TEST Engineering & Management

January - February 2020
ISSN: 0193 - 4120 Page No. 317 - 329

A weighted TF-IDF Uni-Gram Model for Automated Feature Extraction in Multi – Dimensional and Unstructured Big Data

1* A. Jebamalai Robinson,

Research Scholar, Bharathiar University, Coimbatore. Email: jebamalai.robinson@gmail.com ORCID: 0000-0001-7438-2599

2. Dr. V Saravanan,

Dean-Computer Studies, Dr. SNS Rajalakshmi College of Arts and Science, Coimbatore

Article Info
Volume 82
Page Number: 317 - 329
Publication Issue:
January-February 2020

Article History
Article Received: *14 March 2019*
Revised: *27 May 2019*
Accepted: *16 October 2019*
Publication: *02 January 2020*

Abstract

In the recent trends of the Big-data, A large volume of data mostly unstructured are produced in a variety of forms such as Text, Images, Audio and video. Making use of these data in an effective manner is a tedious and challenging task. Feature Extraction methods are used widely for extracting meaningful information from these large data sources. Data dimensionality is a critical issue when Data mining algorithms are applied to these large data. Although many researches have been conducted in the Feature Engineering of unstructured data to address the dimensional complexity, most of the methods suffer from pitfall in one or the other metrics. This paper proposes an automated Feature Extraction method based onweighted -TF-IDF model using uni-gram vector space methodfor a large text corpus.The document similarity features are calculated using the cosine similarity and Document Clustering is done for grouping the similar features from the corpus together.Experimental analysis prove that the proposed methodology outperforms the other state-of-the-art embedded methods for text feature extraction.

Keywords: *Big data, Feature Extraction, Dimensionality, TF-IDF model, Topic modelling and LDA*

INTRODUCITON

The recent advancement in the technology and storage space have made the accumulation of data grow exponentially with time in the recent decade. The Volume, Variety and Velocity of the large data have drastically improved the computational abilities of the systems. The prediction of IDC says that unstructured will contribute to almost 95% percent of the total data available in the next 5 years [1]. The characteristics that make the retrieval of useful information from unstructured data challenging are first, it takes up different formats. Second, these unlike traditional databases that have a proper schema definition. Third, these do not have a standard and finally it is generated from various sources from basic forms till cloud sensors and IoT devices [2-6]. Owing to the reason that these data are too complex and of large volume, extracting useful information from these have always been a trivial task. Feature

January - February 2020
ISSN: 0193 - 4120 Page No. 317 - 329

Engineering is a process of extracting useful information from large unstructured data. These information extracted from large corpus are used for the data analysis. An efficient and correct FE method can be vital in the data analysis that contributes a lot in any business's success. A handful of research have been conducted for this issue in the recent decade.

Text form of data are the most commonly found unstructured data. These are normally consisted of documents that has words, paragraphs and sentences and has a free flow without any written semantics. These data are normally found to be in-consistent, noisy and unstructured that makes the process of extracting features a tedious and tiresome task. Feature Engineering is perhaps known as the tool for creating a efficient data models [7]. This is more important in case of handling text data as the task here would be to convert the freely flowing text into numeric data which a system or alsgorithm understands.

The high-dimensionality nature of text data will end up producing many features that are least important and thus increases the computational cost and memory required. Feature Extraction is a kind of dimension reduction technique which has been proved as an effective way to handle high-dimension data.

Traditionally analysts used to create features using a manual process from domain or/business knowledge. Often it's called handcrafted feature engineering which are more complex, time consuming and un-biased [8]. Feature Extraction in text data are usually performed using three methods. i. The filter methods which are mostly used in the pre-processing phase. Here, the selection of a particular feature is not dependent on any one machine learning algorithms. The features are extracted based on the weighted scores obtained from different statistical methodsfor the correlation of the output variable. ii. Wrapper methods are the one where the subset of the features are then used for training the model [9]. Iii. Embedded methods are the one that combines the qualities of both (i) and (ii). These are implemented using the algorithms that has in-built methods for feature extraction [10]

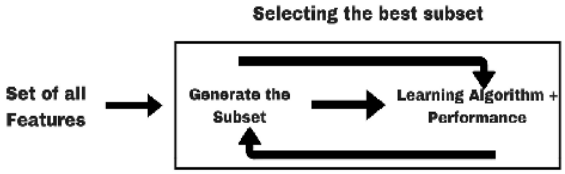

Figure 1- Embedded Feature Extraction Model

This research is carried out using embedded technique namely the Weighted -Term Frequency-Inverse Document Frequency (TF-IDF) with weightedvector space model for automated extraction of features from text documents. Further, document similarity features are calculated using the cosine similarity functions and comparing the pairwise document similarity based on the TF-IDF feature vectors. Document Clustering is done to leverage the unsupervised hierarchical clustering algorithm for grouping the similar documents from the corpus together by leveraging the document similarity features extracted.The rest of the paper is organized as follows Section II gives the gist of available methods in the similar domain. Section III explains the feature extraction problem from text data. Section IV explains the proposed methodology in detail and Section V discusses on the experimental set up and data description. Section VI interprets the results obtained and the comparative analysis. Section VII concludes the paper with closing remarks and with limitations and scope for future research.

I. RELATED WORKS

Jie Chen et al (2016)introduced the adaptive weight concept in the traditional TF-IDF method. This was capable of determining the position weight dynamically based on the position of a particular word. The Vector Space model was

January - February 2020
ISSN: 0193 - 4120 Page No. 317 - 329

introduced and are tested under the scene of Chinese based document clustering. The results showed an increase of 12 % in performance than that of the traditional TF-IDF. Although performed better, this method had a severe drawback in terms of log loss and hinge losses [11].

Ghayda et al (2017) proposed a weighted method which is based on the statistical prediction of the words's importance in a SMS categorization issue. This method was intended to classify the SMS in mobile into a predefined set of classes such as greeting, relation, sales, etc All the messages were converted into unstructured text corpus. Once the pre-processing is over, the Vector model is prepared and the weights are assigned for all the terms. This method was based on the categorization of texts and the experiments proved a great accuracy in the classification. One of the major issue to be addressed is the time complexity when more weights are added to the classifiers [12]

Amandeep Singh et al (2018)introduced the Supervised ML approach as a combination of count vector and the traditional TF-IDF which is based on the Chi-Square for the Feature Extraction. The proposed method combines the different N_gram features that signifies various aspects of the sentiment inside the text data. The proposed method performed better than other model by the use of a count based TF-IDF. The SVM classifier is considered to be the better one after the experimental evaluation with the proposed method [13]

Z.ZHU et al (2019)proposed a re-defined term frequency and the Inverse document frequency for finding the hot topics which is based on the time distribution and attention of the users. A method was also putforth for generating new terms and also to combine terms that are split using the Chinese segmentation method. The hot news are then extracted using the hot terms and K-means clustering is done for grouping for the realization of the hot topics in news. The experiments procve that the result obtained through the proposed refined TF-IDF are effective for finding the hot topics [14]

YAHAV et al (2019)introduced the bias between the participants in the comments mining from social media data. It was found that the content which was extracted from the discourse are often having high correlation which results in structures having dependency in-between recordings in the study and thus introduces a statistical bias. Ignorance of this will lead to a non-robust analysis which can drive the decision makers to a wrong conclusion. The traditional TF-IDF is adjusted here to handle the bias. The experiments were carried out in Facebook data that covers various domains including news, finance , sport and other entertainment[15].

Kim, S et al (2019) proposed a research on classification technique that performs clustering the research papers in to a meaningful group for easy understanding as which research articles are likely to focus on similar subjects. The method proposed extracts the keywords from the asbtracts of the paper using the LDA scheme. The K-Means clustering is performed for the classification of the entire papers into a group of papers that focus on similar subjects based on the TF-IDF value of all the papers [16]

Y. Li et al (2015)presented an innovative solution for the relevant feature identification. It identifies both the positive and the negative patterns in the free flowing text as next level features and then deploys them into lower features. Classification of the terms into suitable categories is also done based on the weights of the specificity and the distribution patters, The experiments show that the proposed method significantly performs better than the other pattern based methods [17].

January - February 2020
ISSN: 0193 - 4120 Page No. 317 - 329

III PROPOSED METHEDOLOGY

The overall architecture can be viewed in four phases namely the preprocessing phase, Feature Extraction phase, Clustering phase and the summarization phase. The Figure 2 shows the architecture of the proposed methodology.

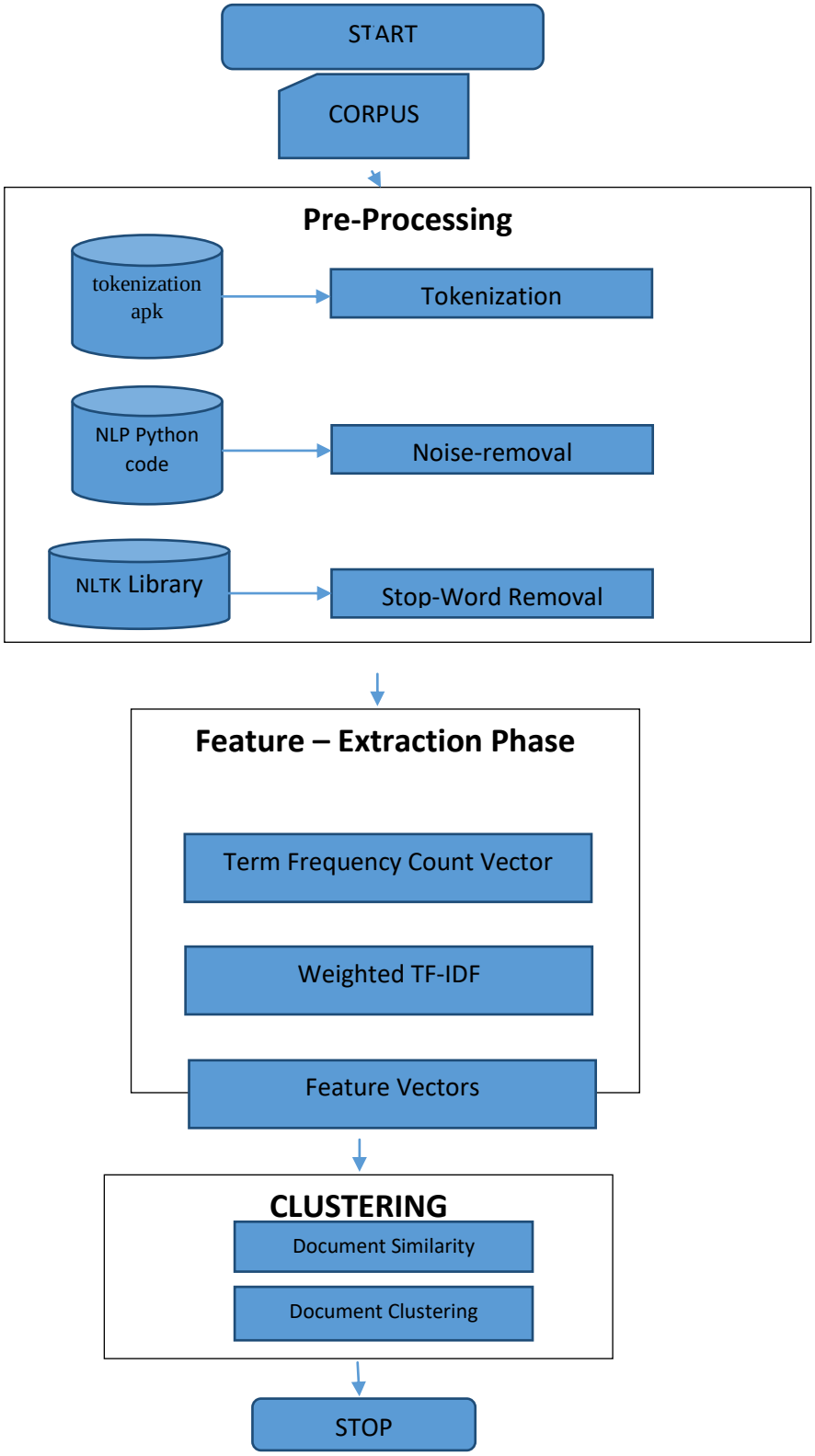

Figure 2 – Architecture of the proposed methodology

January - February 2020
ISSN: 0193 - 4120 Page No. 317 - 329

The proposed architecture is explained on the basis of the corpus considered for the experiment. The text corpus is taken form the US Departmental score card data of size 256 Mb of text content with 2483 text documents. For a better understanding of the proposed methodology. A sample portion as shown in figure 3 from the corpus is considered.

	Document	Category
1	Alex was the topper in the computer paper of class 2016	Performance
2	Tony have got suspension for rude behavior in current term	Conduct
3	Smiley had pneumonia during last term exams	Absentism
4	Alex was born on 19th January 1991 and his parents are no more	personal
5	He got the best outgoing student award of the year	Conduct
6	She has failed in many of the subjects	Performance
7	He is a transgender and his height was 165 Cm	personal
8	All the medical leave submitted are pre-fixed with declared holidays	Absentism

Figure 3 – Sample Corpus

A. Preprocessing

Data preprocessing is carried out in three methods namely tokenization, normalization and stop words removal.

i. Tokenization

The process of chopping the given sentence into smaller parts (tokens) is known as tokenization [18]. By and large, the given crude content is tokenized dependent on a lot of delimiters (for the most part whitespaces).Tokenization is utilized in errands, for example, spell-checking, preparing look, recognizing grammatical forms, sentence discovery, record characterization of reports, and so forth. Tokenizing utilizing OpenNLP.

Step 1 − Instantiating the respective class
Step 2 − Tokenize the sentences
Step 3 –Extract the tokens and save it in the repository that is created manually.

We have used the the tokenization apk available in the Open NLP package for the process .In the data that were considered for our findings

ii. Noise Removal

Noise removal is the next pre-processing step to be performed. The text file do have a lot of noises such as punctuations, short forms and linguistic extensions. These are eliminated by .

- Removing the headers of the files and footers
- Removing the markup such as HTML and XML
- Extracting interesting information from different formats Ex. JSON or from inside the DBs

iii. Stop-word Removal :

Once the cleaning is over, the words like “is, am, are, an” which are considered as the stop words are to be removed. These are done using the NLTK pre-defined stop words [19].The pre-processing pipeline and implemented in the corpus which yielded the results as depicted in figure 4.

```
Norm_Corpus = Normalize_corpus (Corpus)
stop_words = nltk.corpus.stopwords.words('english')
tokens = wpt.tokenize(doc)
OUTPUT :
array (['Alex  topper computer paper class 2016'] , ['Tony, su
pneumonia term exam'],['Ale ,born 19th january 1991 paren
year'] ,['Fail many subject'],['transgender height 165cm'],['N
holiday']
```

Figure 4 – Output of the pre-processed corpus

B. Feature Extraction

i. Term Frequency Count Vector using Uni-Gram model

The VSM (“Vector Space Model”) is a simple mathematical model for the representation of

January - February 2020
ISSN: 0193 - 4120 Page No. 317 - 329

unstructured data , particularly text as numeric vectors in such a way that every dimension that belong to the vector is visoned as specific attribute or a feature. The Uni-gram type of model does the representation of all the documents into a numeric vector where all the dimension will be a definite word inside the corpus and the value eventually is its no of occurrences in the document which is denoted in a binary form (0 or 1)

```
# get all unique words in the corpus
vocab = cv.get_feature_names()
# show document feature vectors
pd.DataFrame(cv_matrix, columns=vocab)
```

Figure 5 – Code snippet for term-frequencyvectorizing -Unigram

The result in figure 6depicts that the documents have thus been converted into numeric vectors which each document represent.

	Alex	Topper	Computer	Paper	Class	Tony	Suspension	Rude	Behavior	…	Medicine	Leave	submit	declare	holiday	Pre-fix
0	1	1	1	1	1	0	0	0	0		0	0	0	0	0	0
1	0	0	0	0	0	1	1	1	1		0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0							
…																
7	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

Figure 6 – Term Frequency Count Vector

ii. Weighted TF-IDF Model :

The TF-IDF model exhibits the importance that a word has, i.e the Term Frequency anad the Inverse of the Document Frequency. The TF gives the total times that a given word appears in a document. The importance of a specific word w(i) is represented as

$$TF\ (Term) = \frac{Total\ \ Terms}{\sum_{n=0}^{i} total\ (Terms\ )_n} \quad (1)$$

In the above equation, the total (terms) represents the count of occurrences of the particular word in the document. The term in the denominator is the addition of the count of occurrences about all of the words in the document. IDF is the measure of the ability of the word for distinguishing among the categories. This can be obtained by the overall count of the documents which can be contained in the document divided by the log of actual number of documents

$$IDF\ (Term) = \log \frac{(Total\ \ docs\ )}{(Total\ \ no\ of\ \ docs\ , words\ )} + 0.01 \quad (2)$$

Total docs is the total count of the documents, the denominator indicates the total count of the document in which the particular word is present. The IDF measure is the count of documents that has the words fewer and which indicates the words has a excellent class discrimination.

$$Weight\ (Term) = TF(Term) * IDF\ (term) \quad (3)$$

The weight is calculated by multiplying the (1) and (2). The classical TF-IDF only considers the weight of the term frequency and do not bother on the weights of the other features and thus the weighted – TF-IDF is proposed. In any text corpus, the title and the initial paragraph plays a vital role in determining the subject of the document. This will be helpful for the document classification and to restate the topic at the end of the document, therefore these words which at the start or end of the document can describe the topic more than other words of the document. Effective extraction of these words at the start or end of the document, will have great determination of the document topic. If the context of each word is analyzed in detail, the algorithm's time efficiency will increase dramatically, The position of first occurrence of a key word and the position of last occurrence of a keyword in a document is considered. The formula for the position of first occurrence of a key word can be described as:

January - February 2020
ISSN: 0193 - 4120 Page No. 317 - 329

$$F.P\ of\ word = \frac{F.P\ prior\ \ count\ \ (Word)+1}{\sum_{i=0}^{n} Count\ (word\ (i) - F.P\ prior\ \ count\ \ (Word)} \quad (4)$$

In (4), the meaning of FPPriorCount(word) is the number of all the words that appear before the position (not including this word) in the first appearance of the key word.$\sum_{i=0}^{n} Count(word(i)$ is the total number of words in the document.The formula for the position of last occurrence of a key word can be described as:

$$L.P\ of\ word = \frac{L.P\ prior\ \ count\ \ (Word)+1}{\sum_{i=0}^{n} Count\ (word\ (i) - L.P\ prior\ \ count\ \ (Word)} \quad (5)$$

In (5), the meaning of LP prior Count(word) is the number of all the words that appear after the position (not including this word) in the last appearance of the key word. The denominator is the total number of words in the document. In summary, the formula for the adaptive weight of key word's position can be described as:

$$Position\ Weight\ Word = \frac{1}{F.P\ (word)+L.P\ (word)} \quad (6)$$

Equation (6) is that the more front position of the key word which appears for the first time is, i.e. thevalue of FirstPosition (word) smaller is, the more likely it isthe word in the document title, abstract or first paragraph, and the more after position of the key word which appears for the last time is, i.e. the value of LPosition(word) smaller is, the more likely it is the word in the document conclusion or last paragraph. Therefore, the formula for the weight of the key word is optimized as

$$Weight\ (Word, Doc) = \frac{TF*IDF*PW}{\sqrt{\sum_{word \epsilon Doc}\ (TF*IDF*PW)^2}} \quad (7)$$

In formula (7), the TF is the occurrence frequency of key word, the IDF is the inverse document frequency of key word and the PositionWeight is the adaptive weight of key word's position.

iii. Feature Vectors

The following is the code snippet applied to the Term Frequency Count vector to obtain the weighted feature vectors

```
from sklearn.feature_extraction.text import TfidfVectorizer
tv = TfidfVectorizer(min_df=0., max_df=1., use_idf=True)
tv_matrix = tv.fit_transform(norm_corpus)
tv_matrix = tv_matrix.toarray()
vocab = tv.get_feature_names()
pd.DataFrame(np.round(tv_matrix, 2), columns=vocab)
```

Figure 7- Code snippet for weighted feature vector generation

	Alex	Topper	Computer	Paper	Class	Tony	Suspension	Rude	Behavior	...	Medicine	Leave	submit	declare	holiday	Pre-fix
0	0.68	0.44	0.57	0.32	0.49	0	0	0	0		0	0	0	0	0	0
1	0	0	0	0	0	0.21	0.34	0.48	0.65		0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0
3	0	0	0	0	0.28	0.42	0	0.28	0		0	0	0	0.68	0.12	0.8
4	0	0.5	0	0.54	0.18	0.1	0	0	0		0.27	0.42	0	0	0.28	0.7
5	0.71	0	0	0.27	0.1	0.54	0.61	0	0		0.18	0.24	0.62	0.18	0.74	0.35
6	0.17	0	0	0	0	0.7	0.37	0.19	0.64		0	0	0	0.27	0.	0
7	0	0	0	0	0	0	0	0	0	0	0.33	0.48	0.62	0.18	0.52	0.64

Figure 7- Weighted Feature Vector representation

The figure 8 represents the weighted Feature vector representation of the corpus considered after applying the relative weights of the words first position weight and last position weights.

C. Clustering

i. Document Similarity :

It is the process where the distance or the similarity is used as a metric which is then used to identify as how similar a certain corpus of text with any other document based on the features that are extracted

January - February 2020
ISSN: 0193 - 4120 Page No. 317 - 329

from the documents. The cosine similarity is calculated as

$$\cos x = \frac{\overrightarrow{a\,b}}{|a|\,|\overrightarrow{b}|} = \frac{\Sigma_1^n a_{ib_i}}{\sqrt{\Sigma_1^n a_i^2}\,\Sigma_1^n b_i^2}$$ where $\vec{a}\,.\,\vec{b}$ is the dot product of the given two vectors. The pairwise similarity is calculated by finding out the document similarity of two corpus. Thus, if there are C no of documents in a given corpus, then CXC matrix can be obtained such that all the row and column will represent the similarity score for a particular pair of document. The cosine similarity is the most used and it gives us the metric which represents the cosine of the angle in-between the vector notations of two documents. Least the cosine angle, greater the similarity. Figure 9 shows the pair-wise similarity matrix of the corpus considered.

	0	1	2	3	4	5	6	7
0	1.000	0.000	0.000	0.121	0.000	0.511	0.000	0.000
1	0.000	1.000	0.000	0.000	0.792	0.000	0.000	0.000
2	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.621
3	0.000	0.000	0.000	1.000	0.000	0.000	0.529	0.000
4	0.265	0.185	0.268	0.158	0.657	0.786	0.621	0.641
5	0.000	0.481	0.101	0.354	0.268	0.157	0.725	0.631
6	0.128	0.264	0.000	0.267	0.741	0.364	0.185	0.369
7	0.327	0.254	0.542	0.000	0.154	0.000	0.265	0.741

Figure 9 pair-wise Document Similarity matrix

Form

the figure, it is understood that the documents (0,5) , (1,4) , (2,7) and (3,6) are falling under similar categories.

ii. Document Clustering

The process of clustering the documents helps to leverage the un-supervised learning into data points as a group. The Minimum variance method is used as the linkage criteria for the minimization of total variance within the cluster. Hence, in each step, the pair of clusters leading to the minimum increase in the total variance is calculated.

```
from scipy.cluster.hierarchy import dendrogram, linkage

Z = linkage(similarity_matrix, 'ward')

pd.DataFrame(Z, columns=['Document\Cluster 1', 'Document\Cluster
2',Distance', 'Cluster Size'], dtype='object')
```

Figure 11 – Code Snipped for the DCM

The figure 11 shows the code snippet of the document clustering method imported for the corpus. The linkage matrix for the corpus obtained is as follows and the figure shows that the features with only low cluster sizes are extracted and in the corpus for our consideration the feature set extracted is as FS= {Performance, conduct , Absentism and personal} .

January - February 2020
ISSN: 0193 - 4120 Page No. 317 - 329

Document	C1	C2	Distance	Cluster Size
0	2	7	0.2145	2
1	0	6	0.2852	2
2	5	8	0.3012	3
3	1	9	0.3421	3
4	3	4	2.8214	8
5	11	12	4.1254	9
6	10	13	6.4587	12

Figure 11 – Linkage Matrix of the corpus

IV EXPERIMENTAL SET UP and METRICS

The experiments are carried out under minimum configuration. Python is used as the programming language and natural language processing tool kit 2.4 is used as the tool to perform pre-defined operations. The student text data from the US educational data hub is taken for the experiment that had 4605 text documents of 256Mb size.

Performance Metrics

The statistical parameters are made used for the error identification and to assess the overall performance of the proposed model. The various metrics used for the evaluation are as follows. Table 1 shows the basic parameters of evaluation.

	Doesn't contain the target object or condition	Contain the target object or condition
Tests Negative or Accepted Null Condition	True Negative	False Negative
Tests Positive or Rejected Null Condition	False Positive	True Positive

Table 1: Fundamental Evaluation Metric

i. True Positive

The true positive is when the final condition marked as matching and correct, which shows the positive condition and denies the null hypothesis [20]. True positive is given with the symbol A. The true positive is given as the following:

$$TP = n_{11} = \text{number of such individuals} \quad (8)$$

ii. True Negative

The true negative is when the final condition marked as non-matching and correct, which shows the negative condition and accepts the null hypothesis. True negative is given with the symbol B. The true positive is given as the following:

$$TN = n_{00} = \text{number of such individuals} \quad (9)$$

iii. False Positive

The false positive is when the final condition marked as matching and incorrect, which shows the positive condition and denies the null hypothesis. False positive is given with the symbol C. The false positive is given as the following:

$$FP = n_{01} = \text{number of such individuals} \quad (10)$$

iv. False Negative

The false negative is when the final condition marked as non-matching and incorrect, which shows the negative condition and accepts the null hypothesis. False negative is given with the symbol D. The false negative is given as the following:

$$FN = n_{10} = \text{number of such individuals} \quad (11)$$

v. Recall

It is the test based on probability for accuracy that indicates the overall performance of the model in the presence of false-negative scenarios. This is calculated as

$$Recall := \frac{True\ Positive}{True\ Positive\ + False\ Negative} \quad (12)$$

vi. Precision

It is the test based on probability for accuracy that indicates the overall performance of the model in the presence of positive scenarios. This is calculated as

$$Precision := \frac{True\ Positive}{True\ Positive\ + False\ Positive} \quad (13)$$

vii. **F1-Measure**

It is the cumulative measure to assess the oveall impact that the precision and the recall has in a case. The F-Measure is represented between 0 and 1 or from 0 to 100 and are normally decided on the case and ranges of precision and recall. It is measure by

$$F1 - Measure := 2 * \frac{(R * p)}{R + P} \quad (14)$$

Where R is recall, and p or P is precision.

TEST Engineering & Management

January - February 2020
ISSN: 0193 - 4120 Page No. 317 - 329

viii. Accuracy

The overall value for the accuracy is obtained by dividing the total number of true cases by all of the cases.

$$Accuracy := \frac{True\ Positive + True\ Negative}{True\ Positive + True\ Negative + False\ Positive + False\ Negative} \quad (15)$$

ix. Logarithmic Loss:

Cross-entropy loss or in other words, the log loss, is a performance measure of a given classification whose output is the probability between 0 and 1. The value increases as the prediction value is deviated from the original label. In case of a multiclass classification, the log loss is calculated using

$$Log\ loss = \sum_{c=1}^{m} y_{0,c} \log p_{0,c} \quad (16)$$

x. Mean Square Error:

Mean square error is the mean of the squared difference in-between predictions and the actual observations. It is calculated as $MSE = \frac{\sum_{i=1}^{n}(y_i - \widehat{y^2})}{n}$ (17)

V RESULTS AND DISCUSSION

The results obtained through various experiments are tabulated and explained in the following section. The results obtained are compared with that of other state-of-the-art models such as TF-IDF (Term Frequency Inverse document Frequency) , Glove (Global Vectors for words representation) , W2V (Word2vec), VSM (Vector Space Model) and BOW (Bag of Words). The table 1 shows the comparative analysis of the no of features extracted from the same documents, their entropy and minimum variance value obtained with that of the other models.

Method	Evaluation Measures		
	No Features Extracted	Entropy %	Minimum variance %
TF-IDF	51	56	57
GLove	62	76	74
W2V	74	71	62
VSM	82	74	70
BOW	95	64	81
WTF-IDF	36	39	52

Table 2- Comparative Analysis of No of Features Extracted, Entropy and MV

It is observed from the table that the weighted TF-IDF model produces the less no of features when compared to other models. This is because the proposed method concentrates more accurately on the relative weights and thus leaving the redundant and un-important features unlike others and it is evident from the low entropy and Minimum variance scores. Figure 12 depicts the same in graphical manner.

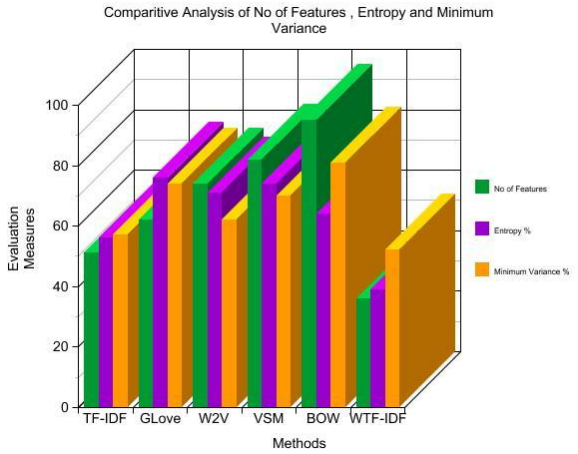

Figure 12 – Comparative analysis of No of Features, Entropy and Minimum Variance

Table 3 brings the comparative results of the most used performance metrics namely Precision, Recall F-Measure. It is seen from the table that the proposed method has high precision and recall and thus proving that the proposed method produces high accurate results (high precision) and also returns a major of positive results (high recall

Method	Performance Metrics		
	Precision	Recall	F-Measure
TF-IDF	0.89	0.81	0.84
GLove	0.81	0.76	0.72
W2V	0.76	0.71	0.69
VSM	0.79	0.74	0.70
BOW	0.68	0.64	0.59
WTF-IDF	0.94	0.96	0.98

Table 3 – Comparative analysis of precision, Recall and F-Score

January - February 2020
ISSN: 0193 - 4120 Page No. 317 - 329

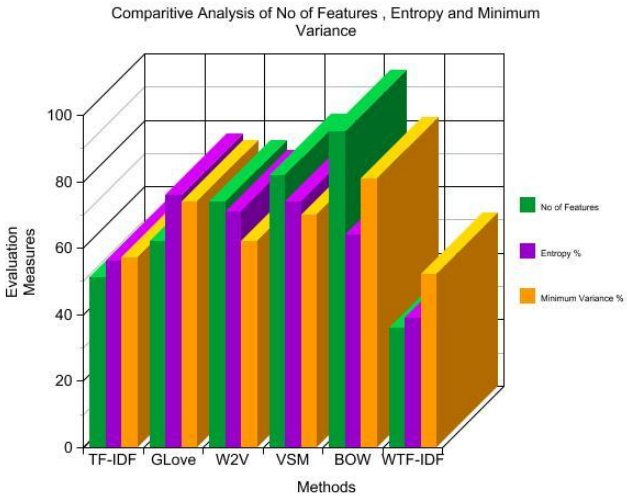

Figure 13 – Comparison of Precision, Recall and F-Measure

Table 4 gives the comparative results of the various models in terms of accuracy. It is seen that the proposed model has the highest accuracy of 94%. The same is depicted in figure 14 as a graphical form.

Methods	Accuracy
TF-IDF	0.89
GLove	0.81
W2V	0.76
VSM	0.79
BOW	0.68
WTF-IDF	0.94

Table 4 – Comparative results of accuracy

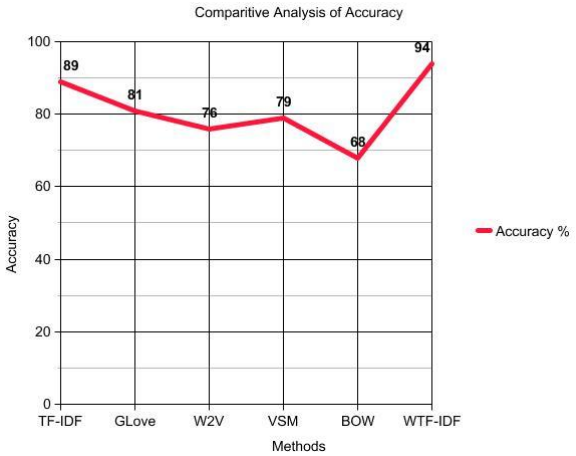

Figure – 14 Comparison of Accuracy

Method	Performance Measures	
	Logarithmic Loss	Mean Square Error
TF-IDF	0.26	0.21
GLove	0.41	0.36
W2V	0.37	0.51
VSM	0.59	0.44
BOW	0.61	0.64
WTF-IDF	0.14	0.11

Table 5 – Comparison of Losses

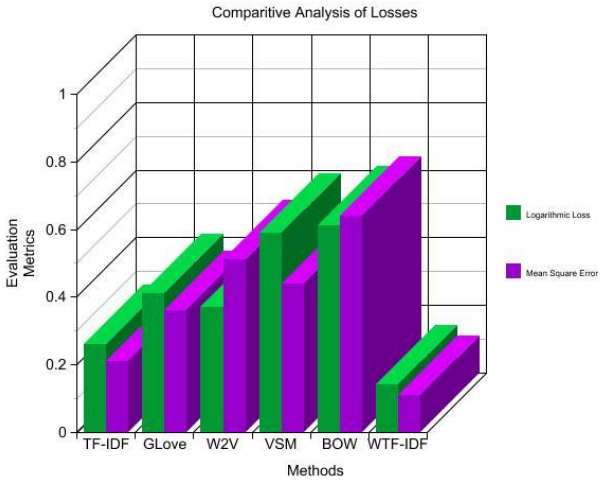

Figure 15 – Comparison of Losses

Table 5 depicts the comparative results of the losses namely the log loss and MSE. The same is represented in graphical form in figure 15. It is obvious that the proposed methodology outperforms the other methods by having minimum losses. Ten performance metrics were chosen for the analysis of the results and the proposed methodology

VI. CONCLUSION

The process of Feature Extraction is used for the extraction of useful information from large unstructured data. The accumulation of data is happening at a very rapid rate because of the technology advancements and modern storage facilities. Effective information extraction from these raw and big unstructured data are always tedious and challenging. Feature engineering is the best known solution for reducing the dimensions of the Big-Data to be analyzed. Even though a handful of researches have already been conducted in the domain, there are

January - February 2020
ISSN: 0193 - 4120 Page No. 317 - 329

few gaps that still exists. This paper proposes an automated Feature Extraction method based on TF-IDF - model using Skip gram from text corpus. Further, document similarity features are calculated using the cosine similarity and Document Clustering is done for grouping the similar features from the corpus together. Experimental analysis prove that the proposed methodology outperforms the other state-of-the-art embedded methods for text feature extraction. The research still has certain limitations such as the sparsity in the features that are extracted and also improvements are planned to be implemented so as to bring down the dimensionality so as to decrease the algorithm complexity and time complexity as well.

REFERENCES

1. Gantz J, Reinsel D. The digital universe in 2020: big data, bigger digital shadows, and biggest growth in the fareast. IDC iView IDC Analyze Future. 2012;2007(2012):1–16.
2. Wang Y, Kung LA, Byrd TA. Big data analytics: understanding its capabilities and potential benefts for healthcareorganizations. Technol Forecast Soc Change. 2018;126:3–13.
3. Lomotey RK, Deters R. Topics and terms mining in unstructured data stores. In: 2013 IEEE 16th international conference on computational science and engineering, 2013. p. 854–61.
4. Lomotey RK, Deters R. RSenter: terms mining tool from unstructured data sources. Int J Bus Process Integr Manag.2013;6(4):298.
5. Schefer T, Decomain C, Wrobel S. Mining the Web with active hidden Markov models. In: International conferenceon data mining. New York: IEEE; 2001; p. 645–6.
6. Lomotey RK, Jamal S, Deters R. SOPHRA: a mobile web services hosting infrastructure in mHealth. In: First international conference on mobile services. New York: IEEE; 2012; p. 88–95
7. Ma Q., Tanaka K. (2005) Context-Sensitive Complementary Information Retrieval for Text Stream. In: Andersen K.V., Debenham J., Wagner R. (eds) Database and Expert Systems Applications. DEXA 2005. Lecture Notes in Computer Science, vol 3588. Springer, Berlin, Heidelberg
8. X. Ren, Y. Zhou, Z. Huang, J. Sun, X. Yang and K. Chen, "A Novel Text Structure Feature Extractor for Chinese Scene Text Detection and Recognition," in IEEE Access, vol. 5, pp. 3193-3204, 2017
9. Y. Ren and D. Li, "Fast and Robust Wrapper Method for N -gram Feature Template Induction in Structured Prediction," in IEEE Access, vol. 5, pp. 19897-19908, 2017
10. L. Jiang, L. Zhang, C. Li and J. Wu, "A Correlation-Based Feature Weighting Filter for Naive Bayes," in IEEE Transactions on Knowledge and Data Engineering, vol. 31, no. 2, pp. 201-213, 1 Feb. 2019
11. Jie Chen, Cai Chen and Yi Liang," Optimized TF-IDF Algorithm with the Adaptive Weight of Position of Word" Advances in Intelligent Systems Research, volume 133 , PP 114-117
12. Ghayda A. Al-Talib, Hind S. Hassan,"A Study on Analysis of SMS Classification Using TF-IDF Weighting" International Journal of Computer Networks and Communications Securit,VOL. 1, NO. 5, OCTOBER 2013, 189–194
13. Amandeep Singh,DineshKumar,"IMPROVED FEATURE SELECTION FORCLASSIFICATION OF SENTIMENTAL REVIEWS USING N-GRAM MACHINE LEARNING APPROACH", GLOBAL JOURNAL OF ENGINEERING SCIENCE AND RESEARCHES , July 2018 , Vol 5 issue 7 ,PP 278-288
14. Z. Zhu, J. Liang, D. Li, H. Yu and G. Liu, "Hot Topic Detection Based on a Refined TF-IDF Algorithm," in IEEE Access, vol. 7, pp. 26996-27007, 2019
15. Yahav, O. Shehory and D. Schwartz, "Comments Mining With TF-IDF: The Inherent Bias and Its Removal," in IEEE Transactions on Knowledge and Data Engineering, vol. 31, no. 3, pp. 437-450, 1 March 2019
16. Kim, S., Gil, J. Research paper classification systems based on TF-IDF and LDA schemes. Hum. Cent. Comput. Inf. Sci. 9, 30 (2019) doi:10.1186/s13673-019-0192-7

January - February 2020
ISSN: 0193 - 4120 Page No. 317 - 329

17. Y. Li, A. Algarni, M. Albathan, Y. Shen and M. A. Bijaksana, "Relevance Feature Discovery for Text Mining," in IEEE Transactions on Knowledge and Data Engineering, vol. 27, no. 6, pp. 1656-1669, 1 June 2015
18. Nayak, Arjun &Kanive, Ananthu&Chandavekar, Naveen &Ramasamy, Balasubramani. (2016). Survey on Pre-Processing Techniques for Text Mining. International Journal Of Engineering And Computer Science.
19. Salloum, Said & Al-Emran, Mostafa &Monem, Azza&Shaalan, Khaled. (2018). Using Text Mining Techniques for Extracting Information from Research Articles
20. Xi Yang, TianchenLyu, Qian Li, Chih-Yin Lee, Jiang Bian, William R. Hogan, YonghuiWu,"A study of deep learning methods for de-identification of clinical notes in cross-institute settings",BMC Med Inform DecisMak. 2019; 19(Suppl 5): PP 232