ISSN (e): 2250 – 3005 || Volume, 11 || Issue, 5|| May – 2021 ||
International Journal of Computational Engineering Research (IJCER)

Foul Language Comment Classification

Urvi Tambe[1], Anvita Thingalaya[2], Saloni Vichare[3],
Guide:- Priya Parate[4]
[1,2,3,4]*Department of Computer Engineering, MCT's Rajiv Gandhi Institute of Technology, Mumbai, Maharashtra, India.*

Abstract- *With the advanced use of smartphones and also the Internet available to the majority of people at the tip of their fingertips, social media such as Twitter, Facebook, Instagram, etc. became accessible to people of all age groups right from children to adults. Children as young as 7-8 years now have access to social media. Hence, we must create a secure space online for people of all age groups. This can be why we chose our project "Foul Language Comment Classification". Our project aims to create online cyberspace a secure environment for people so that all people can interact with one another without any concern of being a theme of trolls or abusers .This will be achieved by ensuring that abusive, foul, filthy language gets censored or flagged off so that it doesn't reach the people. The employment of social media is the most typical trend among the activities of today's people. Social networking sites such as Twitter, Instagram, etc. provide a means of communication to today's youth. They use social media to gather more information from their friends and followers. The vastness of social media sites ensures that not all of them provide an honest environment for youngsters. In the above scenarios, there is an increase of negative and destructive thoughts on the young and impressionable minds of teenagers. This increase could lead to frustration, depression, and an oversized change in their behavior.*

Our project "Foul Language Comment Classification" aims to spot the toxicity of statements then flag them as various levels of toxicity. Fast text algorithm is employed for this purpose.

Keywords: *FastText Algorithm, Machine Learning, Tensorflow, Convolutional Neural Network, Tokenization, Word Embedding, Text Classification*

Date of Submission: 16-05-2021 Date of acceptance: 31-05-2021

I. INTRODUCTION

Today's world has made it very easy for trollers and abusers to bully and harass people online under the garment of invisibility provided by social media. This ends up in a rise in instances of cyberbullying and harassment which might adversely impact an individual's psychology and cause him/her mental stress and trauma. This sort of toxic behavior has become quite common nowadays as everyone has access to the internet and social media at the tip of their fingers. These sort of abusive people make the web virtual space very unsafe for all the people around them. This results in people especially children not having the ability to converse freely online.

In a survey where over 6,000 people participated, within the people of 10-18-year-olds, it had been found that almost half i.e. 50% of kids had experienced a minimum of some form of harassment and cyberbullying within the online forums. Amongst the 11 European countries included within the report, 44% of youngsters who had been cyberbullied before lockdown said it happened even more during the lockdown. Teachers report that cyberbullying is their number 1 safety concern in their classrooms One out of three youngsters in 30 countries said they have been a victim of online bullying, with one in five reporting having skipped school thanks to cyberbullying and violence, in an exceedingly new poll released by UNICEF and therefore the UN Special Representative of the Secretary-General (SRSG) on Violence against Children. In our project "Foul Language Comment Classification" we aim to classify the toxicity in the online comments by multi-labeling the toxicity within six different labels. The labels included are "toxic", "severe toxic", "obscene", "threat" , "identity-hate" and "insult". These labels show the varying levels of toxicity that may be shown online. This allows us to properly classify the toxicity in the statements given online in keeping with the labels mentioned above.

Our classification model works by assigning weights according to the probability of the toxicity labels. This approach makes sure that one comment may belong to multiple labels of toxicity. The FastText algorithm is used for this purpose. Convolutional Neural Network or CNN is used to analyze the sentences. The CNN will correctly analyze the sentence and accordingly, it will classify the sentence into one or more of one of the labels. The Neural Networks can only work if the data is numeric only. Therefore, the sentences that need to be analyzed must be first converted into some form of data that is numeric so that the Neural Network can work on it.

.

II. LITERATURE SURVEY

[1]Navoneel Chakrabarty: "A machine learning approach to comment toxicity approach"
The paper introduces the Machine Learning approach with Natural Language Processing to work out the various kinds of toxicity using 6 headed Machine Learning TF-IDF model. It had been observed that the model gives a mean validation accuracy of about 98.08% and absolute validation accuracy of 91.61%.
[2] Kevin Kheu, Neha Narwhal:"Detecting and Classifying Toxic Comment"
During this paper, Support Vector Machine(SVM), Long-Short Term Networks(LSTM), Convolutional Neural Network(CNN) and, Multilayer Perceptron(MLP) methods are done at the word and character level embedding for identifying the toxicity and was observed that on word-level classifications best results were obtained with LSTM model and for character level classification CNN model works the best.
[3] Mujahed A. Saif, Alexander N. Medvedev, Maxim A. Medvedev, Todorka Atanasova: "Classification of online toxic comments using the logistic regression and neural networks models"
In this paper 4 models were proposed: Logistic Regression model and three neural network models: Convolutional neural network (CNN), Long Short term memory (LSTM) with RNN, and Conv + LSTM and was concluded that the convolutional model works better than RNN models and also the best results were obtained when both Conv+ LSTM (2 LSTM layers and 4 Convolutional Layers) model worked together.
[4] Sindhu Abro1, Sarang Shaikh2, Zahid Hussain Khand3, Zafar Ali4, Sajid Khan5, Ghulam Mujtaba6: "Automatic Hate Speech Detection using Machine Learning: A Comparative Study"
The paper employed 3 different feature engineering techniques (Bigram with TFIDF, word2vec, and doc2vec) and eight machine learning algorithms (Naïve Bayes, Support Vector Machine, K Nearest Neighbor, Decision Tree, Random Forest, Adaptive Boosting, Multilayer Perceptron, and Logistic Regression) to classify comments and was noticed that Bigram with TFIDF maintains the sequence of words and then outperforms Word2vec and Doc2vec. Moreover, Support Vector Machine and Random Forest algorithms showed better results compared to the rest with KNN giving the least accuracy.
[5] Omar Sharif Iand Mohammed Moshiul Hoque: "Identification and Classification of Textual aggression in Social Media: Resource creation and evaluation"
In this paper, the aggressiveness of the Bengali text is finished into various categories like religious, gendered, verbal, and political aggression class. Several classification algorithms such as LR, RF, SVM, CNN, and BiLSTM are implemented and also the developed model gained maximum accuracy when CNN and BiLSTM were combined.
[6] Theodora Chu, Kylie Jue, Max Wang:" Comment Abuse Classification with Deep Learning"
This paper compared three models namely, a recurrent neural network (RNN) with a long-short term memory (LSTM) and word embeddings, a convolutional neural network (CNN) with word embeddings, and a CNN with character embeddings were proposed, and was concluded that CNN with character level embedding works better than CNN with word embedding or RNN with LSTM. Moreover, the CNN models work comparatively faster than the RNN model. The paper also states that RNN with LSTM outperforms RNN with GRU.
[7] Pritom Mojumder, Mahmudul Hasan, Md. Faruque Hossain, and K.MAzharul Hasan: "A Study of fastText Word Embedding Effects in Document Classification in Bangla Language"
In this paper, the Fast text embedding technique was brought into the limelight for general text classification and was observed that with fastText word embedding significant performance are often gained without some preprocessing like lemmatization, stemming, and others.
From the above papers, it may be concluded that between CNN and RNN approach CNN works best and faster for text sentiment classification and character level embedding while requires a lots of training but still gives the most effective results as compared to word level or sentence level embedding. Some research papers experimented with different word embedding methodology in detailed comparison between the GLOVE model, Word2Vec, Doc2vec, and TF-IDF model. Out of which word2vec and glove model have shown effective results, but didn't label the typing mistakes and slangs. But with the newest technique named Fast text word embedding works better even with slang and typos.

III. METHODOLOGY

Our project flow works in such a way that the sentences are classified into the six labels according to their toxicity. The FastText algorithm is used in our project because this is an algorithm which can be used even when there are errors in spellings or some slang words are used. This is a very useful feature as today's youth mostly use shortcuts in spellings.

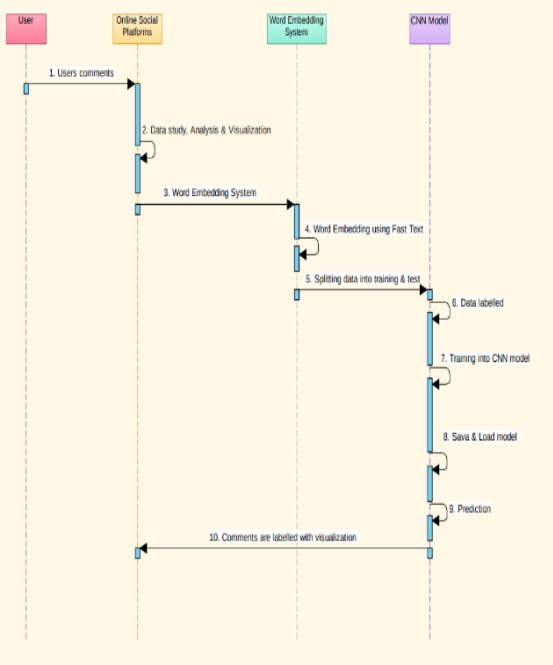

Figure 1: Sequence Diagram

The above figure illustrates the basic flow of our project. The steps are explained in detail as shown below:

[1]Pre-processing of data:

This is the first step which entails cleaning of the dataset. This step includes the tokenization of the sentences. After this common bigrams of the words are found and grouped together .This is followed by the lemmatization of the words in which all the words are converted to their root form.

[2]FastText Word Embedding:

The dataset has then been split into a traning dataset and testing dataset. The ratio of the split is 70:30.The FastText algorithm is then provided with words and sentences which are called word vectors. The FastText algorithm will then learn the word representations.

[3]CNN:

The embedded words are fed to the firs layer of CNN. The activation function used in this step is called RELU. Our Convolutional Neural Network(CNN) has used a total of 4 convolution layers having 32 filters. These filters have a size of (1 X 300), (2 X 300), (3 X 300), (5 X 300). After this feature maps having size of (200 X 1), (199 X 1), (198 X 1), (196 X 1) are produced. After the convolution is complete, the most significant features are captured using the pooling operation. In the ouput layer, the activation function used is the soft-max layer which generates the probabilities of the six different labels of toxicity. As there are a total of 6 labels, there are a total of 6 nodes for each comment label in the output layer.

This process is followed by the compilation of of the above CNN model, which is done using binary cross entropy loss function. Once this process is completed, this model architecture can be saved in the form of a .yaml file which can then be simply loaded and used for further testing. This process is known as transfer learning.

[4]Prediction:

The model which is saved is used as a checkpoint. This means that the model will not need to be reloaded each and every time the classification is to be done. Thus, future predictions can be easily done using the saved models. The prediction is done using un-labelled data so that correct prediction can be given. The labels are resolved into binary values of 0 and 1 which is probabilistic in nature.

[5]Backend:

The backend connection is done using flask. The FastText algorithm and CNN model is saved in a checkpoint format i.e. in a .yaml format. The reason behind saving the model is that the prediction can be carried out without having to load the model all over again. It is also useful as words and sentences which are not seen by

the model before can also be predicted. The app.py file can now be run by using the command prompt which will open our web app user interface on the localhost.
Now, the user has to input any text or sentence in the text box area provided. Once the sentence is submitted, the level of toxicity of the statements are shown through the probabilities of the six labels.

IV. DATASET

The Wikipedia Talk Page Dataset is prepared by Jigsaw. This dataset is now publicly available on the Kaggle website. This dataset consists of 159571 sentences which correspond to the 6 labels namely "toxic", "severe toxic", "identity-hate", "threat", "insult" and "obscene".

V. RESULT

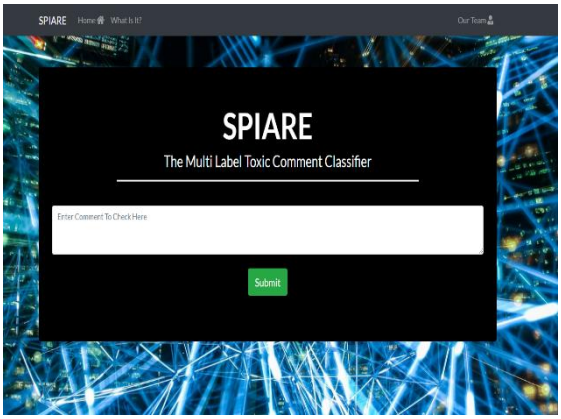

Figure 2: Home Page

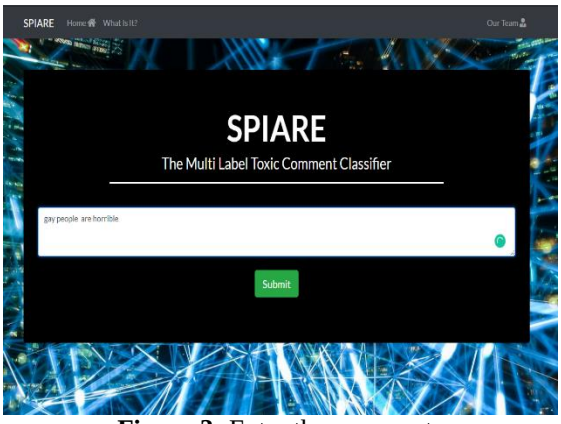

Figure 3: Enter the comments

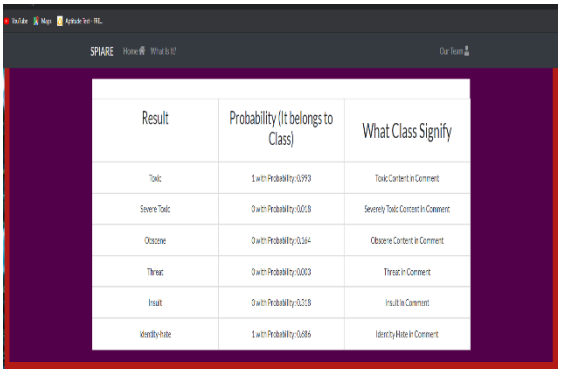

Figure 4: Probability of Toxicity

As illustrated in the Figure 2, the home page consists of a text box where the comments can be entered. It also consists of a "Submit" button so that the toxicity of the comments can be predicted.
Figure 3 illustrates the comments being typed out int the text box.
Figure 4 shows the probability of toxicity of the comments.

The statement used is “Gay people are horrible.” Our model has correctly predicted the above comment as “Toxic” with a probability of 0.993. The model has also classified it as an “Identity-hate” with a probability of 0.686. This is a correct prediction as the person is being threatened for his identity.

VI. CONCLUSION

This project uses Fasttext algorithm and CNN to detect toxicity in online platforms. It is very useful in detecting foul, abusive, toxic language being used nowadays in social media. We have implemented the project using Fasttext algorithm as it suited the needs of our project perfectly. Fasttext algorithm is one of the most effective algorithms in determining the toxicity of statements. It is also very accurate when it comes to dealing with all the slangs, typos, jargons, short forms of words, etc. This is a very important feature as many people especially the young generation likes to write short forms and slangs instead of properly typing all the sentences in the correct grammatical order. The model especially outperforms when there is high variance in data and dataset size is large.

REFERENCES

[1]. Pritom Mojumder, Mahmudul Hasan, Md. Faruque Hossain, and K.MAzharul Hasan**:** "A Study of fastText Word Embedding Effects in Document Classification in Bangla Language**”.** International Conference on Cyber Security and Computer Science, Dhaka, February 2020

[2]. Theodora Chu, Kylie Jue, Max Wang:” Comment Abuse Classification with Deep Learning” ,Stanford University, 2017

[3]. Omar Sharif and Mohammed Moshiul Hoque: "Identification and Classification of Textual aggression in Social Media: Resource creation and evaluation", April 2021

[4]. Sindhu Abro, Sarang Shaikh, Zahid Hussain Khand3, Zafar Ali4, Sajid Khan, Ghulam Mujtaba6: "Automatic Hate Speech Detection using Machine Learning: A Comparative Study", International Journal Of advanced Computer Science and Applications, January 2020

[5]. Mujahed A. Saif, Alexander N. Medvedev, Maxim A. Medvedev, Todorka Atanasova: "Classification of online toxic comments using the logistic regression and neural networks models" ,Proceedings of the 44th International Conference on Applications of Mathematics in Engineering and Economics, December 2018 https://aip.scitation.org/doi/pdf/10.1063/1.5082126

[6]. Kevin Kheu, Neha Narwhal: "Detecting and Classifying Toxic Comment" https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1184/reports/6837517.pdf

[7]. Navoneel Chakrabarty: "A machine learning approach to comment toxicity approach" https://arxiv.org/ftp/arxiv/papers/1903/1903.06765.pdf

[8]. Karthik Diankar, Roi Riechart, Henry Lieberman, Modeling the Detection of Textual Cyberbullying., Massachusetts Institute of Technology, Cambridge MA 02139 USA.,2011.

[9]. Manav Kohli, Emily Kuehler and John Palowitch. Paying attention to toxic comments, Stanford University,2017

[10]. S. V. Georgakopoulus, A. G. Vrahatis, S. K. Tasoulis, V. P. Plagianakos. Convolutional Neural Networks for Toxic Comment Classification arXiv:1802.099574v1 [cs.CL] , 27 Feb 2018

Urvi Tambe, et. al. "Foul Language Comment Classification." *International Journal of Computational Engineering Research (IJCER*), vol. 11, no.5, 2021, pp 15-19.