International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-2, July 2019

# Data Mining K-Means Document Clustering using TFIDF and Word Frequency Count

**Aranga Arivarasan, M. Karthikeyan,**

***Abstract:*** ***In the rapid development of www the amount of documents used increases in a rapid speed. This produces huge gigabyte of text document processing. For indexing as well as retrieving the required text document an efficient algorithms produce better performance by achieving good accuracy. The algorithms available in the field of data mining also provide a variety of new innovations regarding data mining. This increases the interest of the researchers to develop many essential models in the field of text data mining. In the proposed model is a two step text document clustering approach by K-Means algorithm. The first step includes Pre_Processing and second step includes clustering process. For Pre-Processing the method performs the tokenization approach. The distinct words are identified and the distinct words frequency of occurrence, TFIDF weights of the occurrences are calculated to form a document feature vector separately. In the clustering phase the feature vector is clustered by performing K-means algorithm by implementing various similarity measures.***

***Keywords: TFIDF, Word Frequency, Probability, Tokenization, Clustering***

## I. INTRODUCTION

The task of categorizing electronic document automatically in to their corresponding category is the main purpose of Document clustering. The fast increased internet usage leads to handling of enormous terabyte of electronic documents. Since the www efficient usage for several decades made text document classification very widespread as well as implementation in numerous application like web mail spam filtering, web user emotion analysis, customer commodity searching requirements etc. Text clustering is executed by representing the documents as a set of terms of indexes associated with some numerical weights. The goal is always to cluster the given text documents, in a way that they get clustered by means of the similarity measures with certain accuracy. There are many approaches are available for classification of text documents Naïve bayes, Support Vector Machines, DBSCAN, K-medoids, k-means and expectation maximization.

**Revised Manuscript Received on 30 July 2019.**
***** Correspondence Author
**Aranga Arivarasan*,** Assistant Professor, Department of Computer and Information Science, Annamalai University, India.
**Dr. M. Karthikeyan,** Assistant professor, Department of Computer and Information Science, Annamalai University, India.



The performances of the above said algorithms highly rely on the datasets provided to them for training. Before going to execute the text clustering the document representation approaches suffix tree representation of document analysis of similarity or distance metrics and most importantly the correct clustering approach are to be considered very carefully. In some cases Clustering is wrongly referred as automatic classification, because the clusters are formed without having the previous knowledge about distribution and the behavior of the data which are going to perform the clusters.

But the classes are always pre-defined as well the classification algorithm learns the relationship of target output with objects by learning from the training set. The training data set is nothing but a set of data which are correctly labeled to its target class by human influence, and the same is used to identify the learning activities of an unlabeled set of data. Several years were spent on study for carrying an efficient document clustering choosing a correct document features but still it is far from consideration in solving problems. The most promising challenges lie in selecting the exact features that the documents possess is identified and used for clustering. At the same time relevant similarity measures are to be identified to implement the clustering algorithm in an efficient way to make the clustering feasible and finding a way to associate the quality of clustering performance. To solve these issues there are several clustering techniques are available namely Distribution based methods**,** Centroid based methods**,** and Connectivity based methods Density Models and Subspace clustering The remaining portion of this work is organized as seven sections. The Section 2, referrers the related works regarding Document clustering is elaborated. The Section 3 describes the various distance metrics used in this paper. In Section 4, the proposed feature extraction procedure is elaborated briefly. The Section 5, evaluates experiment results in detail. The section 6 produces the conclusion of the paper and section 7 gives the references made.

## II. REVIEW OF RELATED LITERATURE

Chien-HsingChen [1] studied about the distance between the same terms which occur in the same article is given a weight. This weight is sensitive to the similar articles whether them term occurred and in different articles the term doesn't appear. They created a two-stage learning algorithm to identify the term weight and developed intelligent model by Appling the term weight to Reuter`s news articles by adapting classification and clustering operation. DamirKoren, JanSnajder [2]

introduce the notion of document-based topic coherence and propose novel topic coherence measures that estimate topic coherence based on topic documents rather than topic words. Hanan M.Alghamdi, AliSelamat [3] carried out an over view of existing Arabic webpage clustering method. Their goal is to solve the existing problems by examining the feature selection and reduction technique for clustering difficulties. The current research is the joint effort for improve feature selection and vectorization framework to enhance the application of current text analyses techniques to Arabic web pages. Ruksana, Yoojin Chung [4] approaches,

a crossover point may be selected even at a position inside a cluster centroid, which allows modifying some cluster centroids. This also guides the algorithm to get rid of the local minima, and find better solutions than the traditional approaches. Moreover, instead of running only one genetic algorithm as done in the traditional approaches, this article partitions the population and runs a genetic algorithm on each of them. This gives an opportunity to simultaneously run different parts of the algorithm on different virtual machines in cloud environments. Laxmi Lydia, E.Govindasamy, P,Lakshmanaprabu, S.K. [5] describes that the document clustering process based on the clustering techniques, partitioning clustering using K-means and also calculates the centroid similarity and cluster similarity. Monika Gupta, 2Kanwal Garg [6] provides an overview of the document clustering reviewed from different papers and the challenges in document clustering. Sunaina Kotekar Sowmya S.Kamath [7] propose a novel technique to cluster service documents into functionally similar service groups using the Cat Swarm Optimisation Algorithm. Anastasiya Kostkina, Denis Bodunkov, Valentin Klimov [8] investigate the impact of the approach on the quality of classification of documents and describe its application to the implementation of the document categorization. Mariam Thomas, Anisha, Resmipriya, M.G. [9] used the text clustering to generate the classification model for the next text classification step. When a new unlabeled text is incoming, measure its similarity with the centroids of the text clusters and give its label with that of the nearest text cluster. The similarity is calculated using different similarity measures. Jayaraj Jayabharathy ,Selvadurai Kanmani [10] proposes the following dynamic document clustering algorithms. 1.Term frequency based MAximum Resemblance Document Clustering (TMARDC) 2.Correlated Concept based MAximum Resemblance Document Clustering (CCMARDC) and 3.Correlated Concept based Fast Incremental Clustering Algorithm (CCFICA) are proposed. They compared the proposed algorithm with the existing static and dynamic document clustering algorithms through conducting experimental analysis with 20Newsgroups and scientific literature data set.

## III. SIMILARITY METRICS

In document clustering, similarity is typically computed using associations and commonalities among features, where features are typically words and phrases. Two documents are considered as similar if they share similar topics or information. When clustering is employed on documents, we are very much interested in clustering the component documents according to the type of information that is presented in the documents. Accurate clustering requires a precise definition of the closeness between a pair of objects, in terms of either the pair wise similarity or distance. A variety of similarity or distance measures have been proposed and widely applied, such as Spearman similarity, correlation similarity cosine similarity, Jaeeard coefficient, Euclidean distance and so on.

### *3.1 Spearman Similarity*

When documents are represented as term vectors, the similarity of two documents corresponds to the correlation between the vectors. Spearman Correlation measures the correlation between two sequences of values. The two sequences are ranked separately and the differences in rank are calculated at each position, *i*. The distance between sequences ***X*** = (***X1***, ***X2***, etc.) and ***Y*** = (***Y1***, ***Y2***, etc.) is computed using the following formula:

$$1 - \frac{6\sum_{i=1}^{n}(rank(X_i) - rank(Y_i))^2}{n(n^2 - 1)}$$

Where ***Xi*** and ***Yi*** are the *i*th values of sequences ***X*** and ***Y*** respectively. The range of Spearman Correlation is from -1 to 1. Spearman Correlation can detect certain linear and non-linear correlations.

### *3.2 Cosine Similarity*

The most commonly used measure in Document Clustering is the Cosine Similarity. For two documents $d_i$ and $d_j$, the similarity between them can be calculated

$$\cos(d_i, d_j) = \frac{d_i . d_j}{||d_i||\ ||d_j||}$$

where $d_i$, and $d_j$ are m-dimensional vectors over the term set T= $\{t_i, t2, ...t_m\}$. Each dimension represents a term -with its weight in the document, which is non negative. As a result, the cosine similarity is non-negative and bounded between [0, 1]. The cosine similarity is independent of document length. When the document vectors are of unit length, the above equation is simplified to:

$\cos(d_i, d_j) = di . d_j$

When the cosine value is 1 the two documents are identical, and 0 if there is nothing in common between them. Since, document vectors are orthogonal to each other.

### *3.3 Correlation Similarity*

Correlation is a technique for investigating the relationship between two quantitative, continuous variables. There are different forms of Pearson Correlation Coefficient formula. It is given by

$$(d_i, d_j) = \frac{m\sum_k d_{ik} - TF_i\, X\, TF_j}{\sqrt{[m\sum_k d_{ik}^2 - TF_i^2]\,[m\sum_k d_{jk}^2 - TF_j^2]}}$$

Where $TF_i = \sum_k d_{ik}$ and TF1 $= \sum_k d_{ik}$

The measure ranges from +1 to -1. Positive correlation indicates that both variables increase or decrease together, where as negative correlation indicates that as one variable increases, so the other decreases, and vice versa. Two documents are identical when Pearson similarity is ±1.

The spearman distance is a distance measure, while the cosine similarity and Pearson coefficient are similarity measures. We apply a simple transformation to convert the similarity measure to distance values. Because cosine similarity bounded in [0, 1] and monotonic, we take D = 1-SIM as the corresponding distance value. For Pearson coefficient, which ranges from -1 to +1, we take D = 1 - SIM when SIM >0 and D = | SIM | when SIM <0.

## IV. PROPOSED FEATURES EXTRACTION METHODS

### *4.1 Preprocessing*

The clustering process is achieved through variety of preprocessing techniques to produce good accuracy and reliable performance. We elaborated here a variety of preprocessing techniques which are commonly implemented in the process of document clustering. The main objective of preprocessing is to use the training documents in a form of feature vector that can be further utilized for clustering. Some of the ways of representing the documents are, Vector-Model, TFIDF, Probability, graphical model, keyword word count etc. weighing of the documents and their similarities are measured by implementing various techniques. The priority of a word which appears in a given document is usually represented through a associated numerical value which is known as a vector representation. The text mining process highly relies on set of words a bag-of-words that a text document can be efficiently represented. The text processing phase involves, after reading textual documents divides text document characteristic into tokens, words, terms, or attributes. The weight obtained from the frequency of occurrence of important terms in each text document, following removal of attributes which does not have any information, like stop words, numeric digits, and symbols. Despite removing non-informative features, the size of a text document space may be too large. To reduce the large size of the document corpus several operations are performed in the preprocessing. The purpose of this phase is to improve the quality of features extracted to represent the document by reducing the complexity of the text mining operation to improve the quality of features extracted to represent the document. The effective preprocessing leads to achieve better clustering results.

### *4.2 Tokenization*

Tokenization is performed to divides the document into tokens and also to group the individual tokens to develop higher levels interpretations. Tokenization converts a stream of characters into a sequence of tokens. A token is an occurrence of a sequence of particular characters which appear in an document which represent the same class that are grouped together for processing as a meaning full semantic unit. A type is the class of all tokens containing the same character sequence. In the tokenization process first the document is divided in to individual words by identifying the white space character and the next line character. Then to enable the cleaning and filtering the empty sequence of characters are removed. The final output of the process will be only words or terms which are known as the tokens. The tokens are considered as attributes of the text document.

### *4.3 Stop Words*

Commonly repeated tokens which present in each and every text document is identified as stop words. The connecting words and pronouns known as stop words need to be removed due to it does not have any effect in the form of tokens and these words does not have any meaning or does not add any value towards the categorization process of a document representation. The stop words appear in larger amount will increase the number of tokes without having any important in the clustering process. This will increase the processing time and will also affect the accuracy of prediction. The special characters and the numeric characters are to be removed from the tokens because of their non importance characteristics towards the matching of the documents in the user point of view. A list of stop words can be created by identifying the list of words which often appear in a document without having any semantic value to that particular document and having high percentage of occurrence.

### *4.4 Stemming*

Some tokens in the document will have some characters before or after the words. Those prefix and suffix characters removal from a word is known as stemming. In cases the root words can be elaborated using the prefix and suffix word for achieving the grammar of the language. For example from the word ‘going’ we can stem the token in to ‘go’. The process is carried out for reducing the number words to their stem without affecting the context of the text where ‘go’ and ‘going’ is derived from same token but in the corpus they will be identified as different tokens without the stemming operation. The stemming process is carried using the following algorithm

Step 1: Eliminate plurals (-s) and suffixes (-ed, -ly, -ness,-full or -ing).

Step 2: If the vowel occurs in previous step, replace y to i on the next word.

Step 3: From the step 3, Map double suffixes to single ones (-elation,, -ation,-ational).

Step 4: Deducts (-ant, - out , -ence,) etc.

Step 5: If some word ends with a grammatical verb ending, then it has been removed.

Step 6: Finally, removes a (-e).

### *4.5 Word Frequency Count*

An important set of metrics in text mining relates to the frequency of word count (or any token) in a certain corpus of text documents.

However, one can also use an additional set of metrics in cases where each document has an associated numeric value describing a certain attribute of the document. One will first go through the process of creating a simple function that calculates and compares the absolute and weighted occurrence of words in a corpus of documents. This can sometimes uncover hidden trends and aggregates that aren't necessarily clear by looking at the top ten or so values.

They can often be different from the absolute word frequency as well. Then It is simple to do the basic analysis and find out that your words are split 50:50 to measure the absolute frequency of words, and try to infer certain relationships. In this case, you have some data about each of the documents. The key word exists: in which case the assignment is done (adding one). Now the key exists, its value is zero, and it is ready to get assigned an additional 1 to its value.

Although the top word was in the first table, after counting all the words within each document we can see that other words are tied for the first position. This is important in uncovering hidden trends, especially when the list of documents you are dealing with, is in the tens, or hundreds, of thousands. With counted occurrences of each word in the corpus of documents, the weighted frequency can be obtained. This reflects how many times the words appeared to readers; compared to how many times used them.

### *4.6 Bag of Words (bow)*

BOW is simplified version of the corpus which is used in data mining for information retrieval and document clustering. Bag of Word is a simplest method for feature identification and representation of text document. BOW process consists of the following steps,

*Step 1:* All documents in corpus is indexed with the bag full of terms, by vector with a document for each term occurring throughout the whole collection of tokens in document. Each vector has a corresponding value representing the number of occurrences of the term in the document.

*2:* All documents are represented as a point in a vector space with one dimension for every term appears in the vocabulary.

*3:* If a particular word does not appear in a document, that particular word is set a vector value zero.

### *4.7 Tf--Idf*

Feature selection is an essential process in document clustering to produce better accuracy, efficiency, and scalability of a text documents, compared to other techniques. Several procedures are available to group the text documents namely information gain, mutual information, term Frequency, Chi-square process, cross entropy, the term weighting methods of text, index based process. Among these methods, Information Gain, TF, DF and IDF, Chi-square (Statistical term and entropy based), term weighting methods *TSW* and TDW are useful methods to manage the feature selection process. The Enhanced TF-IDF is used for dimensionality reduction. The feature selection and weighting methods contain the following steps. The term weights are set as the simple frequency counts of the terms appear in the documents. This enable the ability of understanding that tokens occurring frequently within a document may reflect its meaning more strongly, than terms occurring less frequently and should be given higher weights. In term-space document d is considered as a vector and represented by the term frequency (TF) vector as a numeric value.

The document vector "d" is represented by,

D= {Term X $freq_1$Term X $Freq_2$ … … Term X $Freq_n$}

Where i= {1, 2 . . . , n} is the term frequency for whole documents. Depending on the Vector Space Model, the weight matrix is calculated by using the matrix derivation.

**Term Frequency (TF):** is a scoring of the frequency of the word in the current document. Since every document is different in length, it is possible that a term would appear much more times in long documents than shorter ones. The term frequency is often divided by the document length to normalize.

$$TF(t) = \frac{Number\ of\ times\ term\ t\ appears\ in\ document}{Total\ number\ of\ terms\ in\ document}$$

To compute a higher weight to the words that occurs in some documents the words which occur frequently across the entire collection are not helpful. Terms that are occur in few documents are very much useful in differentiating the documents from the rest of the collection. In the inverse document frequency term weighting the higher weights is assigned to these more discriminative words. IDF is defined by fraction N/ni,

where, N- is the total number of Documents in the collection and

ni -is the number of documents in which term i occurs.

Because of the large number of document collections, this measure is usually compressed with a log function. The resulting definition IDF is thus:

$$idf = \log\left(\frac{N}{n_i}\right)$$

Term frequency and IDF results are combined to produce TF-IDF weighting

$$w_{i,j} = tf_{i,j}\ X\ idf_1$$

The TF-IDF depiction with Document d is

$$d_{tf-idf} = \left[tf_1 \log\left(\frac{n}{df_1}\right), tf_2 \log\left(\frac{n}{df_2}\right), ,, ..., tf_D \log\left(\frac{n}{df_D}\right),\right]$$

Normalized unit vector to all document vectors is

$$\left|\left|d_{tf-idf}\right|\right| = 1$$

Centroid vector $c_j$ is

$$c_j = \frac{1}{|c_j|} \sum_{d_j \in c_j} d_i$$

**Inverse Document Frequency (IDF)**: IDF scoring is done by considering how rare the word is across the documents. If the term occurs vary rarer then it is given more IDF score.

$$IDF(t) = log_e\left(\frac{Total\ number\ of\ documents}{Number\ of\ documents\ with\ term\ t\ in\ it}\right)$$

The Term Frequency (TF) of all the words appear in the document is normalized through the inverse document frequency that helps to find TF/DF of the documents. TF/DF describes coordinates of the term weights that are given by the term frequencies as well as to calculate Cosine similarity for all vector space models.

The cosine similarity equation is as

$$CoSim(Q, D_i) = \frac{\sum_i w_{q,j} w_{i,j}}{\sqrt{\sum_j w_{Q,j}^2} \cdot \sqrt{\sum_i w_{i,j}^2}}$$

Q – Query of frequent term
I - IDF
W – Weight
J – TF
D – Document vector
Thus,

$$TF - IDF\ score = TF * IDF$$

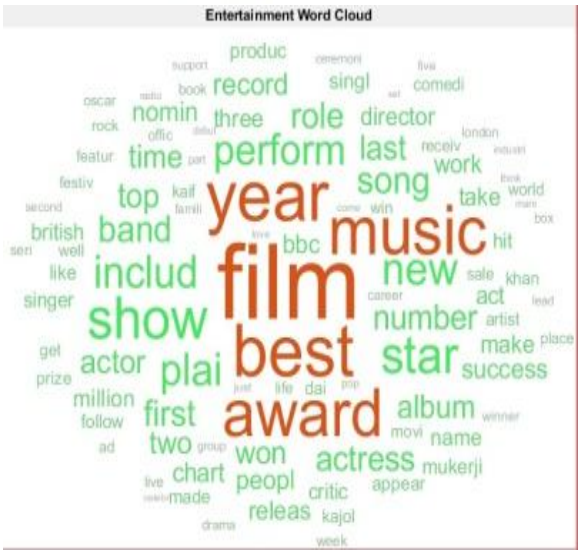
## V. EXPERIMENTS AND RESULTS

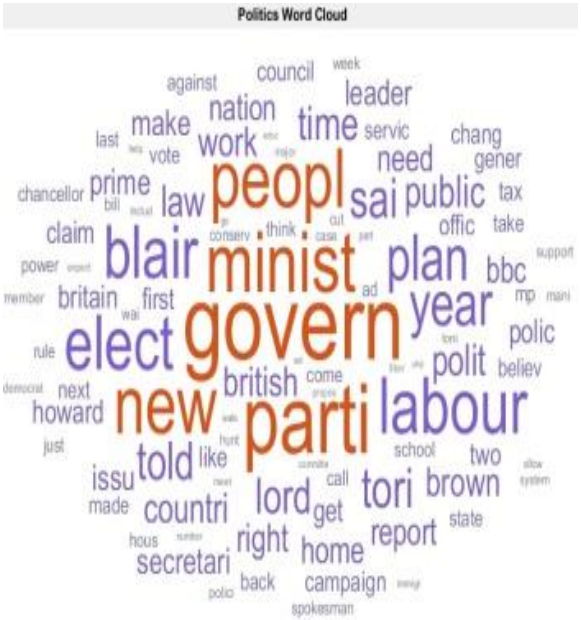
For our experimental purpose the proposed system collected 300 documents for the five categories Business, Entertainment, Politics, Sports and Technology. First the proposed algorithm splits each document in the individual tokens. The all the tokens are used for calculating the key word occurrences and the corresponding findings are listed in Table1.

**Table.1. Word count and unique words**

| Category | Total Number Of Unique Words | Total Number of words |
|---|---|---|
| Business | 6544 | 60786 |
| Entertainment | 8929 | 64622 |
| Politics | 7422 | 76277 |
| Sports | 6919 | 59930 |
| Technology | 8207 | 87550 |

The fig.1 shows the word cloud of each category of documents. This consists the total keywords that are occurred frequently are differentiated with the colour.

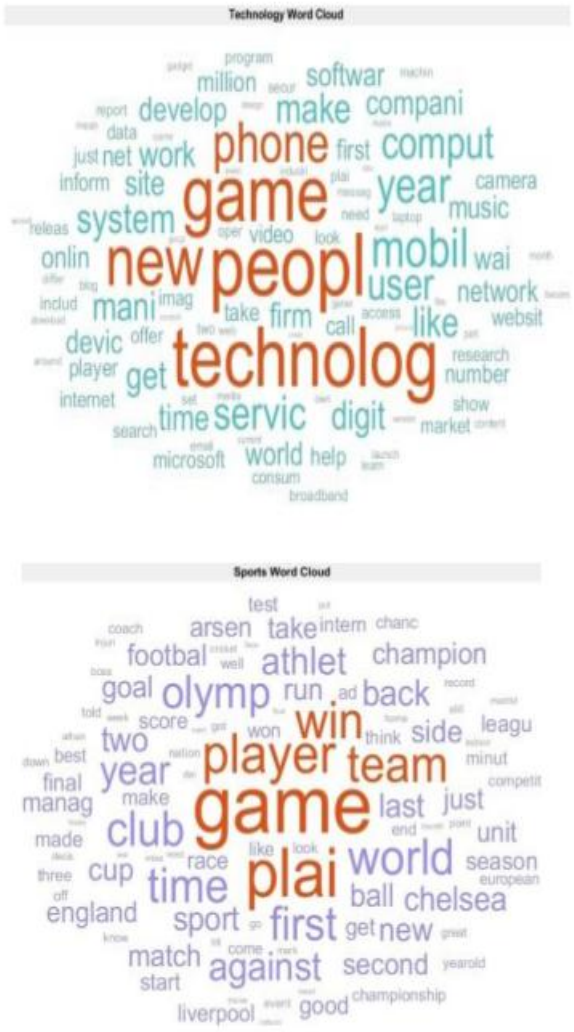



**Fig.1. Word Cloud**

-

The TF-IDF frequency and the number of word occurrences is calculated the values are used to cluster with K-Means algorithm with different distance measures. Hear the three distance metrics Spearman, Cosine Similarity and the correlation similarity are performed to evaluate the clustering process. With the clustering results the confusion matrix is determined. From the confusion matrix the precision recall accuracy and F-measures are calculated those results are listed in table 2. precision, Recall and F-measures are computed through the following formula.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$Recall = \frac{True\ positive}{True\ Positive + False\ Negative}$$

$$F1 = 2\ X\ \frac{Precision * Recall}{Precision + Recall}$$

**Table.2. Performance evaluation of Similarity metrics**

| Distance Measures | WordCount | | | | TF-IDF | | | |
|---|---|---|---|---|---|---|---|---|
| | Accuracy | Precision | Recall | F-Measures | Accuracy | Precession | Recall | F-measures |
| Spearman Similarity | 94.46 | 9.20 | 9.23 | 9.21 | 94.80 | 9.37 | 9.43 | 9.40 |
| Cosine Similarity | 76.87 | 7.81 | 7.50 | 7.65 | 80.80 | 8.50 | 8.53 | 8.19 |
| Correlation Similarity | 72.33 | 6.44 | 6.76 | 6.60 | 74.40 | 6.59 | 6.76 | 6.67 |

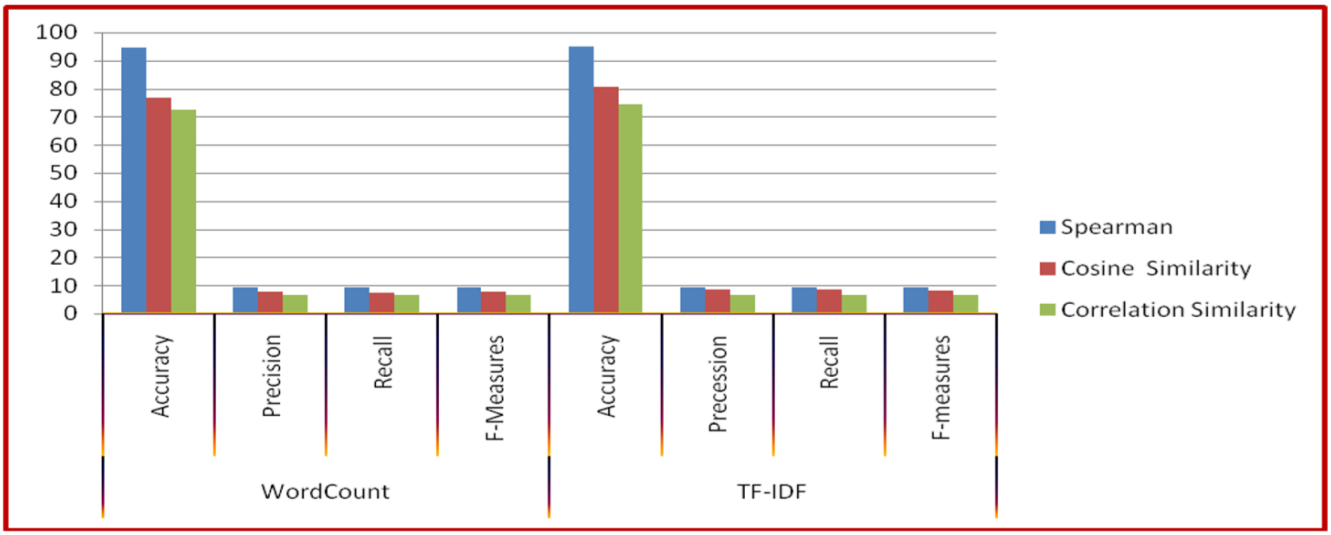



**Fig.2. Graphical representation of similarity metric performance.**

For our better understanding and demonstration the retrieved results were shown in fig.2. The results shown that the Spearman Similarity achieves an overall accuracy of 94.46 for Word Count and 94.80 for TF-IDF. The Cosine Similarity achieves and overall accuracy of 76.87 for word count and 80.80 for TF-IDF. The Correlation Similarity achieves and overall accuracy of 72.33 for Word Count and 74.40 for TF-IDF. From the results we cal clearly identify that the Spearman Similarity measures achieves the best overall performance than the other two similarity measures.

## VI. CONCLUSION

The proposed data mining clustering approach is performed with the WordCount and TF-IDF features of the documents of five different categories. For each category we have taken 300 documents. The proposed system retrieves 6544 key words out of 60786 unique words for the business category. The Entertainment category documents retrieve 8929 key words out of 64622 unique words. The Politics category retrieves 7422 key words out of 59930 unique words. The sports category retrieves 6919 key words out of 59930 unique words. The Technology category retrieves 8207 key words out of 87550 unique words. The proposed model uses three similarity measures to compute the clustering operation. The Spearman Similarity measure yields an accuracy of 94.46 and 94.80 for WordCount and TF-IDF respectively. Cosine Similarity measure yields an accuracy of 76.87 and 80.80 for WordCount and TF-IDF respectively. Correlation Similarity measure yields an accuracy of 72.33 and 74.40 for WordCount and TF-IDF respectively. In future the model may be extended by increasing the number of categories. The features used hearer can also be examined by other clustering approaches like DBSCAN, KNN, and GMM so on.

## REFERENCES

1. Chien-HsingChen Improved TFIDF in big news retrieval: An empirical study, 93(2017) 113-122.
2. DamirKoren, JanSnajder Document-based topic coherence measures for news media text 114(2018) 357-373
3. Hanan M.Alghamdi, AliSelamat, Arabic Web page clustering: A review 31(1)(2019) 1-14.
4. Ruksana, Yoojin Chung An Improved Genetic Algorithm for Document Clustering on the Cloud 8(4) (2018) 9-19.
5. Laxmi Lydia, E.Govindasamy, P,Lakshmanaprabu, S.K. Document Clustering Based On Text Mining K-Means Algorithm Using Euclidean Distance Similarity, 10(2018) 208-214
6. Monika Gupta, 2Kanwal Garg , A Review on Document Clustering, International Journal of Advanced Research in Computer Science and Software Engineering, Volume6,Issue 5, May 2016.
7. Sunaina Kotekar Sowmya S.Kamath Enhancing service discovery using cat swarm optimisation based web service clustering, 8(2016) 715-717.
8. Anastasiya Kostkina, Denis Bodunkov, Valentin Klimov , Document Categorization Based on Usage of Features Reduction with Synonyms Clustering in Weak Semantic map, 145 (2018) 288-292.
9. Mariam Thomas, Anisha, Resmipriya, An Efficient Text Classification Scheme Using Clustering, 24(2016) 1220-1225.
10. Jayaraj Jayabharathy ,Selvadurai Kanmani , Correlated concept based dynamic document clustering algorithms for newsgroups and scientific literature 1:3 (2014) 1-21.

## AUTHORS PROFILE

**Aranga Arivarasan** is a Research Scholar who is working as Assistant Professor in Division of Computer and Information Science, Annamalai University, India. He completed his B.Sc[Computer Science] and M.Sc [Computer Science] From Madras university in 1998 and 2000 respectively, the M.B.A and M.Phil [Computer Science] from Annamalai University in 2005 and 2007 respectively.

**Dr. M. Karthikeyan** is an Assistant professor in Division of Computer and Information Science, Annamalai University, India. He completed his M.Sc[Computer Science] from Bharather University in 1993 and M.Phil [Computer Science] and Phd from Annamalai University in 2005 and 2014 respectively. His area of interest is Data Mining, Digital Image Processing, and Artificial Neural Networks.