ISSN 2278-3091

Volume 8, No.5, September - October 2019

International Journal of Advanced Trends in Computer Science and Engineering

Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse21852019.pdf

https://doi.org/10.30534/ijatcse/2019/21852019

Performance Assessment of Various Text Document Features through K-Means Document Clustering Approach

Dr.M.Karthikeyan[1], Aranga Arivarasan[2], D.Kumaresan[3]

[1] Department of Computer and Information Science, Annamalai University,Annamalai Nagar, Tamilnadu, India,
mkshkarthik@yahoo.co.in.

[2]Department of Computer and Information Science, Annamalai University,Annamalai Nagar, Tamilnadu, India,
profarivarasan@yahoo.com.

[3]Department of Computer and Information Science, Annamalai University,Annamalai Nagar, Tamilnadu, India,
aucsedks@yahoo.co.in.

ABSTRACT

The text documents are very important in the usage of www. Many users require so much text document to gather the information in their required field of interest. To serve the internet surfers the appropriate required topic documents are to be retrieved. For this purpose for indexing and retrieving the text document the researchers tend to produce many algorithms in the field of text document mining. The entire effort of clustering is achieved relying on the selection of appropriate similarity metrics. The document clustering is performed by two steps which begin with feature extraction prior to clustering operation. The features from the text document are extracted through various operations like preprocessing, tokenization, Stop word removal, streaming and bag of Words were performed. By performing the previous mentioned operations successively the Document representing features namely WordCount, TF_IDF and probability of words were determined to perform the next process with clustering algorithm. In the clustering phase the three features and some of the similarity measures were used to perform the clustering operation. The proposed method yields better results for Probability based features compared with other two TFIDF and WordCount.

Key words : TFIDF, Word Frequency, Probability, pre-processing, Clustering, K-Means.

1. 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 wide spread as well as implementation in numerous application like web mail spam filtering, web user emotion analysis, customer commodity searching requirements etc.

Document through set of terms with indexes associated with some numerical weights. The idea behind the clustering of given text documents, in a way that it get clustered by means of the similarity of previous clustered documents 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. 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.

One of the wide spread weighting done in the field of document handling is tf-idf weight. It represents the Term Frequency-Inverse Document Frequency. This is used hear as a statistical procedures to determine the importance of a particular word to identify the type of that document in the bulk compilation of data corpus. The tf is mainly focusing of the frequency of presence of a particular term in that document. The TF weight is determined by the value obtained through dividing the particular words number of occurrences with the total amount of words present in that document. The IDF determine the significance of the particular term which present in the document. It is computed by measuring the log value of the total count of documents present in the corpus divided through the total count of documents which possess the particular term in that document.

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.

2. LITERATURE REVIEW

There are several clustering algorithm and similarity metrics for document clustering and indexing. The clustering is one of the main mechanisms to extract useful information from documents as unstructured set of objects from a group by identifying the similarity among them. Because of its easy to use nature and effectiveness the K-Means algorithm gets the first priority to perform the clustering operation. Saqib Alam et al,[1] have used the text mining techniques to analyze the old and modern English languages and have introduced the Common-Words Counting algorithm that identifies common words of 15th century that diminishes gradually in the later centuries as well as computed the speed of linguistic changes and identified the reasons behind them. Vladimer B. Kobayashi1 et al.[2] aims to acquaint organizational researchers with the fundamental logic underpinning text mining, the analytical stages involved, and contemporary techniques that may be used to achieve different types of objectives. The specific analytical techniques reviewed are (a) dimensionality reduction, (b) distance and similarity computing, (c) clustering, (d) topic modeling, and (e) classification. Kasula Chaithanya Pramodh et al.[5] present a novel approach to extract the concept from a document and cluster such set of documents depending on the concept extracted from each of them by transforming the corpus into vector space by using term frequency–inverse document frequency then calculated the cosine distance between each document, followed by clustering them using K means algorithm and used multidimensional scaling to reduce the dimensionality within the corpus. Yuqiang Tong et al. [13] presents a new clustering algorithm, this algorithm expresses the news text as a series of Text labels, which effectively solves the problem that the data latitude is too high, and the clusters is too hard to express. At the same time, by using a conceptual clustering algorithm, this method effectively reduces the number of comparisons. Marzieh Oghbaie et al. [14] introduce a novel text document similarity measure based on the term weights and the number of terms appeared in at least one of the two documents. The effectiveness of the measure is evaluated on two real-world document collections for a variety of text mining tasks, such as text document classification, clustering, and near-duplicates detection. Marmar MoussaIon et al. [15] present novel computational approaches for clustering scRNA-seq data based on the Term Frequency - Inverse Document Frequency (TF-IDF) transformation that has been successfully used in the field of text analysis. Yehang Zhu et al.[16] propose a novel clustering algorithm CARDBK—“centroid all rank distance (CARD)” which means that all centroids are sorted by distance value from one point and “BK” are the initials of “batch K-means”—in which one point not only modifies a cluster centroid nearest to this point but also modifies multiple clusters centroids adjacent to this point, and the degree of influence of a point on a cluster centroid depends on the distance value between this point and the other nearer cluster centroids.

3. 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 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= $\{t_i, t\mathbf{2}, \ldots t_m\}$. 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:

$\cos(d_i, d_j) = \mathrm{di}.\ d_j$

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 Pearson 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.

4. METHODOLOGY

There are many ways to represent a text document. First it can be represented as a bag of words, where all the words are assumed to appear independently and the order is not given any priority. One of the wide spread document representation model is the Bag of Words Model in information retrieval and text mining. Words are counted in the bag as individual tokens. Each word corresponds to a dimension in the resulting data space and each document then becomes a vector consisting of non-negative values on each dimension. Here we use the occurrence of each term as its weight, in a result of terms that appear more frequently are more important and descriptive for the document. Even though the most frequent terms of a document are very important at the same time some of the words like a, and, the etc are not have much priority to represent the document type. So those should be removed from the Bag of Words. For this purpose the documents in the corpus are meant to be undergoing some sequence of processing.

4.1 Preprocessing

The clustering methods depend on various preprocessing techniques to achieve optimal quality and performance. We discuss here some of the common preprocessing methods. The main objective of preprocessing is to represent the data in a form that can be utilized for clustering. Several ways of representing the documents are, Vector-Model, graphical model, TFIDF, Probability, keyword word count etc. weighing of the documents and their similarities are measured by implementing various techniques. The importance of a word within a given document is usually represented in a vector representation where for each word a numerical value is stored. The text mining approaches 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 the terms of each text document, followed by the removal of non-informative attributes, such as stop words, numbers, and special characters. The remaining characteristics are then standardized by reducing to the root during the error correction process. Despite removing non-informative features, the size of a text document space may be too large. Certain constraints to reduce the size of the character space of each document input text in addition with the frequency of the characteristic of each document. The purpose of this phase is to improve the quality of features extracted to represent the document and at the same time reducing the complexity of the mining process.

4.2 Tokenization

Tokenization, in our sense, not only divides the tokens into processing but also interprets and groups of individual tokens to create higher levels interpretations. Tokenization converts a stream of characters into a sequence of tokens. A token is an instance of a sequence of characters in some particular document that are grouped together as a useful semantic unit for processing. A type is the class of all tokens containing the same character sequence. The data must be processed in all three operations: the first operation is to convert the documents into the number of words equal to BOW. The second operation is to remove an empty sequence, that is, this step involves cleaning and filtering. Finally, each text input document is divided into a list of characteristics, also called tokens, words, terms, or attributes.

4.3 Stop words

Stop words list is a list of commonly repeated tokens which emerge in every text document. The common tokens such as conjunctions and pronouns need to be removed due to it does not have any effect in the form of tokens and these words add a very little or no value on the categorization process of a document representation. Some extremely common words that would appear to be of little value in helping documents matching a user need are excluded from the vocabulary entirely. For the same reason, if the tokens are a special character or a number then those tokens should be removed. In order to find the stop words, we can arrange our list of terms by frequency and pick the high frequent ones according to their lack of semantics value.

4.4 Stemming

Stemming is the process of removing prefixes and suffixes from tokens. the process is carried out for reducing the derived words to their stem. The stem need not to be identified to the original morphological root of the word and it is usually sufficiently related through words map to the similar stem. This process is used to reduce the number of tokens in the feature space and improve the performance of the clustering when the different forms of features are stemmed into a single feature. The streaming process is carried using the following algorithm

Step 1: Eliminate plurals (-s) and suffixes (-ed or -ing).

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

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

Step 4: Additionally, reduces the suffixes like (-full, -ness) etc.

Step 5: Deducts (-ant, -ence) etc.

Step 6: If a word ends with a grammatical verb ending, then it has been removed.

Step 7: 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 a simplified representation 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: Each document is indexed with the bag of the terms, by a vector with one document for each term occurring in the whole collection of tokens in document . Each vector has a corresponding value representing the number of times the term occurred in the document.

Step 2: All document represented as a point in a vector space with one dimension for every term in the vocabulary.

Step 3: If a word does not appear in a feature document, that particular vector is set to the 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:

In Term Frequency and Term Frequency Inverse Document Frequency the term weights are set as the simple frequency counts of the terms in the documents. This reflects the ability of understanding that terms occurring frequently within a document may reflect its meaning more strongly than terms occurring less frequently and should be given higher weights. Each document d is considered as a vector in the term-space and represented by the term frequency (TF) vector.

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 give a higher weight to words that occurs only in a few documents the words that occur frequently across the entire collection are not helpful. Terms that are limited to few documents are useful for differentiating those documents from the rest of the collection. The inverse document frequency term weight is one way of assigning higher weights to these more discriminative words. IDF is defined by fraction N/ni, where, N is the total number of documents in the collection and the number of documents in which term i occurs. Due to the large number of documents in many collections, this measure is usually squashed with a log function. The resulting definition IDF is thus:

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

Combining the term frequency with IDF results are defined as TF-IDF weighting

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

The TF-IDF representation of the Document d is

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

Normalized unit vector to all document vector is

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

Centroid vector c_j is

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

Inverse Document Frequency (IDF): is a scoring of how rare the word is across documents. IDF is a measure of how rare a term is. Rarer the term, more is the IDF score.

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

The Term Frequency (TF) for each word is normalized by 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 term frequency
I - Inverse document frequency
W – Weight
J – Term frequency
D – Document vector
Thus,

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

From the above formula If a document containing 100 words wherein the word *India* appears 5 times. The term frequency for *India* is (5/ 100) = 0.05. Now, assume we have 10 million documents and the word *India* appears in one thousand of these. Then,t he inverse document frequency is calculated as log(10,000,000 / 1,000) = 4. Now, the Tf-idf weight is the product of these determined values: 0.05 * 4 = 0.20.

4.8 Probability

The important contribution in this proposed method is to find the probability distribution of similar documents to perform the clustering process. The proposed method determines a unique probability distribution equation to achieve the most significant accurate clustering process. In the document clustering phase each document is selected from the dataset and by using the probability distribution function the corresponding probability of each unique word in that document is calculated for the purpose of clustering. For each word in the document the relationship between the selected document and the corresponding cluster is determined. Depending on the probability values the document which has the overall maximum probability value is assigned to that cluster.

$$P(D_i, C_j) = \sum_{i=1}^{k} P(D_{wt}) * P(w_i | C_j)$$

$$where, P(D_{wt}) = \frac{no\text{:}\ of\ wt\ count\ in\ selected\ Doc}{Total\ word\ count\ in\ selected\ Doc}$$

$$P(w_i, C_j) = \frac{No\ of\ wt\ count\ in\ cluster}{Total\ word\ count\ in\ cluster}$$

5. EXPERIMENTAL RESULTS

For our experimental purpose the proposed system collected 300 documents for the five categories Business, Entertainment, Politics, Sports and Technology. It is very important to perform the very effective preprocessing task to generate the unique words. The outcome of our proposed system to generate the unique words was shown in Table1

Table 1: Total word and unique word count

Category	Total Number Of Unique Words	Total Number of words
Business	16544	160786
Entertainment	18929	164622
Politics	17422	176277
Sports	16919	159930
Technology	18207	187550

Initially the proposed system splits the entire documents in the corpus in to individual tokens. Then all tokens are calculated to find the keyword occurrence. With table 1. results the WordCount, TF_IDF frequency and probability of keyword occurrence were calculated. The determined values are formed in to clusters by means of K-Means clustering algorithm. To perform the clustering the K-means uses three Similarity metrics Spearman Similarity, Cosine Similarity and the Correlation Similarity. From the clusters the confusion matrix is determined. The confusion matrix is used to find the accuracy, precision, recall and F-measures.

The precision, Recall and F-measures were calculated by using the following formulas

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

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

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

The calculated values are shown in Table 2. The fig.1. Graphically represent the accuracy of the three similarity metrics for the WordCount, TF_IDF and the Probability values. It clearly shows that the clustering operation gives better results by using the

Probability values for all the three similarity metrics. Among the similarity metrics the Spearman similarity gives better results than the other two Cosine Similarity and Correlation Similarity for all three WordCount, TF_IDF and Probability value. Figure 1 Shows the word cloud chart of the five categories Business, Entertainment, Politics Sports and Technology.

Table 2: Results using wordcount tf-idf and probability of words

	Distance Measures	Spearman Similarity	Cosine Similarity	Correlation Similarity
WordCount	Accuracy	94.46	76.87	72.33
	Precision	9.20	7.81	6.44
	Recall	9.23	7.50	6.76
	F-Measures	9.21	7.65	6.60
TF_IDF	Accuracy	94.80	80.80	74.40
	Precision	9.37	8.50	6.59
	Recall	9.43	8.53	6.76
	F-Measures	9.40	8.19	6.67
Probability	Accuracy	95.7	88.4	86.8
	Precision	9.45	8.84	8.77
	Recall	9.33	8.66	8.33
	F-Measures	9.39	8.75	8.54

The word cloud is an image consist of words appeared in a selected document, in that image the occurrence of each word is indicated as its frequency and that is given the priority to perform the clustering. The tag cloud is a new technique of representing the text data as to visualize keywords and metadata in a document corpus. Tags are always single words, and the importance of each tag is shown with different font size and color

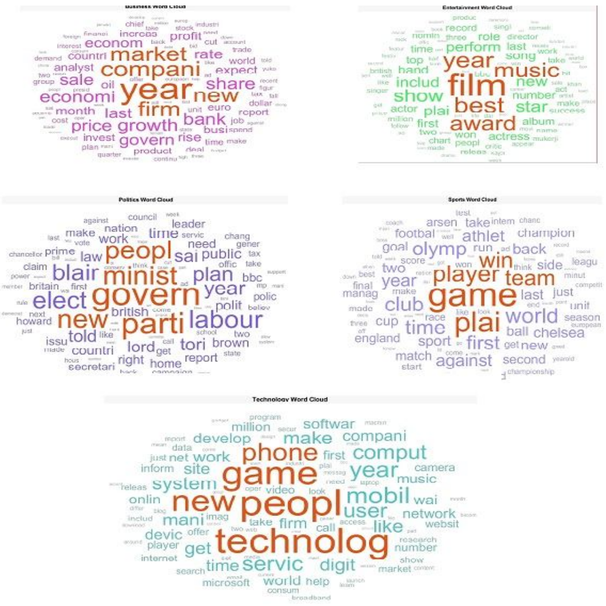


Figure 1: Word Cloud of Business, Entertainment, Politics Sports and Technology

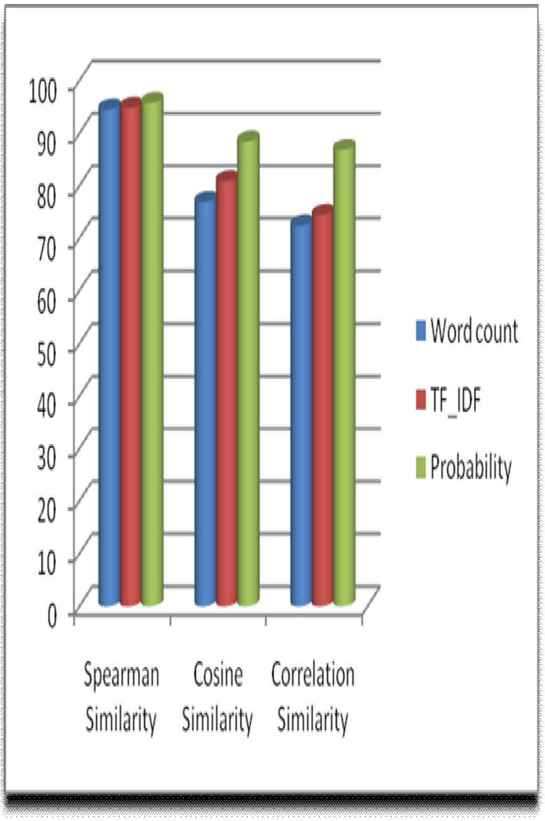


Figure 2: Bar chart showing the Accuracy

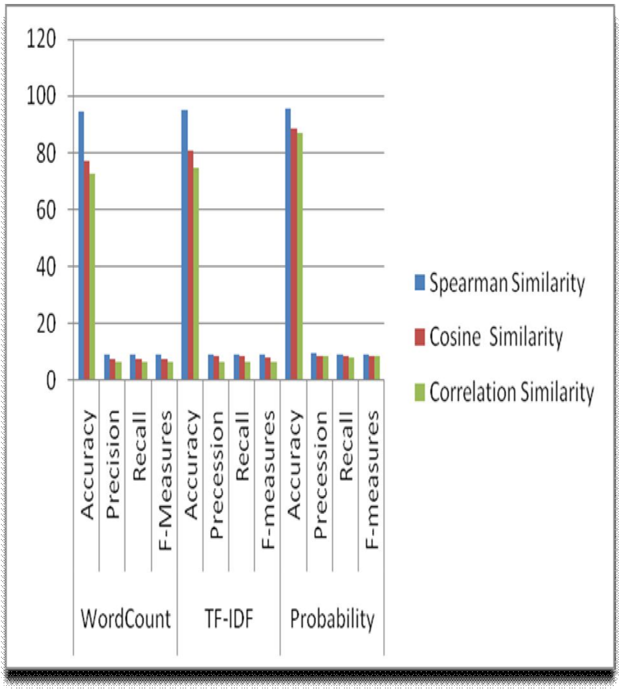
For our better understanding and demonstration purpose all the calculated results were shown in Figure 2. Our proposed system calculates Accuracy, Precision, Recall and F-Measures by using the WordCount, TF_IDF and Probability values through K-Means clustering. The clusters are determined by using the three similarity metrics Spearman Similarity, Cosine Similarity and the Correlation similarity.

From the Figure 3. we can clearly understand that the Spearman Similarity measures achieves the best overall performance than the other two similarity measures.

Figure 4 describes the line chart of Precision, Recall and F-Measures determined by our proposed system. Precision, Recall and F-Measure are the external measures to analysis the performance of the system. Precision retrieves the number of correct assignments out of the number of total assignments made by the system.

Recall retrieves the number of correct assignments made by the system, out of the number of all possible assignments. F-measure is a combination of the precision and recall measures used in system. From Figure 4 shows our system yields better performance.

Figure 3: Accuracy, Precision, Re call and F-Measures



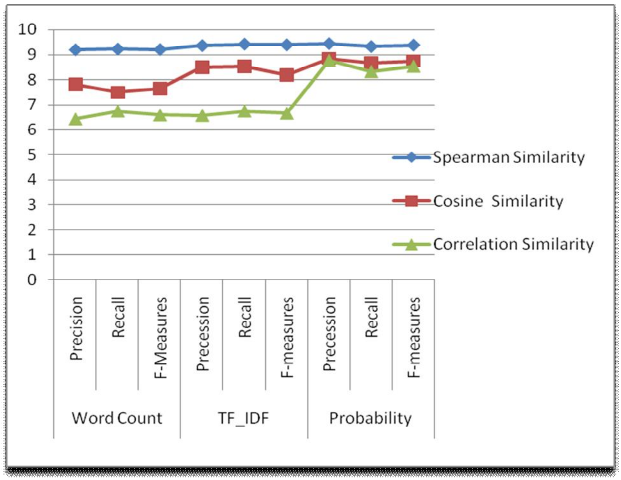


Figure 4: Performance Analysis chart.

6. CONCLUSION

In this paper, the proposed Novel K-Means clustering approach is performed by the WordCount, TF-IDF and Probability features of the documents of five different categories. For each category we have taken 300 documents. The proposed system retrieves 16544 key words out of 160786 unique words for the business category. The Entertainment category documents retrieve 18929 key words out of 164622 unique words. The Politics category retrieves 17422 key words out of 159930 unique words. The sports category retrieves 16919 key words out of 159930 unique words. The Technology category retrieves 18207 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, 94.80, 95.70 for WordCount,TF-IDF and Probability respectively. Cosine Similarity measure yields an accuracy of 76.87, 80.80, 88.40 for WordCount,TF-IDF and Probability respectively. Correlation Similarity measure yields an accuracy of 72.33, 74.40, 86.80 for WordCount, TF-IDF and Probability respectively. The proposed method yields better results for probability based features compared with other two TFIDF and WordCount.

REFERENCES

1. Saqib Alam, Nianmin Yao. **Big Data Analytics, Text Mining and Modern English Language**, *journal of Grid Computing 2018* https://doi.org/10.1007/s10723-018-9452-4
2. Vladimer B. Kobayashi1, Stefan T. Mol1, Hannah A. Berkers1, Ga´bor Kismiho´k1and Deanne N. Den Hartog. **Text Mining in Organizational Research**, *Organizational Research Methods, 21(3), 733-765.2018.*
3. Robert Wing Pong Luk, Kam-Fai Wong, Kui-Lam Kwok "Interpreting TF-IDF Term Weights as Making Relevance Decisions", ACM Transactions on Information Systems, Vol. 26(3) 2008. https://doi.org/10.1145/1361684.1361686
4. Dibyendu Mondal Pushpak, Raksha Sharma. **Comparison among Significance Tests and Other Feature Building Methods for Sentiment Analysis: A First Study**, *International Conference on Computational Linguistics and Intelligent Text Processing, pp.3-19, 2017.*
5. Kasula Chaithanya Pramodh, Dr.P.Vijayapal Reddy. **A Novel approach for Document Clustering using concept extraction**, *International Journal of Innovative Research in Advanced Engineering, 05(3),pp.59-65, 2016.*
6. Charu C. Aggarwal, ChengXiang Zhai. **A survey of text classification algorithms**, *Mining text data. pp.163–222, 2012.* https://doi.org/10.1007/978-1-4614-3223-4_6
7. Borovikov, E. **A survey of modern optical character recognition techniques**, *Computer Vision and Pattern Recognition 2014.*
8. Bsoul, Q., Salim, J., Zakaria, L. Q. **An intelligent document clustering approach to detect crime**

patterns, *Procedia Technology, 11, pp.1181–1187, 2013.*

9. Cohen Priva, U., Austerweil, J. L. **Analyzing the history of cognition using topic models**, *Cognition, 135, pp.4–9, 2015.* https://doi.org/10.1016/j.cognition.2014.11.006
10. Aranzabe, M. J., A. D. de Ilarraza & I. Gonzalez-Dios. **TransformingComplex Sentences using Dependency Trees for Automatic Text Simplificationin Basque**, *SEPLN, pp. 61–68. 2012.*
11. Matthew Honnibal and Ines Montani. Spacy, **Natural language understanding with bloom embeddings**, *convolutional neural networks and incremental parsing. 2017.*
12. Sowmya Vajjalla and Detmar Meurers. **Readability assessment for text simplification: From analysing documents to identifying sentential simplifications**, *International Journal of Applied Linguistics, 165(2)pp.194–222, 2015.*
13. Yuqiang Tong, authorLize Gu. **A News Text Clustering Method Based on Similarity of Text Labels**, *Advanced Hybrid Information Processing,279 pp.496-503, 2018.* https://doi.org/10.1007/978-3-030-19086-6_55
14. Marzieh Oghbaie, Morteza Mohammadi Zanjireh,. **Pairwise document similarity measure based on present term set**, *Journal of Big Data, 5:52,2018.*
15. Marmar MoussaIon, I. Măndoiu. **Single cell RNA-seq data clustering using TF-IDF based methods**, *BMC Genomics 19(Supl 6) : 569 , 2018.*
16. Yehang Zhu,Mingjie Zhang, Feng Shi. **Application of Algorithm CARDBK in Document Clustering**, *Wuhan University Journal of Natural Sciences, 23:6, pp.514-524, 2018.* https://doi.org/10.1007/s11859-018-1357-3