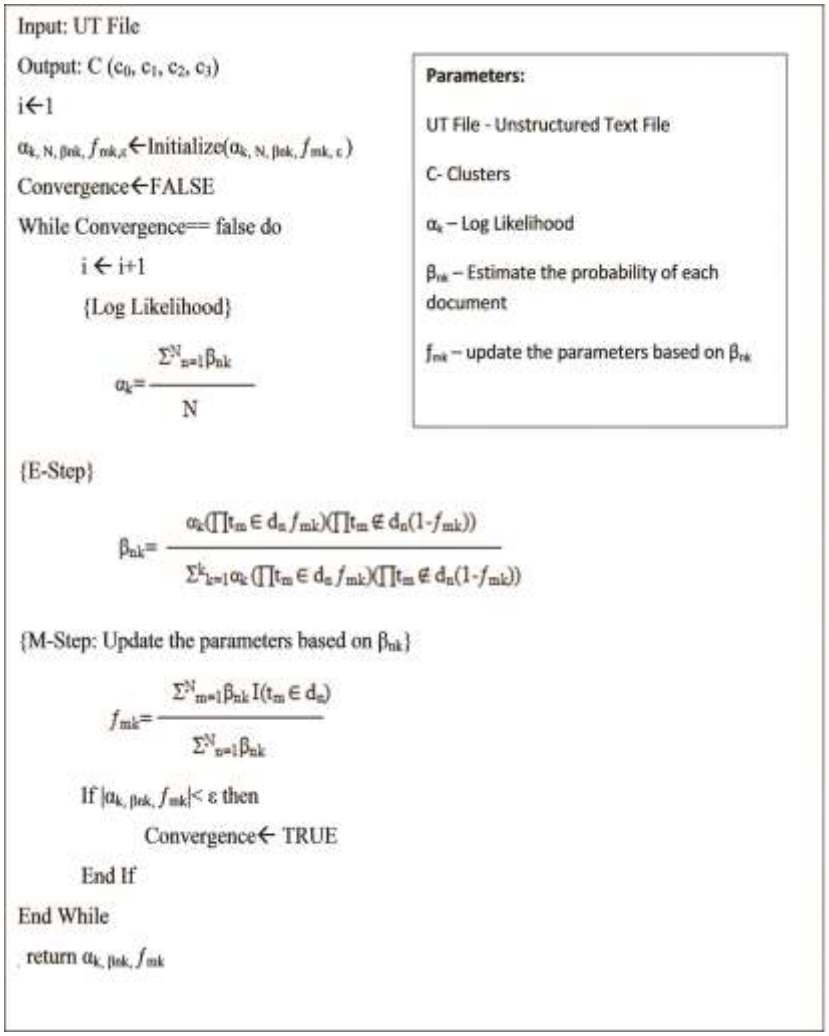

Fig.2.Proposed MEEM Algorithm

Figure 2 shows the proposed MEEM algorithm. It takes the input of unstructured text file and the output is the number of clusters. It initializes all parameters. If the convergence is false find the log likelihood. Increment the value; find the probability of each document (E-Step), and then update the parameters (M-Step) based on E-Step. If the convergence is true return all the parameters. Due to the speed of the algorithm in working with only two mixture components, the process of the E-Step and M-Step can be iterated repeatedly until the model parameters do not change by a specified ε where ε is a small number, such as 0.0001.

4. Experimental Results

In this section, the accuracy and efficiency of MEEM clustering algorithm are compared with the earlier systems KMeans and HC clustering. All the experiments are executed on Intel Core i5 3.40 GHz with 8 GB RAM, running Windows 10 64 bit Operating system and implemented using Java based on WEKA (Ian and Frank, 2000), R Tool, and eclipse with java.

4.1 MEEM

After classification the data set contains 124 classes, 26109 numbers of documents, and the 251152 number of features. These features are detected using their tf-idf weight.

TF (tf) means term frequency, ($tf_{t,d}$) of term t in document d is defined as the number of times that the term t occurs in d. In Inverse document frequency (idf), which diminishes the weight of normally utilized words and expands the weight of words that are not used very much in a collection of documents. This can be combined with term frequency to calculate a term's tf-idf, the frequency of a term adjusted for how rarely it is used. It is expected to gauge how significant a word is to a report in an group of documents. The inverse document frequency for any given term is defined as ($idf_{t;d}$)=log(N=df_t) Where N-Number of documents, t-terms (features). The tf-idf weight of a term is the product of its tf-weight and its idf weight.

Figure 3 shows the classification results to acquire the features using tf-idf values. The figure shows the highest tf-idf values that the data set contained are in boldface. The number of features is calculated using tf-idf values. The data set having the different number of features.

Fig. 3. Features Obtained by MEEM

4.2 Comparison of MEEM VS K-Means

Figure 4 indicates the number of clusters obtained from MEEM clustering algorithm. This graph is drawn with the x-axis parameter value of data set and the y-axis parameter value of cluster mean. The purpose of this graph is that the proposed MEEM clustering achieves highest mean values and the samples are best clustered compared to KMeans and HC clustering.

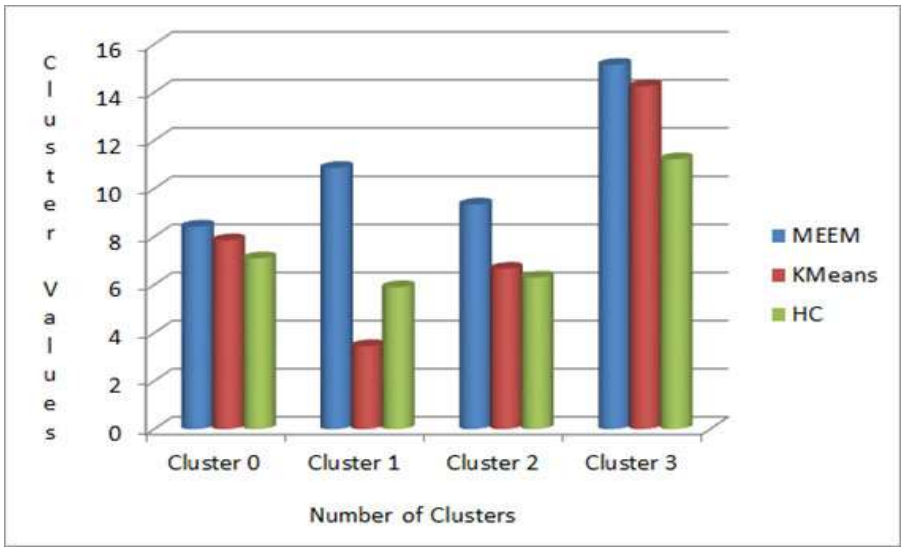

Fig. 4. Clusters Obtained by MEEM

5. Performance Analysis of MEEM

The results of MEEM are compared with the KMeans and HC clustering by using the following measures such as

- Precision

It measures the accuracy of the retrieval model by calculating the percentage of correctly retrieved documents to the whole retrieved result set. It is defined as

Precision = Correctly Retrieved Documents / Total Number of Retrieved

- Recall

It measures the ability of the retrieval model to retrieve correct documents from the whole data set by calculating the percentage of correctly retrieved documents to all the documents that should be retrieved. It is defined as

- F Recall = Correctly Retrieved Documents / Total Number Documents

It measures the harmonic average of precision and recall. It is defined by

Figure 5 indicates F-Score = 2*Precision*Recall / Precision + Recall drawn with the x-axis parameter value of data set and the y-axis parameter value of performance analysis values. The purpose of this graph is that the proposed MEEM clustering achieves highest precision, recall and F-Score values and the samples are best compared to KMeans and HC clustering.

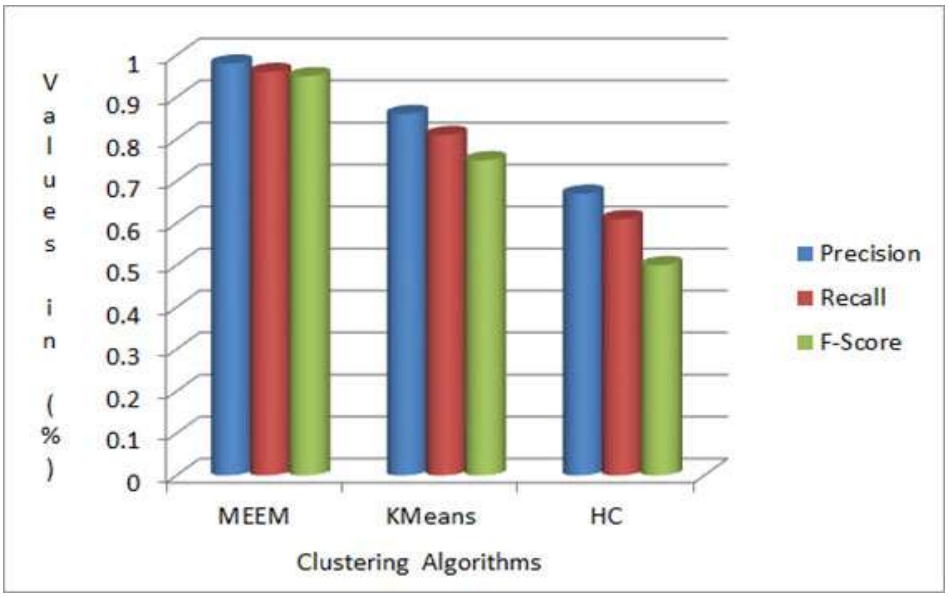

Fig. 5. Performance Analysis (Precision, Recall, F-Score) of MEEM.

- Entropy

Entropy evaluates the quality of a clustering by measuring how different categories of documents distributed within each cluster. Quality of clustering is better if entropy is small It is defined by

$$\text{Entropy} = -1/\log(\sum^{k}_{c=1} n_i^c / Ln_i)(\log n_i^c / n_i)$$

Figure 6 shows the comparison between the quality of MEEM ,KMeans, and HC clustering. This graph is drawn with x-axis takes the value of data set, and y-axis takes the value of entropy. This graph indicates that the proposed MEEM achieves smaller entropy to shows the better quality of clustering compared with KMeans and HC clustering algorithms.

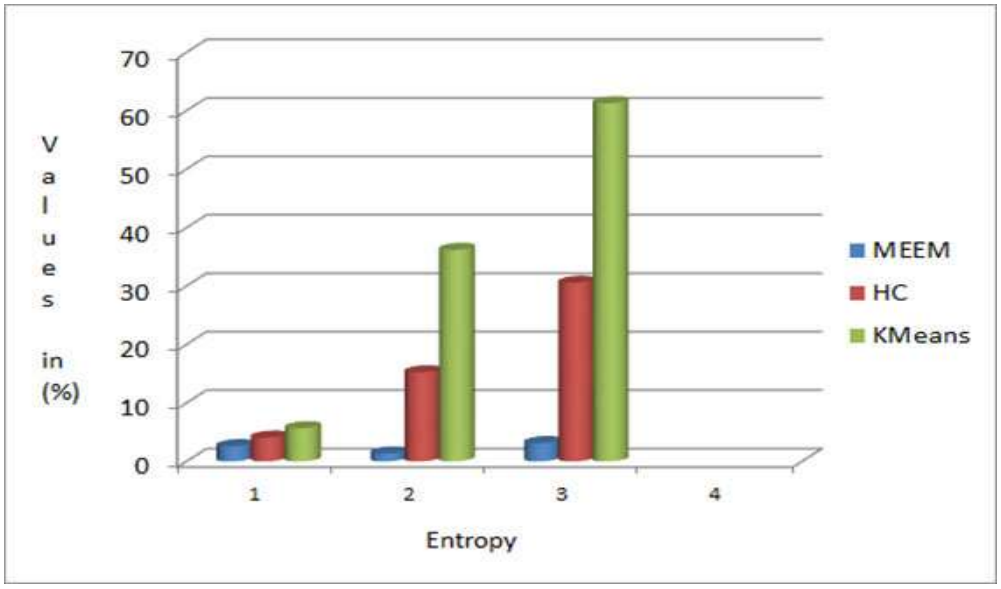

Fig. 6. Entropy of MEEM.

- KB Information

Kononenko and Bratho suggested to calculate the amount of information gained or lost in the clustering of each sample and then in the clustering of the whole data set. In prior, the amount of information necessary for confirming that e is in class c is (-logPe(c)) bits, the amount of information necessary to correctly divide that e does not belong to c is (-log(1-Pe(c)) bits. The information

score is the difference between the quantity of gained information and the quantity of lost information. Figure 7 indicates the KBInformation of MEEM clustering algorithm.

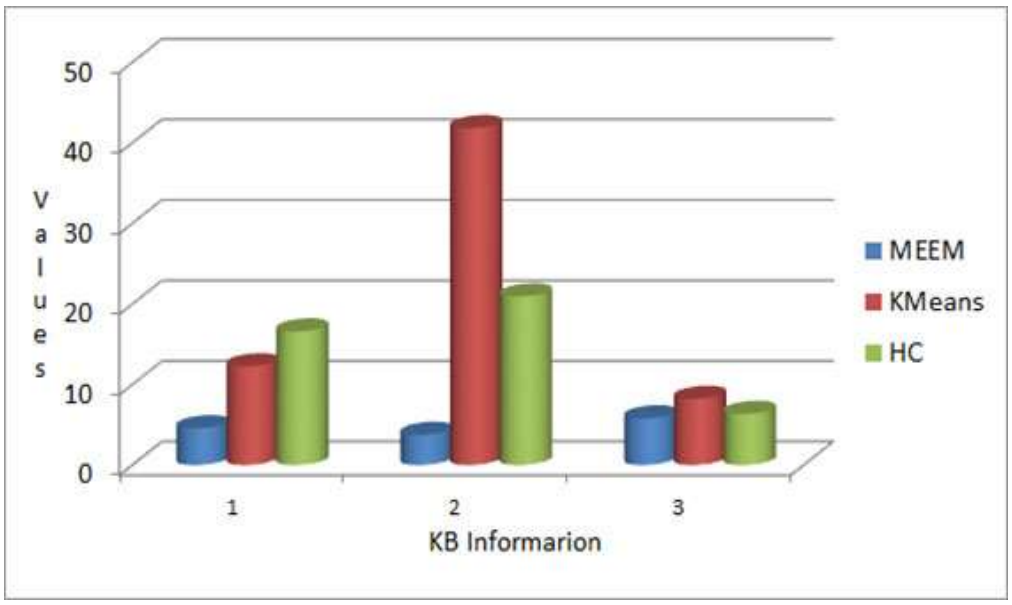

Fig. 7. KB Information of MEEM.

6. CONCLUSION

This research paper addressed the modified feature selection model (MEEM) for unstructured text document representation. This proposed MEEM model achieved the accuracy of 92.5 % when compared with K-Means and HC clustering algorithm. This proposed technique can be used in large data sets for feature extraction and efficient clustering. This model also proved that the computational time also reduced when compared with K-Means and HC Clustering methods. This MEEM model can be used for real-time applications like e-mail analysis, classification and clustering of malwares.

References

1. Aljaber, Bader, Nicola Stokes, James Bailey, and Jian Pei. "Document clustering of scientific texts using citation contexts." *Information Retrieval* (Springer) 13 (2010): 101–131.
2. Bouguila, Nizar. "Clustering of count data using generalized Dirichlet multinomial distributions." *IEEE Transactions on Knowledge and Data Engineering* (IEEE) 20 (2008): 462–474.
3. Bouguila, Nizar. "Count data modeling and classification using finite mixtures of distributions." *IEEE Transactions on Neural Networks* (IEEE) 22 (2010): 186–198.
4. Chen, Jiansheng, Zhengqin Li, and Bo Huang. "Linear spectral clustering superpixel." *IEEE Transactions on image processing* (IEEE) 26 (2017): 3317–3330.
5. Chen, Xiaojun, Xiaofei Xu, Joshua Zhexue Huang, and Yunming Ye. "TW-k-means: Automated two-level variable weighting clustering algorithm for multiview data." *IEEE Transactions on Knowledge and Data Engineering* (IEEE) 25 (2011): 932–944.
6. Consoli, Sergio, Kenneth Darby-Dowman, Gijs Geleijnse, Jan Korst, and Steffen Pauws. "Heuristic approaches for the quartet method of hierarchical clustering." *IEEE Transactions on Knowledge and Data Engineering* (IEEE) 22 (2009): 1428–1443.
7. Dai, Zhenwen, and Jörg Lücke. "Autonomous document cleaning—a generative approach to reconstruct strongly corrupted scanned texts." *IEEE transactions on pattern analysis and machine intelligence* (IEEE) 36 (2014): 1950–1962.
8. Fan, Wentao, Nizar Bouguila, and Djemel Ziou. "Unsupervised hybrid feature extraction selection for high-dimensional non-Gaussian data clustering with variational inference." *IEEE Transactions on Knowledge and Data Engineering* (IEEE) 25 (2012): 1670–1685.
9. Gu, Yu, Chunpeng Gao, Gao Cong, and Ge Yu. "Effective and efficient clustering methods for correlated probabilistic graphs." *IEEE Transactions on Knowledge and Data Engineering* (IEEE) 26 (2013): 1117–1130.
10. Guan, Renchu, Xiaohu Shi, Maurizio Marchese, Chen Yang, and Yanchun Liang. "Text clustering with seeds affinity propagation." *IEEE Transactions on Knowledge and Data Engineering* (IEEE) 23 (2010): 627–637.
11. Han, Longfei, Senlin Luo, Huaiqing Wang, Limin Pan, Xincheng Ma, and Tiemei Zhang. "An intelligible risk stratification model based on pairwise and size constrained Kmeans." *IEEE journal of biomedical and health informatics* (IEEE) 21 (2016): 1288–1296.

12. Hu, Xiaohua, E. K. Park, and Xiaodan Zhang. “Microarray gene cluster identification and annotation through cluster ensemble and EM-based informative textual summarization.” *IEEE Transactions on Information Technology in Biomedicine* (IEEE) 13 (2009): 832–840.
13. Huang, Jianbin, Heli Sun, Qinbao Song, Hongbo Deng, and Jiawei Han. “Revealing density-based clustering structure from the core-connected tree of a network.” *IEEE transactions on knowledge and data engineering* (IEEE) 25 (2012): 1876–1889.
14. Iam-On, Natthakan, Tossapon Boongeon, Simon Garrett, and Chris Price. “A link-based cluster ensemble approach for categorical data clustering.” *IEEE Transactions on knowledge and data engineering* (IEEE) 24 (2010): 413–425.
15. Janani, R., and S. Vijayarani. “Text document clustering using spectral clustering algorithm with particle swarm optimization.” *Expert Systems with Applications* (Elsevier) 134 (2019): 192–200.
16. Jiang, Jung-Yi, Ren-Jia Liou, and Shie-Jue Lee. “A fuzzy self-constructing feature clustering algorithm for text classification.” *IEEE transactions on knowledge and data engineering* (IEEE) 23 (2010): 335–349.
17. Kao, Ben, Sau Dan Lee, Foris K. F. Lee, David W. Cheung, and Wai-Shing Ho. “Clustering uncertain data using voronoi diagrams and r-tree index.” *IEEE Transactions on Knowledge and data engineering* (IEEE) 22 (2010): 1219–1233.
18. Lin, Yung-Shen, Jung-Yi Jiang, and Shie-Jue Lee. “A similarity measure for text classification and clustering.” *IEEE transactions on knowledge and data engineering* (IEEE) 26 (2013): 1575–1590.
19. Liu, James N. K., Yu-Lin He, Edward H. Y. Lim, and Xi-Zhao Wang. “A new method for knowledge and information management domain ontology graph model.” *IEEE Transactions on Systems, Man, and Cybernetics: Systems* (IEEE) 43 (2012): 115–127.
20. Papapetrou, Odysseas, Wolf Siberski, and Norbert Fuhr. “Decentralized probabilistic text clustering.” *IEEE Transactions on Knowledge and Data Engineering* (IEEE) 24 (2011): 1848–1861.
21. Park, Jinuk, Chanhee Park, Jeongwoo Kim, Minsoo Cho, and Sanghyun Park. “ADC: Advanced document clustering using contextualized representations.” *Expert Systems with Applications* (Elsevier) 137 (2019): 157–166.
22. Shabtai, Asaf, Lior Rokach, Yuval Elovici, and others. “OCCT: A one-class clustering tree for implementing one-to-many data linkage.” *IEEE Transactions on Knowledge and data Engineering* (IEEE) 26 (2013): 682–697.
23. Sim, Kelvin, Ghim-Eng Yap, David R. Hardoon, Vivekanand Gopalkrishnan, Gao Cong, and Suryani Lukman. “Centroid-based actionable 3D subspace clustering.” *IEEE transactions on knowledge and data engineering* (IEEE) 25 (2012): 1213–1226.
24. Singh, Jasmeet, and Vishal Gupta. “A novel unsupervised corpus-based stemming technique using lexicon and corpus statistics.” *Knowledge-Based Systems* (Elsevier) 180 (2019): 147–162.
25. Skabar, Andrew, and Khaled Abdalgader. “Clustering sentence-level text using a novel fuzzy relational clustering algorithm.” *IEEE transactions on knowledge and data engineering* (IEEE) 25 (2011): 62–75.
26. Song, Yangqiu, Shimei Pan, Shixia Liu, Furu Wei, Michelle X. Zhou, and Weihong Qian. “Constrained text coclustering with supervised and unsupervised constraints.” *IEEE Transactions on Knowledge and Data Engineering* (IEEE) 25 (2012): 1227–1239.
27. Wu, Junjie, Hongfu Liu, Hui Xiong, Jie Cao, and Jian Chen. “K-means-based consensus clustering: A unified view.” *IEEE transactions on knowledge and data engineering* (IEEE) 27 (2014): 155–169.
28. Xiong, Sicheng, Javad Azimi, and Xiaoli Z. Fern. “Active learning of constraints for semi-supervised clustering.” *IEEE Transactions on Knowledge and Data Engineering* (IEEE) 26 (2013): 43–54.
29. Xu, Jinglin, Junwei Han, Feiping Nie, and Xuelong Li. “Re-weighted discriminatively embedded k -means for multi-view clustering.” *IEEE Transactions on Image Processing* (IEEE) 26 (2017): 3016–3027.
30. Yan, Yang, Lihui Chen, and William-Chandra Tjhi. “Semi-supervised fuzzy co-clustering algorithm for document categorization.” *Knowledge and information systems* (Springer) 34 (2013): 55–74.