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 text 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.

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

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 Ti-Idf a Natural Language processing technique. The Ti-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 Ti-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.

Naïve Bayes classifier algorithm is a 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].

\[
\text{Probability (C}_k | x) = \frac{\text{Probability (C}_k) \times \text{Probability (x | C}_k)}{\text{Probability (x)}}
\]

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:

\[
\text{Probability (C}_k, x_1, x_2, \ldots, x_n | C_k) = \text{Probability (x_1 | x_2, \ldots, x_n | C_k)} \times \text{Probability (x_2 | x_3, \ldots, x_n | C_k)} \times \ldots \times \text{Probability (x_n | C_k)}
\]

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.
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.

IV. ANALYSIS
The random forest and Bayes Naïve classifier models are 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:

V. CONCLUSION
In this paper, to contribute 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 L^2 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


AUTHORS PROFILE

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