ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 11, Issue 1, January 2024

Toxic Chat Detection using Deep Learning

[1] Pratik Davange, [2] Pratik Chaudhari, [3] ST Patil, [4] Arwa Bhojawala

[1] [2] [3] [4] Department of Computer Engineering, Vishwakarma Institute of Technology (VIT), Pune, Maharashtra, India
Corresponding Author Email: [1] pratik.davange21@vit.edu, [2] pratik.chaudhari21@vit.edu, [3] patil.st@vit.edu, [4] arwa.bhojawala21@vit.edu

Abstract— ***The value of remaining respectful in an online discussion cannot be emphasised. "Wilful and repetitive harm perpetrated through the medium of electronic messaging" is characterised as cyberbullying. It includes delivering derogatory, threatening, and/or sexually explicit messages and photos to targets using websites, blogs, instant messaging, chat rooms, e-mail, cell phones, websites, and personal online profiles. As a result, identifying and removing harmful communication from public forums is a critical task that human moderators are incapable of performing. Body shaming may now be seen on every social media site. People are willing to go to any extent to humiliate others. This type of issue affects our society's females more significantly. We may make use of the model.***

Keywords — Hate Speech, Cyber bullying, Social Threats, Machine Learning, Text Vectorization and Tokenization.

I. INTRODUCTION

Recent advancement in NLP has resulted in the release of machine based toxic chat detection. On the one hand, individuals are always developing new types of nasty statements that are easily recognised by humans but not by robots. So, we want to use our expertise of ML and DL to categorise the poisonous comment into categories such as toxic, severe toxic, Threat, and insult, and so on, and to create a flexible filter that will act on that toxic comment. The use of hostile or insulting language on the internet has increased dramatically in recent years, and the problem is now flourishing. Toxic online remarks have even led in real-life violence in several situations, ranging from religious nationalism in Myanmar to neo-Nazi propaganda in the United States. Social media sites, which rely on thousands of human reviewers, are failing to keep up with the growing volume of damaging content. According to a 2019 study, Facebook moderators are at danger of developing PTSD as a result of repeated exposure to such distressing content. Outsourcing this effort to machine learning can assist in managing the increasing volumes of hazardous content while reducing human exposure to it. Indeed, several tech behemoths have been incorporating algorithms into their content management for many years.

This Project aims to develop a Deep Learning model which will analyse the text data on six factors including toxicity, severe toxicity, threat, insult, obscene and identity hate. As use of social media is increasing exponentially, the use of this DL model in the backend of any application will improve user experience and decrease the hate spread across the platform. The dataset used in the project is publicly available on Kaggle. The dataset contains text messages between people and six target classes as mentioned above.

II. LITERATURE SURVEY

For this project many research papers were reviewed and the relevant information is given below.

A paper [1] which was published in Jan 2022 named “Understanding and identifying the use of emotes in toxic chat on Twitch” really helped the research to understand different variations of common expressions which are mixture of text and emotes. The data used in this research paper was collected from famous streaming platform Twitch. The Deep Neural Network used in this research was based upon the Bidirectional LSTM Layer. Based on the dataset, a neural network classifier was built which identified visual toxic chat that would otherwise be undetected through traditional methods and caught an additional 1.3% examples of toxic chat out of 15 million chat utterances.

Another paper [2] was reviewed named “Machine learning and semantic analysis of in-game chat for cyberbullying.” Since researchers are typically obliged to rely on survey data where victims and perpetrators self-report their opinions, a significant challenge with cyberbullying research is the lack of data. The in-game conversation data from one of the most well-liked online multiplayer games is continually collected using an autonomous data collection system demonstrated in this paper: Game of Tanks. The data collected was used for sentimental analysis and a classifier was developed to classify if a comment is toxic or not.

In 2022, A paper [3] named “An Automated Toxicity Classification on Social Media Using LSTM and Word Embedding” was published. This research paper focuses on automated identification of toxicity in texts. The deep learning model used for this research is based on LSTM with word embeddings generated by the Bidirectional Encoder Representations from Transformers (BERT). The binary categorization of comments using LSTM with BERT word embeddings produced satisfactory accuracy of 94% and an F1-score of 0.89. (Toxic and nontoxic).

Another paper [4] from same year named "Deep-Learning-Based Automated Scoring for the Severity of Toxic Comments Using Electra" was published. In this paper, a deep-learning-based natural language processing technique is proposed using ELECTRA to automatically score the toxicity of a comment. Three head layers are implemented separately: multi-layer perceptron, convolutional neural network, and attention.

III. DATA COLLECTION

The dataset used for this study is publicly available on Kaggle. Kaggle is a online community of data scientists and machine learning enthusiasts. This website contains millions of datasets which can be used for research. The dataset was uploaded on this website by Conversation AI team at Jigsaw.

IV. DATA PREPROCESSING

First step in any standard Deep Learning method is Data Pre-processing. As the data used in this study is text based, concepts of Natural Language Processing are used.

The data contains multiple combinations of a word which was meant to be same, such as use of word "amerikan" instead of "american". So, converting various such combinations into single word will decrease the complexity of the model. The following combinations were replaced by their respective word.

Various Combinations	Replaced with
'amerikan'	american
'f\-ing', 'f\.u\.', 'f###', ' fu ', 'f@ck', 'f u c k', 'f uck', 'f ck'	fuck
'biatch', 'bi**h', 'bytch', 'b i t c h', 'b!tch', 'bi+ch', 'l3itch'	bitch
'sucks', '5uck', 's u c k'	suck
'bullsh*t', 'bull\$hit'	bull shit
'returd', 'retad', 'retard', 'wiktard', 'wikitud'	retard
'dumbass', 'dubass'	dumb ass
' motha ', ' motha f', ' mother f', 'motherucker'	mother fucker

To perform text analysis on data, removing stop words is one of the most important steps. Stop words are words that are filtered out of a stop list either before or after natural language data processing because they are unimportant. For that purpose, spacy [12] library is used. Spacy is a open source library used for natural language processing.

Next step is to vectorize the data, as Deep Neural Network can only be trained on numbers. Text Vectorization is process in which a sentence or large number of words are represented as a vector. In this process, each word is tokenized with number and the sentence formed by words is vector containing all the tokenized numbers. Each integer reflects the embedding of different properties. For this purpose, Keras layer from tensorflow [11] has an in-built function named TextVectorization is used.

V. MATHEMATICAL EXPLANATION OF MODELS

a. tanh activation function

$$f(x) = (e^{x} - e^{(-x)})/(e^{x} + e^{(-x)})$$

tanh a hyperbolic tangent function which is used to scale a number into a range of (-1,1).

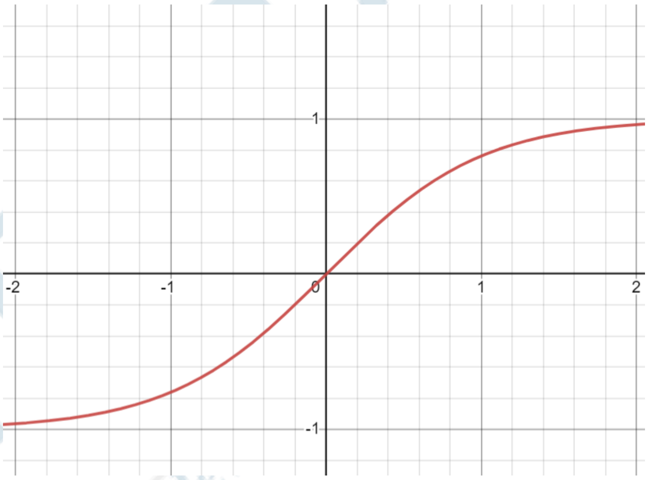


Figure 1

b. Relu activation function

$$f(x) = max(0, x)$$

The rectified linear activation function, often known as ReLU, is a non-linear or piecewise linear function that, if the input is positive, outputs the input directly; if not, it outputs zero.

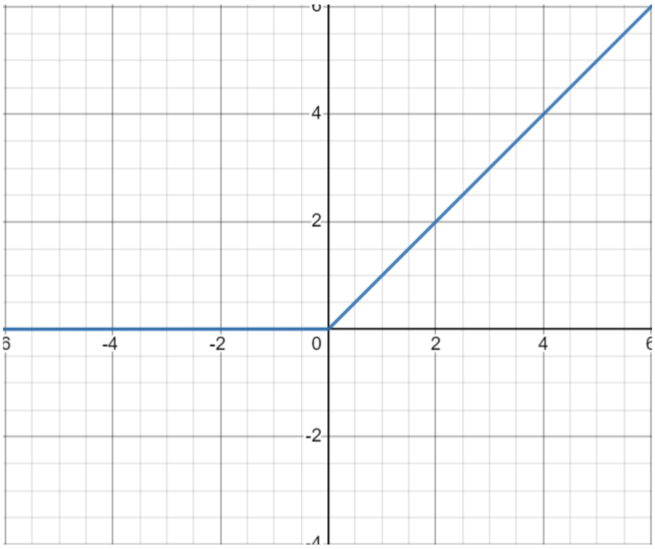


Figure 2

c. Sigmoid function

$$f(x) = 1/(1 + e^{(-x)})$$

The sigmoid function is used to scale a number into a range (0,1).

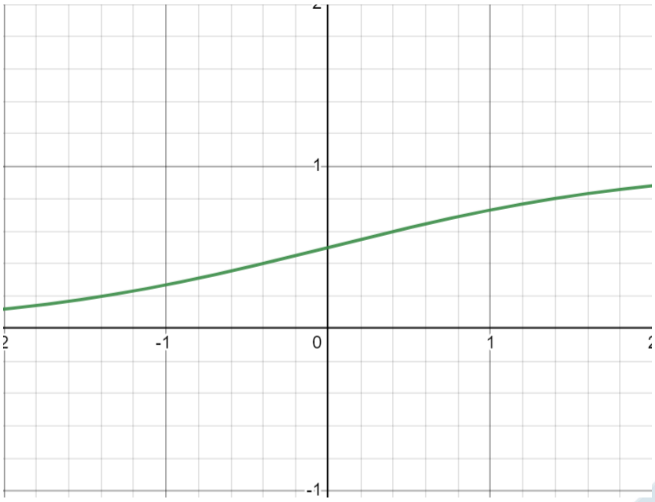


Figure 3

VI. METHODOLOGY

ML is the process of teaching a computer to use its prior knowledge to address a problem that has been presented to it. By modelling how people think and learn, the DL subfield of ML enables computers to automatically extract, analyze, and comprehend the meaningful information from the raw data. The method used to extract the characteristics from the data that the classifier uses is the main distinction between ML and DL.

The use of AI for predicting diseases can be broken down into three steps: data collection, modelling and prediction. As data is pre-processed, next step is to create a Deep Neural Network.

The data is divided into two parts i.e.,1) Train data, 2) Test data. Train data contains 70% of the original data and Test data contains remaining 30%. The Test data is reserved because it will be used further for evaluating the Deep Neural Network.

For this study, Sequential model is used. For a simple stack of layers, where each layer has precisely one input tensor and one output tensor, a sequential approach is acceptable. For importing the Sequential API, TensorFlow is used. Now certain layers from TensorFlow are used to construct Deep Neural Network.

A. Embedding Layer

The Embedding layer looks for the embedding vector for each word-index using the vocabulary that has been integer-encoded. As the model is trained, these vectors are learnt. The output array gains a dimension thanks to the vectors. The embedding layer is added with 20,000 input layer dimension and 32 output layer dimension.

B. LSTM

In time series and sequence data, an LSTM layer learns long-term relationships between time steps. The hidden state, sometimes referred to as the output state, and the cell state make up the layer's state. The output of the LSTM layer for time step t is contained in the hidden state.

C. Bidirectional Layer

Two hidden layers connected in opposing directions to the same output are referred to as bidirectional recurrent layers. Due to this generative deep learning, the output layer simultaneously receives input from previous or backwards states and future or forward state. The combination of bidirectional and LSTM is added with activation function as tanh.

D. Dense Layer

Each neuron in the basic layer of neurons known as the dense layer receives information from every cell in the layer below it. Dense layer works as Feature Extractor. Three dense layers are added with activation function ‘relu’. And last dense layer is added with activation function ‘sigmoid’ to scale the output in the range (0-1).

Model Summary:

Layer (Type)	Output shape	Parameter
Embedding	(None, None, 32)	6400032
Bidirectional	(None,64)	16640
dense	(None,128)	8320
dense_1	(None,256)	33024
dense_2	(None,128)	32896
dense_3	(None,6)	774

Total params: 6,491,686
Trainable params: 6,491,686
Non-trainable params: 0

As the Deep neural network is constructed, the train data is used to train the date for 150 epochs. An epoch is the total number of iterations required to train the machine learning model using all the training data at once. It is measured in cycles. The number of trips a training dataset makes around an algorithm is another way to define an epoch. When the data set has made both forward and backward passes, one pass is counted.

VII. SYSTEM OVERVIEW

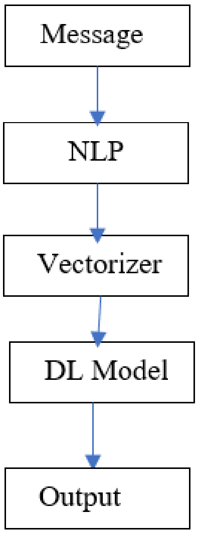


Figure 4

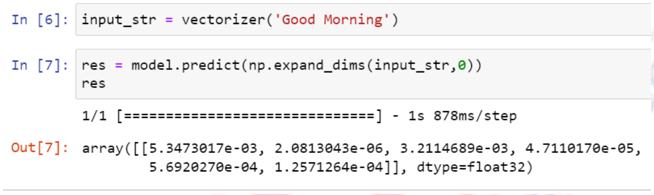
The output of the DL model will be vector of size 6. The output vector will have the probabilities of six factors i.e., Toxic, Severe toxic, threat, insult, obscene, Personality hate respectively.

VIII. RESULT AND DISCUSSION

After training the model for 150 epochs, the accuracy of model on test data is 82.06%, Precision is 81.08% and Recall is 78.23%.

Some predictions of the model:

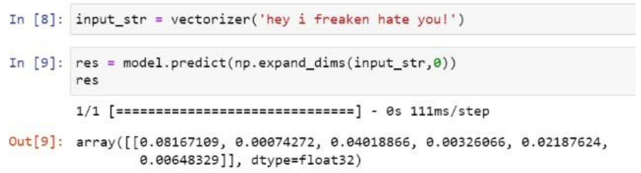
i. "Good Morning"



```
In [6]:  input_str = vectorizer('Good Morning')

In [7]:  res = model.predict(np.expand_dims(input_str,0))
         res

         1/1 [==============================] - 1s 878ms/step

Out[7]:  array([[5.3473017e-03, 2.0813043e-06, 3.2114689e-03, 4.7110170e-05,
                 5.6920270e-04, 1.2571264e-04]], dtype=float32)
```

Figure 5

ii. "Hey i freaken hate you!"



```
In [8]:  input_str = vectorizer('hey i freaken hate you!')

In [9]:  res = model.predict(np.expand_dims(input_str,0))
         res

         1/1 [==============================] - 0s 111ms/step

Out[9]:  array([[0.08167109, 0.00074272, 0.04018866, 0.00326066, 0.02187624,
                 0.00648329]], dtype=float32)
```

Figure 6

IX. LIMITATIONS

Many times, unknowingly there is Data leakage. Sometimes, the accuracy of ML model is not satisfactory due to shortage of data. And supposing we get good accuracy then it is also not enough.

The model will show low accuracy when the input text contains words which are not present in train data. The model cannot predict toxicity of a text referencing to offensive meaning.

X. CONCLUSION

In this study, A Deep learning model was developed using Sequential model and 4 types of layers i.e., LSTM, Bidirectional, Dense, Embedding. The accuracy of this deep neural network was 82.06%. The use of words like "fuck", "bastard" highly affect the output probability of factor 'Toxic'. The words like "Kill" has high weight for the factor "Threat"

The model can be used in backend of various social media applications to decrease the overall spread hate across platform and provide a better experience to users.

XI. FUTURE SCOPE

Using more advanced Machine Learning techniques, such as deep learning to improve the performance and hence improving the accuracy of the result as well as the model. Future research may investigate a classifier that recognizes reference terms by utilizing complex language models like BERT [10] and XLNet [11].

Google created Bidirectional Encoder Representations from Transformers (BERT [10]), a transformer-based machine learning approach for pre-training natural language processing (NLP). BERT was developed and released by Google employees Jacob Devlin and his team in 2018.

When it comes to data collection, data used in this study is static, so the model does not perform well for a word which is not present in Train data. Future work may include addition of a dynamic database and continuous model training.

ACKNOWLEDGEMENT

We would like to express our deep gratitude to Prof. Anant Kaulage sir for his kind direction, ongoing assistance, and inspiring encouragement in getting the project done. We also thank the project coordinators for their advice, constructive criticism, and assistance in setting up the facilities required to complete the project work effectively. Additionally, I want to express my thanks to my team members for their assistance in formatting this work. I also want to express my gratitude to all of my other colleagues for their constructive comments on this research. Finally, I'd want to thank my parents for their help with the research.

REFERENCES

[1] Jaeheon Kim, Donghee Yvette Wohn, Meeyoung Cha, "Understanding and identifying the use of emotes in toxic chat on Twitch", 2022.

[2] Shane Murnion, William J.Buchanan, Adrian Smales Gordon Russell, "Machine learning and semantic analysis of in-game chat for cyberbullying", July 2018.

[3] Ahmad Alsharef, Karan Aggarwal, Sonia, Deepika Koundal, Hashem Alyami, and Darine Ameyed, “An Automated Toxicity Classification on Social Media Using LSTM and Word Embedding”, Feb 2022.

[4] Tiancong Zang, “Deep-Learning-Based Automated Scoring for the Severity of Toxic Comments Using Electra”, July 2022.

[5] Ellen Riloff, Ashequl Qadir, Prafulla Surve, Lalindra De Silva, Nathan Gilbert, Ruihong Huang, “Sarcasm as contrast between a positive sentiment and negative situation”, in: Proc. of the EMNLP, 2013.

[6] Walaa Medhat, Ahmed Hassan, Hoda Korashy, “Sentiment analysis algorithms and applications: A survey”, Dec 2014.

[7] Xing Fang, Justin Zhan, “Sentiment analysis using product review data”, 2015.

[8] Zulfadzli Drus, Haliyana Khalid, “Sentiment Analysis in Social Media and Its Application: Systematic Literature Review”, 2019.

[9] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, Quoc V Le, XLNet: Generalized autoregressive pretraining for language understanding, 2019, arXiv preprint arXiv:1906.08237.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2018, arXiv preprint arXiv:1810.04805.

[11] Tensorflow documentation: https://www.tensorflow.org/api_docs

[12] Spacy documentation: https://spacy.io/api/doc