Automatic Text Classification Of News Blog using Machine Learning.

Rucha Jayantkumar Nikam[1], Vishakha Nitin Salunkhe[2], Asst.prof.S.S.Jadhav[3]

[1st]Rucha Jayvantkumar Nikam, MCA YTC, Satara
[2nd]Vishakha Nitin Salunkhe, MCA YTC Satara
[3rd]Prof.S.S. Jadhav Dept,of MCA Yashoda Technical Campus, Satara-415003

***Abstract*—**In recent years, due to the tremendous growth of information, text classification becomes a need for humans. In this project the data is to be classified into the various groups as per the existing content. This can be done by the training data to the machine. A set of full-text documents is used to train the machine. This paper illustrates the classification process by using automatic text classification. We have vectorized the training data using a count vectorizer. Then the TF-IDF (Term Frequency-Inverse Document Frequency) is used for the normalizing data. Finally the Stochastic Gradient Descent Machine algorithm is used to classify the data.

Keywords - component; Classification, Word2Vec conversion, TF-IDF information retrieval, text classifier

I. INTRODUCTION

The machine learning technology is now playing an important role in the research and development sector. Now a day's many people have their busy schedule in their day to day life. Due to this it is difficult to manage time for them. Machine learning will help them to reduce the time. This paper illustrates a system which is proposed to save the user time and their troubling. In this paper the classification of news data takes place. As the news spread in a wide range and it has a great influence on our society [9]. Therefore the classification and processing of news become an important factor. This system consists of the following processes: preprocessing, vectorization, normalization, and classification [1]. Previously the Chinese text has been classified but the classification of the Chinese text is different than the English text classification [4]. This paper suggests the proper algorithm of text classification. It can also state that the accuracy score of text classification in different algorithms. These accuracy scores will be calculated by using mean functions in the python and different libraries of python.

II. LITERATURE SURVEY

The M. Ikonomakis, Kotisiantis, Tampakias S.V suggested the text classification process using machine learning algorithms and gives an overview of all algorithms which can be preferable to implement text classification. Text classification plays an important role in information extraction, summarization, text retrieval, and question-answering [1]. Mita K. Dalal, Mukesh A. Zaveri suggested that automatically classification of sports blog data, to the appropriate category of the sport by steps like pre-processing, feature extraction and Naïve Bayes classification [2].Cai-Zhi Liu, Yan-Xiu Sheng, Zhi-Qiang Wei, YongQuan Yang suggested a vector representation of feature words based on the deep learning tool Word2vec, and the weight of the feature words is calculated by the improved TF-IDF algorithm. By multiplying the weight of the word and the word vector, the vector representation of the word is realized and finally the vector representation by accumulating all the word vector [3]. Fang Miao, Pu Zhang, Libiao Jin comparison of 3 different machine learning algorithms used for text classification. Knearest Neighbor, Naïve Bayes and Support Vector Machine, among the algorithm SVM is more compatible with the bigger dataset and smaller datasets as compared to K-nearest and Naïve Bayes[4].

III. RELATED WORK

This system provides the facilities to users such as to save their time by using automatic text classification. This paper suggests a supervised machine learning algorithm. This system will train the machine which contains 20 L news data which is in the tabular form. This training data holds the news and its categories itself. Firstly the machine will extract the features from the training data, tokenize it then extracted data will be arranged in vector format by using count vectorizer, which converts text documents into tokens [2]. This count matrix is given to the TF or TF-IDF vectorizer (Term Frequency & Term Frequency Inverse Term Frequency). This vectorizer transforms a count matrix to a normalized TF or TF-IDF representation. This is a complete process of training the machine. After successful completion of the training, the machine will give the testing data to measure the performance or to check the machine working, whether the machine is been trained accurately or not. The accuracy ratio totally depends on the number of training data. As much bigger the training data that accurate the result might be given by the trained machine [6].After successful completion of the training of the machine, testing data is passed to the machine. This testing data will consist of news as well as the type or category of the news. This testing data will also be present in the tabular format i.e. it can be in.csv file. We will pass the testing data in the .csv file which can be known as a comma-separated file. This consist of only the data which will be passed to the machine for categorizing purpose only. This file is passed to the pipeline in which the count vectorizer, TF-IDF vectorizer and SGD classifier will work parallel to each other. In count vectorizer and TF-IDF the same operation will be performed [3]. This whole pipeline will cover the operations like tokenizing, creating vector by using the count vectorizer, normalize this vector using TF-IDF transformer. Then this normalized data is taken by the SGD Classifier. SGD Classifier is a linear classifier (SVM, logistic regression,) with SGD training [10].

IV. METHODOLOGY

The Automatic text classification uses different types of methods. This can include various phases of classification [5]. The phases mainly consist of preprocessing, vectorization, normalization and classification. Fig. (I) shows how the text classification takes place by using the training and testing data. The training and testing data should be pipelined to form the classification result. This training set contains 20 Lac news. Which are stored in a CSV (Comma Separated Values) file. The training set consists of labels. These data is in a structured form because supervised machine learning doesn't allow unstructured data.

In supervised machine learning, the data is to be extracted from the training dataset which is in the labeled form. A supervised learning algorithm analyzes the training data set and produces a conclude function which can be used for mapping a new example. Supervised text classification means that you have a set of examples where we know that the correct answer. for example if we have set of flower and color data like, 'Rose','Tulip',' Sun-flower'] belongs to flower category and['Red','Black','Blue'] belongs to a color category.

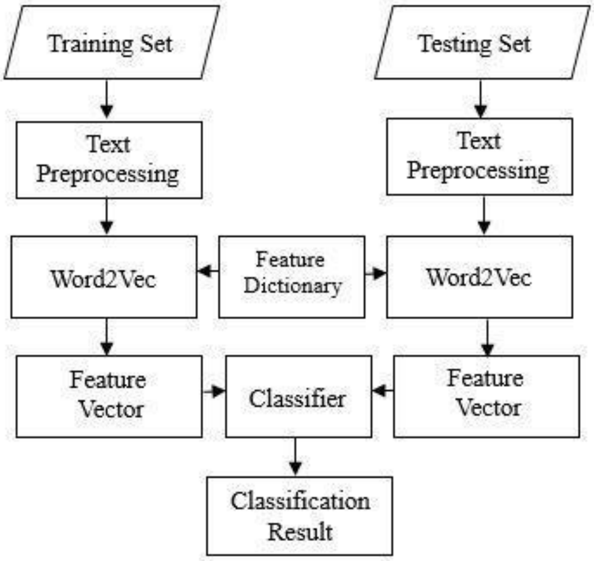


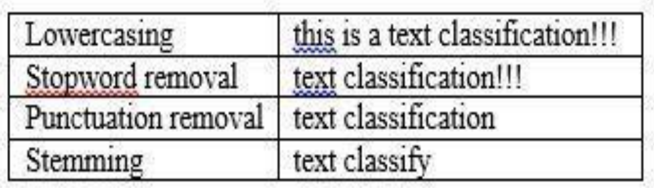
Figure 1. Classification diagram

A. Preprocessing

The text pre-processing is said to be a phase in which the data is pre-processed in such a way that it is predictable to the machine. The first step of preprocessing is lowercasing and stop-ward removal [8]. This pre-processing is very important in natural language processing.

The next is the removal of punctuations and stemming. Stemming means to remove the stem from the word. The pre-processing also called the cleaning of the text document. The cleaning of all the documents will take place here. The pre-processing of a text document is shown in the following table: E.g. this is a text classification!!!

TABLE I. PREPROCESSING



Lowercasing	this is a text classification!!!
Stopword removal	text classification!!!
Punctuation removal	text classification
Stemming	text classify

B. Vectorization

The vectorization is the process of mapping the words from the vocabulary. This can then convert into the vector form. These vectors can be used for finding semantics and to predict the word. For this vectorization, a count vectorizer is used. This count vectorizer is mainly used in the building of the vocabulary of the known datasets. By using count vectorizer raw data can be converted into the vector representation i.e. into the n-grams. Finally n number samples and n number of features will be displayed. The following Figure shows the process of vectorization.

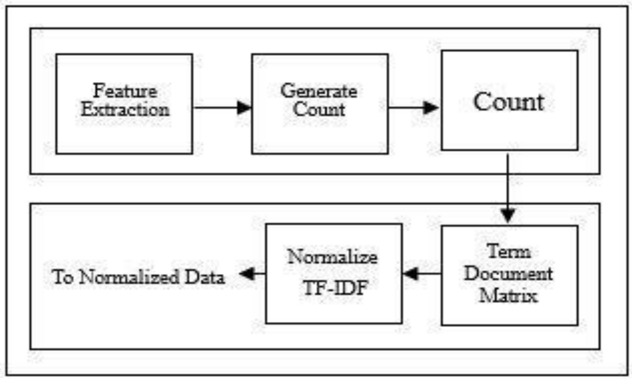


Figure 2. Vectorizer

C. Normalization

The normalization is the process of reducing the weight of the text so that it becomes easy to classify the text. The TF-IDF (Term Frequency-Inverse Document Frequency) is used for the normalization process. The TF-IDF is said to be another way of representing text [3]. This TF-IDF can be calculated by using the following formulas:

TF (Term Frequency)-

$$\mathrm{tf}(t,d) = 0.5 + 0.5 \cdot \frac{f_{t,d}}{\max\{f_{t',d} : t' \in d\}}$$

Where,

TF (t,d) : Term frequency in a document. ft,d : Raw count

IDF (Inverse Document Frequency)-

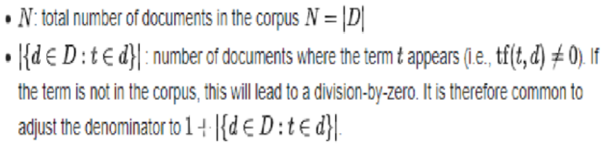
$$\mathrm{idf}(t,D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

Where,

TF-IDF(Term Frequency-Inverse Document Frequency)-

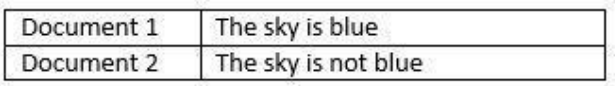
$$\mathrm{tfidf}(t,d,D) = \mathrm{tf}(t,d) \cdot \mathrm{idf}(t,D)$$

The TF-IDF of following document is as follows:

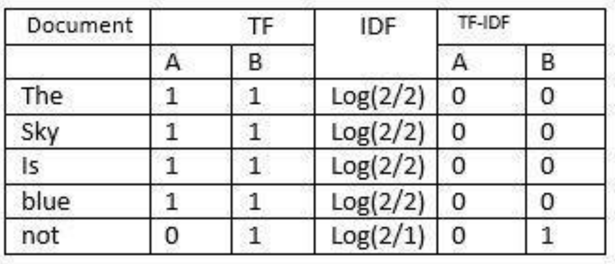


- N: total number of documents in the corpus $N = |D|$
- $|\{d \in D : t \in d\}|$: number of documents where the term t appears (i.e., $\mathrm{tf}(t,d) \neq 0$). If the term is not in the corpus, this will lead to a division-by-zero. It is therefore common to adjust the denominator to $1 + |\{d \in D : t \in d\}|$.

The above table shows how TF-IDF is calculated.



Document 1	The sky is blue
Document 2	The sky is not blue



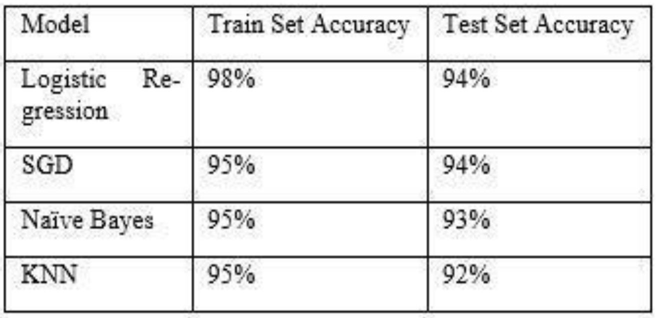
Document	TF		IDF	TF-IDF	
	A	B		A	B
The	1	1	Log(2/2)	0	0
Sky	1	1	Log(2/2)	0	0
Is	1	1	Log(2/2)	0	0
blue	1	1	Log(2/2)	0	0
not	0	1	Log(2/1)	0	1

Basically TF-IDF is used to transform the count vectorizer into the normalized data [3]. The TF-IDF can vectorize the document until it fits into the normalized form. The TF is the tern frequency which can reflect the internal feature of the text document and shows into 0 or 1 form. IDF is the inverse document frequency. It is the inverse of the TF. The IDF is used for the distribution of features in the text document. So that it shows the result in the n number of samples and n number of features. TF-IDF is an algorithm that is very easy and simple to learn. Therefore TF-IDF is suitable for the normalization.

D. Classification

There are various classification algorithms there but the SGD (Stochastic Gradient Descent) is the most preferable algorithm in the machine algorithm. The SGD is based on the gradient descent algorithm which is used to improve the speed. This can works on the iterations so that it is also called an iterative method [10]. The SGD can select the data points from a large number of datasets and calculates the gradient. Hence the speed is improved. The evaluation metrics of the different models are as follows:

TABLE III. COMPARISON OF ALGORITHMS



Model	Train Set Accuracy	Test Set Accuracy
Logistic Regression	98%	94%
SGD	95%	94%
Naïve Bayes	95%	93%
KNN	95%	92%

From the above evaluation it is cleared that the logistic regression has high training set accuracy but has low testing set accuracy [7]. The SGD has the highest test sets accuracy and is near to training set accuracy. Others can't fit in the accuracy matrics. So the SGD is the best for the text classification.

V. RESULT AND DISCUSSION

From the table of evaluation matrices i.e. TABLE (III), it is cleared that stochastic gradient descent is the best algorithm for the text classification. The experimental data which is used for the text classification is the raw data. This raw data is then preprocessed and vectorized by using vectorization algorithms.

First the Naïve Bayes algorithm is used for the classification but later it is identified that this algorithm can take too much time and it gives less accuracy than the stochastic gradient descent. The Naïve Bayes classifier is not efficient for a large number of datasets. Then the SGD algorithm is used for the classification. With the help of a gradient descent algorithm, the linear regression problem is solved. Gradient descent is the iterative method that can be used to increase the speed. This gradient descent algorithm is based on the slope and the gradient.

This algorithm can be worked in the following steps:

1. Compute the gradient function by finding the slope of the objective with respect to the feature set.
2. Take any random initial value for the parameter (e.g. differentiate 'y' with respect to 'x'. If there are more features then take a partial derivative of 'y' with respect to all features).
3. By plugging the parameter update the gradient function.
4. Then calculate the step size for every feature. The step size is the product of gradient and learning rate
5. Then calculate the new parameter and it is the difference between old parameter and the step size.
6. Finally repeat the steps 3-5 until the gradient becomes almost 0.

The SGD can randomly pick the one data point from the set of whole data so that computations are enormously reduced [10].

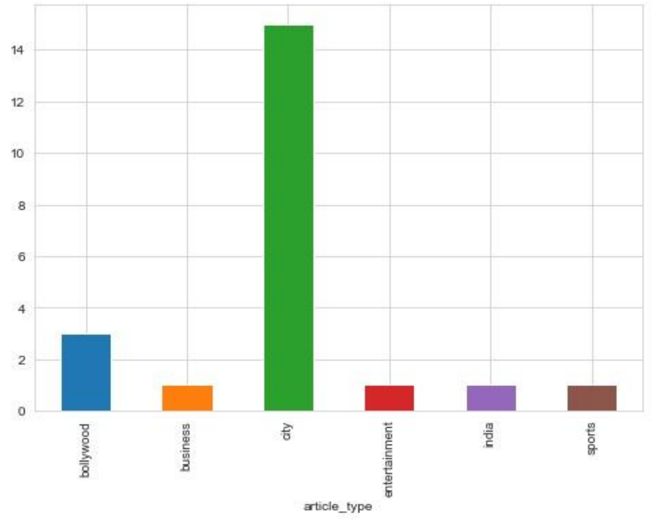


Figure 3. Classification graph

After using the classifier the testing data and training data are pipelined for the simultaneous execution. And the result of the classification is as follows:

The above graph classifies the categories of news. This can be classified by using the SGD classifier and the graph is plotted by using the marplots library. The marplots are the python library which is used in the python for plotting graph. In testing data, there are several number of news will be passed in which the machine learns this data with the help of a training set and it will also identify which news is from which category. The machine understands it and shows the counts of categories. So it can be identified that news can be classified in a different category.

VI. CONCLUSION

This paper suggests that there are various algorithms used for the text classification. But some of them take more time to classify; some can work on small datasets etc. The SGD is the algorithms that overcome all these points. The Stochastic Gradient Descent is said to be a speedy algorithm because it can work with slope and gradient [10]. With the help of the SGD algorithm the text classification system is designed. This system constructs a module that predicts the accurate ratio of the testing data which consists of different news data. The predicted ratio is between 0 to 1. This ratio will be a means of all that testing data i.e. news. And finally the news data is classified which is shown by the graph. The graph denotes the specific category of that news. It can be concluded that SGD is the most preferable algorithm in the text classification from the different algorithms like naïve Bayes, KNN, logistic regression etc., so the system can classify the text with high accuracy and more speed.

REFERENCES

[1] M. Ikonomakis, S. Kotisiantis, V. Tampakias (2005) "Text Classification Using Machine Learning Techniques".

[2] Mita K. Dalal, Mukesh A. Zaveri (2012) "Automatic text classification of sports blog data".

[3] Cai-zhi Liu, Yan-xiu Sheng, Zhi-qiang Wei, YongQuan Yang(2018) "Research of Text Classification Based on Improved TF-IDF Algorithm".

[4] Fang Miao, Pu Zhang, Libiao Jin (2018) "Chinese News Text Classification Based on Machine learning algorithm".

[5] IFIP conference paper (2006) on "comparison of SVM and some older classification algorithms in text classification tasks".

[6] Zhenzhong Li, Wenqian Shang, Menghan Yan (2016)"News Text classification model based on topic model".

[7] IOP Conference series (2018) "Comparison of Naïve Bayes and K Nearest Neighbour Methods to predict divorce issues".

[8] Mita K. Dalal, Mukesh A. Zaveri (2013) “Automatic Classification of unstructured blog text”.

[9] Wu Da-Sheng, Yu Qin-fen, Liu Li-juan (2009) “An Efficient Text Classification Algorithm in ECommerce Application”.

[10] Sebastian Ruder (2017) “An overview of gradient Descent Optimization algorithms”.

1. Then calculate the new parameter and it is the difference between old parameter and the step size.

2. Finally repeat the steps 3-5 until the gradient becomes almost 0.

The SGD can randomly pick the one data point from the set of whole data so that computations are enormously reduced [10].