International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 07 | July 2021 www.irjet.net p-ISSN: 2395-0072

Profanity Filter and Safe Chat Application using Deep Learning

Sanjana Kumar[1], Srikrishna Veturi[2], Varun Sreedhar[3]

[123] *UG Students Department of Computer Engineering, SIES Graduate School of Technology, University of Mumbai, Mumbai, Maharashtra, India*

Abstract - *Internet is a place where freedom of speech is thoroughly enjoyed but, in many cases, it is exploited. In the current pandemic situation, internet and especially chat applications have become a necessity instead of a luxury. Many children, at their tender age are introduced to the internet and are prone to verbal abuse or exposed to illegitimate content. There is absolutely no check or regulation to prevent children and other concerned people from toxic content. It is this nature of the internet which inevitably brings rise to social stigmas such as cyberbullying and cybercrime which affects the mental state of children as well as adults. Since internet-based applications are a boon for humanity and a necessity, boycotting them cannot be a solution. The better approach to this problem would be to regulate the content that is flowing through the internet according to the receiver's consent.*

Our solution focuses on developing a filter for chat application where each message is checked for toxicity using Deep Learning and Natual Language Processing, we are focusing on understanding the context of each message. This is done using a custom TensorFlow deep learning model utilizing other features like synonym detection. Each message is passed through this filter to check for toxicity, if a message is found toxic, it is blocked at the sender's end and is not transmitted.

Key Words: **Deep Learning, Natural Language Processing, TensorFlow, Toxicity Detection, Cyber Bullying**

1.INTRODUCTION

1.1 Motivation

With the internet being used the most in today's environment, everyone has sometime or the other experiences cyber bullying or online hate. This has become a problem that needs to be addressed with a huge amount of concern. No human being should be unnecessarily burdened by unwanted toxic content, which has been quite prevalent in the modern society. Children, who are naïve and innocuous should not be burdened by trauma or fear instead of being instilled with curiosity and aspirations. Every human being has the right to be spoken with respect and courtesy and there is a need to ensure that this right is being accessed by everyone on the internet where the exploitation of the freedom of speech has become a new norm. This step would potentially cut off cybercrimes at it' and make internet much healthier.

1.2 Proposed System

There are two parts to our proposed system, one is the profanity filter and second is the chat application which serves as a proof of concept of the product. The filter, which is the core of the project is built in such a way that it can be integrated into any of the existing chat platforms with minimal change in the codebase. The filter is an API (Application Programming Interface) based response system built using Python Flask which accepts a message as an input and using the custom TensorFlow model and other algorithms like synonym matching, classifies the message into one of two categories, toxic and non-toxic and finally returning this classification as a response. The response can be later used by the chat platform to decide whether to block the message or to transmit it through the chat server.

1.3 Objectives

The objective of this project is to develop a well-functioning profanity filter which can be seamlessly integrated with any new or existing chat applications and protect end users from toxic comment and discourage those who adulterate the internet with any inappropriate content thereby making the internet much safer place for everyone.

2. LITERATURE REVIEW

In recent years, the prevalence of profanity has been daunting. This has resulted in a compelling requirement of a robust technical model to keep such profanity in check.

Perusing various research papers makes it evident that a perfect blend between Natural Language Processing and Neural Networks would assist to build the required technical model. In recent years, a powerful Neural Network architecture coupled with LSTM approach have shown to be a viable solution.

A study [1] showed how Deep Learning is highly effective to solve such problems. An impactful combination of Convolutional Neural Networks with Bidirectional LSTM (Long Short-Term Memory) helps in constructing a strong novel architecture which not only analyzes the local features but also analyzes the global semantics. It was proposed that the above architecture would be imbibed in

search engines and messengers as these are the platforms that witness profanity to a major extent. The purpose of the architecture was quite definitive as the model was trained with a concrete purpose to flag textual contents that are (a) rude, discourteous or exhibit lack of respect towards certain individuals or group of individuals (b) to cause or capable of causing harm (to oneself or others) (c) related to an activity which is illegal as per the laws of the country or (d) has extreme violence. It was empirically laid that the proposed architecture displayed statistical supremacy when word embedding techniques were adopted.

Another similar research [2] employed word embedding using the FastTextModel. The character vector strings were formed by adding the substrings of the character vector. The information regarding the morphology was used for training. This method enabled the model to detect transformed profanity with a good precision and made it a point to regard morphology as an important metric along with the semantics.

Yet another research [3] proposed a method pertinent to restrict cyberbullying. Their model was evaluated by two classification models which were the SVM (Support Vector Machine) and Neural Networks. Both models were assisted by TFIDF (Term Frequency Inverse Document Frequency) and sentiment analysis algorithms for feature extraction purpose. The Neural Network classifier turned out to be the better performer. It was also suggested by the research that for a large data set a Deep Learning model would be more effective.

3. SYSTEM WORKING

The System can be broken down into two different entities, the chat application and the toxicity filter that is the core of the toxicity detection which can also be integrated into any of the existing chat platforms with minimal changes in their own codebases. The toxicity detection filter works as an API which takes in the message as input and send a simple json as a response which can either have “Offensive” or “Not Offensive” in the body. The API is developed using python for easy compatibility with the machine learning model. First the API preprocesses the text by lowering all the text, removing any unimportant symbols, removing any stop words (words like “and” which do not contribute to the result) in the text then stemming the words which are later vectorized. To make sure the model does not only look at a certain group of words, but there is also a synonym detection before the count vectorization process. This could be understood as replacing a word that the machine learning architecture does not recognize with a word that the same architecture understands and has insights as to how the interpretation of the word being in a sentence should be.

Once preprocessed, the vector is passed into a custom neural network model built using TensorFlow which has an architecture of 500 neurons in the first layer (ReLu activation), 200 neurons in the second layer (ReLu activation) and finally a single neuron in the third and last layer (Sigmoid activation). The output of the model is a single numeric value which serves as an indicator to check if the message was interpreted as Toxic and Offensive or not. The output of the model is always a number between 0 and 1, if it is closer to 1, it is interpreted as a toxic message and should not be sent to the receiver, on the other hand is it is closer to 0, it is interpreted as a normal message and can be transmitted along the message server.

Chart 1 provides a picturized representation of the working of the profanity/toxicity filter API.

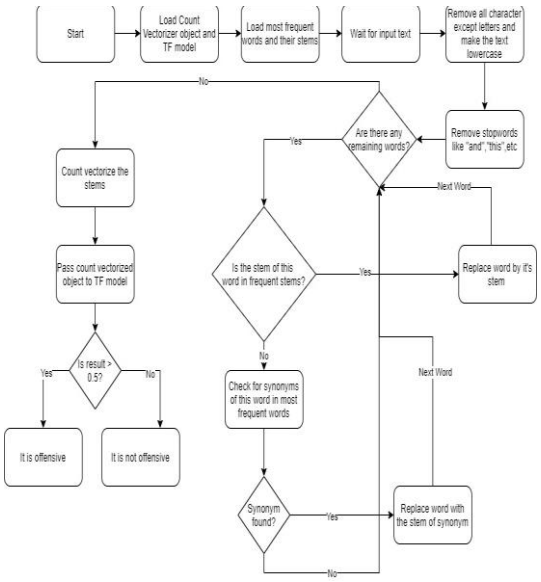

Chart 1: Flow Chart for working of the profanity/toxicity filter API

The chat application built using technologies like React JS, Node JS and Socket.io has features like adding a group conversation, turning the toxicity filter on and off (only available to adults) and a blocking feature. Whenever a user signs up for the application, the age of the user is stored along with the username and password. Once the user signs in with the previously made username and password, according to the age of the user if their age is above 16 years, they are given an option to turn on or off “Safe Mode” this is the option to let the user choose if they want to filter out messages or not. For users below 16 years of age, the Safe Mode option is not available and is turned on by default. A user who has turned on safe chat can neither send nor receive toxic messages. In a case where a user who has turned Safe Mode off tries to send a message which would be considered toxic by our model to a user who has Safe Mode turned on, this message would not be transmitted.

Chart 2 provides a picturized representation of the working of the chat application:

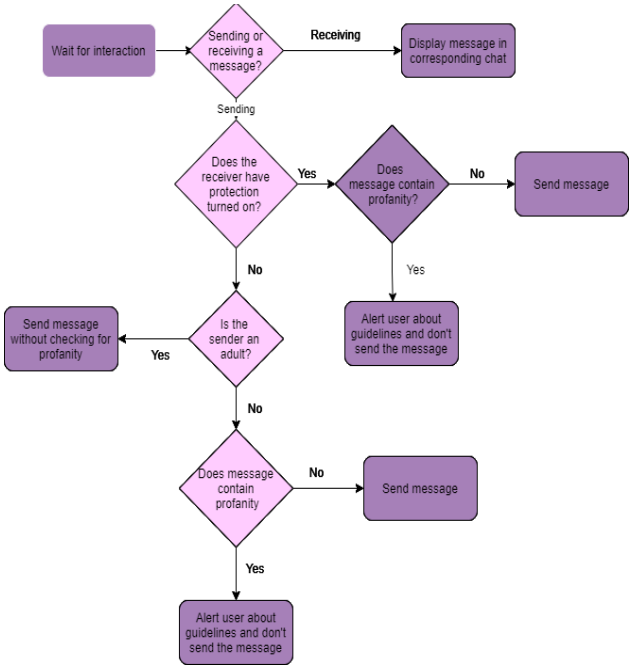

Chart 2: Flow Chart for working of the chat application

Fig 1 shows the working of the application, it can be seen blocking a toxic message written in the message box and other normal messages are sent and can be seen as a part of the conversation.

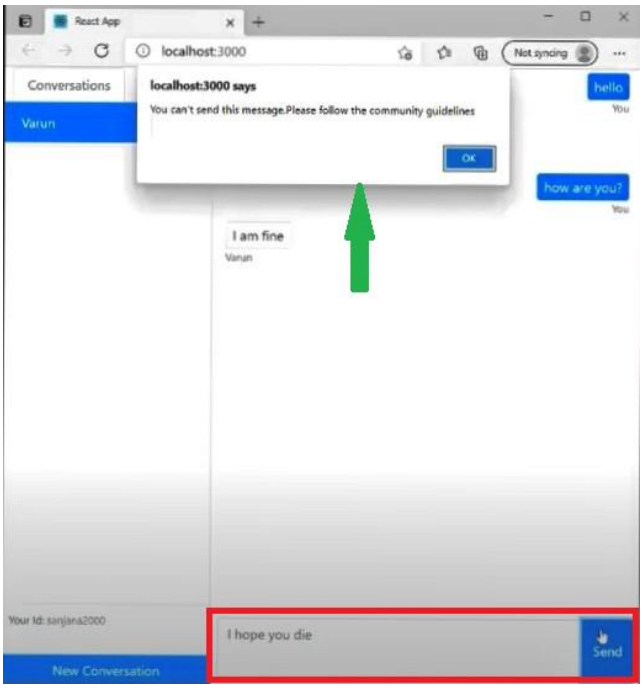

Fig 1: Working of the chat application

4. DEEP LEARNING MODEL

The Deep Learning model used for predicting if a message sent by a user is toxic or not was built with a database containing 1,84,354 entries with each entry consisting of two values, the first one being a text message and the second being the "is toxic" value which would be 1 for a toxic message and 0 for a normal message.

Out of the 1,84,354 entries, while count vectorizing, we focus on the 2000 most frequent stems (stemming of a word is the root of a word for example the stem of the word "teaching" is "teach") and words and train our model on the number of occurrences of these 2000 words in the text. To make the model flexible in understanding more words, the synonym feature is used where in each new text message, we find out of if there are any words that are not the same words or stems that are the most frequent which our model recognizes but are remarkably similar the frequent words which our model knows how to interpret. For example, if in a case where a word like "murder" is in the most frequent words, a word like "kill" would be interpreted as having the same meaning using the synonym feature thereby making the system more flexible. The flow can be seen in Chart – 3

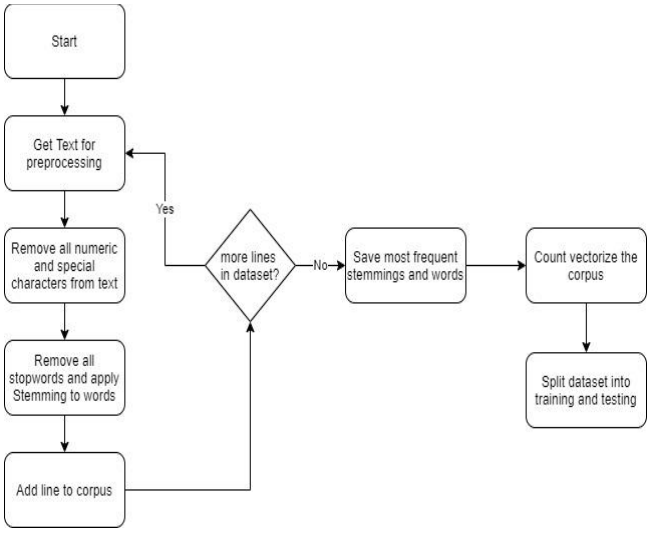

Chart 3: Flow Chart for preprocessing of data before training Deep Learning Model

The model was trained over a training set of 1,47,483 entries and the rest of the 36,871 which was an 80-20 split for training and testing data. The training was done with a batch size of 124 entries and over 20 epochs with the optimization function as "Adam" and the loss function as "binary cross entropy" since the model had only one binary output. This training method yielded the results, training set accuracy as 96% and test set accuracy as 92%. The loss

function value when plotted against the number of epochs is shown in Chart 4.

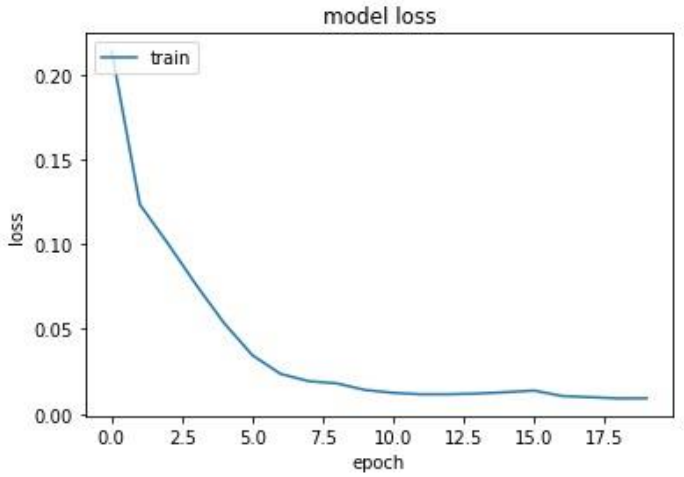

Chart 4: Plot of Loss function against the number of epochs while training neural network model.

The performance of our Deep Learning is displayed in the Fig 2.

```
enter your input : you are such a bitch
[[0.99999774]]

enter your input : i will kill her
[[0.99532485]]

enter your input : oh my god, you look so pretty
[[0.00049109]]

enter your input : i really love when i am around you
[[1.0292016e-05]]

enter your input : can you just fuck off?
[[1.]]

enter your input : i had a bad day today
[[0.23228352]]

enter your input : i hate you, you faggot
[[1.]]

enter your input : fucker you should just dissapear
[[0.9999999]]

enter your input : i am going to rape you
[[0.9994917]]

enter your input : hey dude how are you doing today
[[0.27252144]]
```

Fig 2: Results provided by our model (Values closer to one imply toxicity and values closer to zero imply non-toxicity)

5. CONCLUSIONS

The deep learning toxicity detection model is successfully implemented and works seamlessly with any application and has produced impressive results. The chat application that was developed as a proof of concept also proved to be an excellent addition to the project where we have added functionalities like blocking users from future conversations and Safe Mode, where the user (if an adult) can choose if they want to block toxic messages or not, this way, it will not be mandated for adults that the filtering must remain on. This filter if implemented into chat platforms with many active users, could impact a lot of lives in a positive way and help in making the internet a much safer environment for people who are prone to being cyber bullied or swarmed with toxic messages.

REFERENCES

[1] "Deep learning for detecting inappropriate content in text" - Harish Yenala, Ashish Jhanwar, Manoj K. Chinnakotla Jay Goyal, International Journal of Data Science and Analytics (2018) 6:273–286

[2] Method of Profanity Detection Using Word Embedding and LSTM (Long Short-term Memory)" - MoungHo Yi, Myung Jin,Lim,Hoon Ko, JuHyun Shin, Hindawi Mobile Information Systems Volume 2021, Article ID 6654029, 9 pages

[3] "Social Media Cyberbullying Detection using Machine Learning." - John Hani, Mohamed Nashaat, Mostafa Ahmed, Zeyad Emad,Eslam Amer, Ammar Mohammed, (IJACSA) International Journal of Advanced Computer Science and Applications