International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-5, March 2020

Detection of Hate Speech and offensive Language on Sentiment Analysis using Machine Learning Techniques

Guduri Sulakshana, R Siva jyothi, Aluri Lakshmi

Abstract: Toxic online content (TOC) has become a significant problem in current day's world due to uses of the internet by people of distinct culture, social, organization and industries background and followed Twitter, Facebook, WhatsApp,Instagram, and telegram, etc. Even now, there is lots of work going on related to single-label classification for the text analysis and to make less comparative to errors and more efficient. But in recent years, there is a shift towards the multi-label classification, which can be applicable for both text and images. But text classification is not much popular among the researchers when compared to the grading for images. So, in this work, we are using the dataset which is going to be a short messages dataset, to train and develop a model which can tag multiple labels for the messages. Hate speech, and offensive language is a key challenge in automatic detection of toxic text content. In this paper, to contribute term frequency–inverse document frequency(Tf-Idf), Random forest, Support Vector Machine (SVM),and Bayes Naïve classifier approaches for automatically classify tweets. After tuning the model giving the best results, it achieves an Efficient accuracy for evaluating test data analysis. In this contribution of work also moderate and encapsulate paradigms which will communicate and working between the user and Twitter API. Instead of using the traditional techniques like Bag of words or word counter, a new technique which uses Tf-Idf is built to find the similarity, and the text is transformed into the vectors using Tf-Idf, and this is used to train the model using supervised learning technique along with the labels from the dataset. The accuracy of the model is quite good and more efficient with better results.

Keywords : Twitter, toxic text, Tf-Idf, machine learning.

I. INTRODUCTION

Multi-label classification is one of the most difficult and interesting technique in a classification where we generate not one but many classes for the input. As text falls into a natural language process and the classifier cannot work on natural language, we need to transform them into some other format so the classifier can understand. Many text transformation techniques are used for this purpose that is used in common they are bow (bag of words) and word embedding, which includes a glove, word2vec. We use these techniques to transform and work with text/natural language. In this work, we are going to implement a multi-label(n-grams) classifier using machine learning for our short message's dataset. Short messages are similar to a short message that we use to communicate in our daily life. We build our model using pipelining technique/ pipelines to automate the workflows/process and annotation to handle our text. The system is implemented in the following steps: Data Collection/Generation**:** The tweets related to Hate Speech and Offensive languages are retrieved using the Twitter API and Tweepy module of Python. Data Preprocessing: This step involves cleaning and simplifying the data collected by applying various preprocessing techniques such as removal of stop words, handling missing values, removal of irrelevant characters, etc. Feature Extraction: The feature extraction step identifies the features of the four classes used in this work. The feature extractor function is responsible for generating feature vectors. Feature Extraction improvement: The most important features are considered, and the features which have similar context are manually added to the feature vector. This helps the model to be trained in a better way and classify the tweets with higher accuracy. Training the Naïve Bayes Classifier: The feature vectors are used for training the Naïve Bayes Classifier, which calculates the probabilities of each term for each class. Prediction: The model is now capable of making predictions of which class the tweet belongs to with higher accuracy than the baseline model. The tweet is given as an input to the model, which gives the label of the tweet as the output.

II. BACKGROUND WORK

Existing system

Unigrams and Pragmatic approaches are used in the hate speech detection, and it becomes a major problem in current day's world due to uses of the internet by people of distinct culture, social, organization and industries background on Twitter, Facebook, WhatsApp, Instagram, and telegram, etc. Even now, there is lots of work going on related to single-label classification for the text analysis and to make less comparative to errors and more efficient. So, this is the reasons behind people facing a lot of problems of HSOL on sentiment analysis. The disadvantage of existing is Critical to find out the toxic text content problem for all perspectives.

Revised Manuscript Received on March 30, 2020.
* Correspondence Author
Mrs. Guduri Sulakshana, Dept. of CSE, Institute of Aeronautical Engineering, Hyderabad, India. suhaasraj21@gmail.com
Ms. R Siva Jyothi*, Dept. of CSE, KSRM College of Engineering, Kadapa, India. rentalasivajyothi@ksrmce.ac.in
Ms. Aluri Lakshmi, Dept. of CSE, Institute of Aeronautical Engineering, Hyderabad, India. alurilakshmi@gmail.com

III. PROPOSED SYSTEM

The proposed system balanced and address the Tf-Idf, NLP, SVM, and Random Forest to the existing problem of hate speech offensive language with all sample inputs based on sentiment analysis using Twitter API.

The advantage of the system made automatically detects toxic text content and to avoid the hateful, offensive word from the tweets.

Methodology: The methodology of a system improves on the baseline model or paradigms by introducing a new technique to identify the similarity of the hate speech sentences and offensive languages, and it addresses the issues of the existing baseline model. The main aim of this system is to increase the accuracy and more reliable in finding the similarity of the sentences on hate speech and offensive. The proposed system employs the Tf-Idf a Natural Language processing technique. The Tf-Idf vectorizes the text, which is in Natural Language into a vector which is used by the Machine Learning model to find the similarity of hate speech and offensive languages. The vector is generated by assigning some weights to the words by using the Tf-Idf technique, which is the productivity of two parts, which is based on the frequency and the Inverse document frequency. The vector is also normalized because if the number of documents or the questions increased the weight also increases and becomes difficult to perform operations on this so by using normalization, we reduce the range and also the weights in the vector. The model is now able to identify the similarity between the sentences more effectively. This has also helped to increase the accuracy of the model.

Naive Bayes classifier algorithm is s Classification technique which maps the specific class/label to the input. It can be any of the categories below they are Binary classification, Multi-class classification, and Multi-label classification. NB classifier algorithm is the concept of Bayes Theorem and which makes strong independence assumptions between the features [15].

$$\textbf{Probability}\ (C_k \mid x) = \frac{\textbf{Probability}(Ck)\ \textbf{Probability}(x \mid Ck)}{\textbf{Probability}\ (x)}$$

Calculate the probability of each feature in the tweet with the help of Bayes Theorem. Probability of each feature in a tweet should be calculated for all the classes. In the end, the tweet is classified under the class having the highest probability. How Naïve Bayes is used in sentiment classification**:**

$$\textbf{Probability}\ (C_k,\ x_1, x_2, \ldots\ldots, x_n) = \textbf{Probability}\ (x_1, x_2, \ldots\ldots, x_n, C_k)$$

$$= \textbf{Probability}\ (x_1 \mid x_2, \ldots\ldots, x_n, C_k)\ \textbf{Probability}\ (x_2, \ldots\ldots, x_n, C_k)$$

$$= \textbf{Probability}(x_1 \mid x_2, \ldots\ldots, x_n, C_k)\textbf{Probability}(x_2 \mid x_3, \ldots\ldots, x_n, C_k)\textbf{Probability}(x_3, \ldots\ldots, x_n, C_k)$$

$$= \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$= \textbf{Probability}\ (x_1 \mid x_2, \ldots, x_n, C_k)\ \textbf{Probability}\ (x_2 \mid x_3, \ldots, x_n, C_k) \ldots.\ \textbf{Probability}(x_{n-1} \mid x_n, C_k)\ \textbf{Probability}\ (x_n \mid C_k)\ \textbf{Probability}\ (C_k)$$

The model is built by using Support Vector Machines along with linear model and word embedding techniques to extract the exact structure and meaning of the words in the messages. The word embedding technique is a technique which has advantages over the traditional and most common bow technique (Bag of Words) which does not extract the structure of the words but only the frequency of the words. On top of that, we are using the pipeline which automates the workflow of the entire model along with the linear SVC model. The trained model is successful in generating multiple labels to the input text/ messages. We are using a threshold value to find the value/ probability for the labels to be assigned to the messages. The threshold value is calculated by using the mean of the probabilities for the classes when input is given to the model. The accuracy of the model is quite good with better results. Dataset generated in the form of CSV format:

sno	count	hate speech	offensive language	neither	class	tweet
0	3	0	0	3	2	!!! RT @mayasolovely: As a woman you shouldn't complain about cleaning up your house. & as a man you should always take the trash out...
1	3	0	3	0	1	!!!!! RT @mleew17: boy dats cold...tyga dwn bad for cuffin dat hoe in the 1st place!!
2	3	0	3	0	1	!!!!!!! RT @UrKindOfBrand Dawg!!!! RT @80sbaby4life: You ever fuck a bitch and she start to cry? You be confused as shit
3	3	0	2	1	1	!!!!!!!!! RT @C_G_Anderson: @viva_based she look like a tranny
4	6	0	6	0	1	!!!!!!!!!!!!! RT @ShenikaRoberts: The shit you hear about me might be true or it might be faker than the bitch who told it to ya 
5	3	1	2	0	1	!!!!!!!!!!!!!!!!!!"@T_Madison_x: The shit just blows me..claim you so faithful and down for somebody but still fucking with hoes! 😂😂😂"
6	3	0	3	0	1	!!!!!!"@__BrighterDays: I can not just sit up and HATE on another bitch.. I got too much shit going on!"

DOWNLOADING THE NLTK

The following snippet is an example used to download the NLTK library, which is used for pre-processing the text in the SMS dataset [15].

```
In [5]: nltk.download('stopwords')
        nltk.download('punkt')

[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\User\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\User\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!

Out[5]: True
```

Retrieval Number: E1985039520/2020©BEIESP
DOI: 10.35940/ijitee.E1985.039520
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

DATASET BEFORE PREPROCESSING

This is the sample dataset, and its labels format we are using in this work, and it contains text along with the special characters and numbers.

	ts	user	text	class	key
0	1.503303e+09	Balaemar	I have to pick up my car from the garage tomor...	1	1503303350U035FRUCY
1	1.503302e+09	Ragaenys	I won't be here tomorrow, one day vacation	2	1503301710U4A2FRAQ4
2	1.503296e+09	Myke	Missed connection in Zurich. Will be about 5-1...	1	1503296123U0MGNKETU
3	1.503260e+09	Drevyn	Enjoy!	8	1503259722U035B8PRU
4	1.503258e+09	Gaelralis	I am away for 2 weeks in iceland :flag-is:	2	1503258060U0HLAK1T6

DATASET AFTER PRE-PROCESSING

This is the dataset format after it is pre-processed when all the special characters and numbers are removed from the dataset.

```
df_messages.head(5)
```

```
[.] Number of training samples: 1719
```

Out[7]:

	ts	user	text	class	key
0	1.503303e+09	Balaemar	I have to pick up my car from the garage tomor...	1	1503303350U035FRUCY
1	1.503302e+09	Ragaenys	I won't be here tomorrow, one day vacation	2	1503301710U4A2FRAQ4
2	1.503296e+09	Myke	Missed connection in Zurich. Will be about 5-1...	1	1503296123U0MGNKETU
3	1.503260e+09	Drevyn	Enjoy!	8	1503259722U035B8PRU
4	1.503258e+09	Gaelralis	I am away for 2 weeks in iceland :flag-is:	2	1503258060U0HLAK1T6

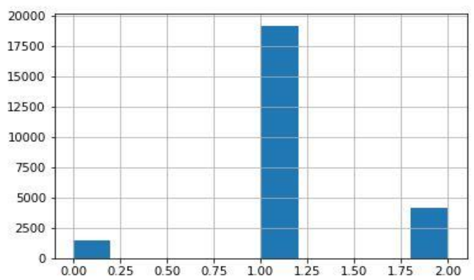
IV. ANALYSIS

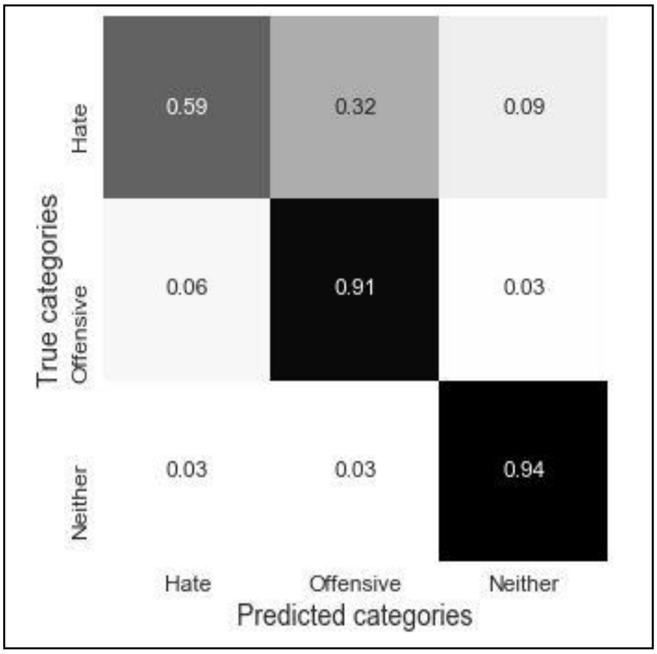
The random forest and Bayes Naïve classifier modelare trained only with a subset of the actual dataset. Only 18,000 short messages and their labels have been used for training due to lack of dataset. The model performs above expectations even when a subset of the whole dataset is used. The trained model is tested with the messages from the testing dataset. The generated labels for the texts are as follows:

```
0 - hate speech
1 - offensive  language
2 - neither
```

tweet = raw tweet text

```
In [8]: df['class'].hist()
Out[8]: <matplotlib.axes._subplots.AxesSubplot at 0xb5b0e80ba8>
```





V. CONCLUSION

In this paper, to contribute the a solution of an efficient novel approach for prediction of hate speech and offensive language on Twitter API using ML And n-gram features weighted with TF-IDF data exploration and determined comparative analysis of (LR)Logistic-Regression, (NB)Naïve-Bayes, (RF)Random forest and SVM on various sets of future values and model hyper parameters. The results showed that Logistic Regression performs better with the optimal n-gram range from 1 to 3 for the L2 normalization of TF-IDF. The model is built by using Support Vector Machines and Random Forest along with a linear model and word embedding techniques to extract the exact structure and meaning of the words in the messages. The word embedding technique is a technique which has advantages over the traditional and most common bow technique (Bag of Words) which does not extract the structure of the words but only the frequency of the words. On top of that, we are using the pipeline which automates the workflow of the entire model along with the linear SVM model. The trained model is successful in generating multiple labels to the input text/ messages on HSOL and using a threshold value to find the value/ probability for the labels to be assigned to the messages. The threshold value is calculated by using the mean of the probabilities for the classes when input is given to the model. The accuracy of the model is quite good with better results.In future work, to build a strong dictionary of HSOL paradigms that can be Moderated along with a uni-gram dictionary paradigm, to predict an efficient hateful and offensive online texts. We will make a quantitative and quality research study of the presence of hate speech among the different genders, age groups, and regions,etc. and the method of manually adding features can be automated. This work requires the users to manually add features which have a similar context to the most informative features of the provided dataset. The process would become much easier and efficient if the addition of features can be automated without any human involvement. This would make the process of training the model faster and optimal.

REFERENCES

1. G Zephoria.com, 2018. [Online]. Available: https://zephoria.com/top-5-valuable-facebook-statistics/. [Accessed: 22- Jun- 2018]. Twitter Usage Statistics Internet Live Stats, Internetlivestats.com, 2018.[Online].Available:http://www.internetlivestats.com/twitter-statistics/.
2. S. Hinduja and J. Patchin, "Bullying, Cyberbullying, and Suicide," Archives of Suicide Research, vol. 14, no. 3, pp. 206-221, 2010.
3. H. Wang, D. Can, F. Bar and S. Narayana, "A system for real-time Twitter sentiment analysis of 2012 U.S.presidential election cycle", Proc. ACL 2012 System Demostration, pp. 115-120, 2012.
4. O. Almatrafi, S. Parack and B. Chavan, "Application of location-based sentiment analysis using Twitter for identifying trends towards Indian general elections 2014". Proc. The 9th International Conference on Ubiquitous Information Management and Communication,2015
5. H. Zang, "The optimality of Naïve-Bayes", Proc. FLAIRS, 2004. C.D. Manning, P. Raghavan and H. Schütze, "Introduction to Information Retrieval", Cambridge University Press, pp. 234-265, 2008.
6. A. McCallum and K. Nigam, "A comparison of event models for Naive Bayes text classification", Proc. AAAI/ICML-98 Workshop on Learning for Text Categorization, pp. 41-48, 1998.
7. M. Schmidt, N. L. Roux and F. Bach, "Minimizing finite Sums with the Stochastic Average Gradient", 2002.
8. Y. LeCun, L. Bottou, G. Orr and K. Muller, "Efficient BackProp", Proc. In Neural Networks: Tricks of the trade 1998.
9. T. Wu, C. Lin and R. Weng, "Probality estimates for multi-class classification by pairwise coupling", Proc. JMLR-5, pp. 975-1005, 2004 "Support Vector Machines" [Online], http://scikit-learn.org/stable/modules/svm.html#svm-classification, Accessed Jan 2016.
10. P. Pang, L. Lee and S. Vaithyanathan, "Thumbs up? sentiment classification using machine learning techniques", Proc. ACL-02 conference on Empirical methods in natural language processing, vol.10, pp. 79-86, 2002.
11. P. Pang and L. Lee, "Opinion Mining and Sentiment Analysis. Foundation and Trends in Information Retrieval", vol. 2(1-2), pp.1-135, 2008.
12. E. Loper and S. Bird, "NLTK: the Natural Language Toolkit", Proc. ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics ,vol. 1,pp. 63-70, 2002.
13. Subbarayudu .y , Patil. S ,Ramyasree .B , Praveen Kumar. C ,Geetha.G Assort-EHR graph based semi-supervised classification algorithm for mining health records Journal of Advanced Research in Dynamical and Control Systems EID: 2-s2.0-85058439255.

AUTHORS PROFILE

Mrs. Gudurri Sulakshana, Assitant Professor of computer Science and engineering, Institute of Aeronautical Engineering, Hyderabad

Ms. R Siva jyothi, Asst professor of computer science and engineering ,ksrm college of engineering, kadapa

Ms. Aluri Lakshmi, Assitant Professor of computer Science and engineering, Institute of Aeronautical Engineering, Hyderabad .