Toxic Language Detection in Online Video Games Using BERT and Naïve Bayes Algorithm

Auninda Alam , Marjan Tahreen , Shohag Rana , Md Humaion Kabir Mehedi , Mohammed Julfikar Ali Mahbub and Annajiat Alim Rasel

Department of Computer Science and Engineering
Brac University
66 Mohakhali, Dhaka - 1212, Bangladesh
{*auninda.alam, marjan.tahreen, shohag.rana, humaion.kabir.mehedi, mohammed.julfikar.ali.mahbub*}
@g.bracu.ac.bd
annajiat@bracu.ac.bd

***Abstract*—Toxic language and hate speeches are very common in online video games and other social platforms which can have adverse psychological effects on the minds of children and even on adults. Even while efforts have been made to filter out this type of language from platforms, it has not been viable to do so completely because low precision can limit users' freedom. In this paper we have proposed two models, one using TF-IDF and Naïve Bayes and the other using BERT. Both models were trained and tested using a dataset containing both toxic and regular tweets from twitter. The two models were able to achieve very high classification accuracy with the TF-IDF and Naïve Bayes model reaching around 91.4% and the BERT based model reaching around 95.91% accuracy. The BERT based model performed remarkably and proved to be the best alternative for detecting toxicity in text based dataset.**

***Index Terms*—BERT, NLP, TF-IDF, toxic, Naïve Bayes**

I. Introduction

The history of video games started in 1950's when computer scientists started to develop elementary video games in primitive computers. The dawn of modern video games really began when Steve Russell at MIT created "Spacewar!". Video games has evolved since then and according to a survey by J. Clement the gaming market value worldwide at 2020 is 159.3 USD which is expected to rise over 200 billion USD by 2023 [1]. This booming industry has gradually attracted a huge audience and now online multiplayer games has become not only a major source of entertainment but also a major social platform. Especially during quarantine situation arising from COVID-19 pandemic online video games has become the prime source of entertainment and socialization for the young people. However, according to a 2020 survey by ADL, about 81 percent of adults aged 18-45 have experienced some form of harassment while playing online video games [2]. Alarmingly, most of these games are not age restricted and as a result the children playing them are being exposed to toxic language and behavior which is bound to have a negative impact on their minds. However, toxic and abusive language is not exclusive to video games as social platforms like twitter, Facebook and YouTube are brewing with hate speech. Computer Scientists from all over the world have tried to combat this using several kinds of NLP models in the past decade.

On a separate note, the world of NLP changed completely after the paper "Attention is all you need" by Vaswani et al. was published in 2017 [3]. Attention was inaugurated as an independent learning model which made NLP extremely transformer dependent. Transformers have been the driving force of the evolution of NLP since then. However, with the release of BERT a new age of NLP really started. As BERT broke multiple records held by previous NLP models and since it was open sourced and was already trained on huge datasets, anyone is now able to develop an advanced NLP model with ease. In this paper, we have trained a model using TF-IDF vectorizer and another model using BERT on the same dataset. Afterwards, we have compared the potency of these models based on their performance while detecting toxic language on the validation set. The subsequent of the paper is organized as follows: Section II contains information about other researches relevant to our work. Section III contains the details of our data set and its preprocessing as well as a discussion about our proposed method. Finally, Section IV contains the comparison between the performance of the two models that we used to train the dataset.

II. Existing works

A. S. Saksesi [5], looks into categorizing text from various Twitter accounts by focusing on the presence of hate speech elements. The paper proposed Deep Learning method with Recurrent Neural Network (RNN) algorithm to do the classification. This algorithm was beneficial for creating patterns that helped the classification process based on hate speech or not hate speech. The research was able to reach 91% accuracy in recognizing hate speech with recall 90% and average precision of 91%. To address the issue of using social media platform such as twitter to defame women using hate speech, a model for detecting cyber hate was proposed by H. Sahi [6]. The research suggested a supervised learning model where they used different machine learning algorithms such as Naive Bayes, Random tree, Random forest, J48, SVM using Poly

Kernel and SVM using RBF Kernel. They have focused on Turkish text for the classification of hate speech towards the women on Twitter. Among all the algorithms, SVM gave them the best result with 0.97 precision though the recall values were not as desired. Lastly, the research also suggested that the proposed framework will also be effective on different textual content apart from tweets.

Many noteworthy speech-detection based works can be found that focuses on detection of hate speech on social media. For instance, M. U. S. Khan [7] proposed a CNN-based service framework which categorizes the hate speech on social media under three categories which are hate speech, offensive and non-offensive. Rather than considering a multiclass problem, this research focuses on multilabel problem for the detection of hate speech. Multilabel classification for sequential CNN based model has showed 20% better performance compared to the multiclass problem. Again, when it comes to NLP, we must consider emojis as they possess the potential of holding the real meaning behind a sentence.

M. Aquino et al. [8], proposed a machine learning technique for detecting the toxicity of a comment by looking at both the text and the emojis in the comment. They used word embeddings generated by GloVe and emoji2vec to train a bidirectional Long Short Term Memory (biLSTM) model. A fresh labeled dataset comprising text and emoji comments was also created. On first findings, the model's accuracy score was 0.911. We can see another notable use of Naive Bayes in paper [9]. This paper basically uses Naïve Bayes and HMM to classify emails on the basis of their category. The mentioned paper differentiates both these algorithms on the basis of accuracy using multiple NLP methodologies. In general, the paper tries to find whether an email is a spam or not.

On the other hand, Bert is a state-of-the-art model that is being used widely now with NLP. For instance, M. Tan, D. Chen [10] offers a BERT model which utilizes the structural transformation of BERT to address the issue of a large proportion of erroneous strings generated by spelling errors in the process of official document writing. To provide detection of complete spelling error and rectification, the BiLSTM network is used to identify the position of erroneous characters, and then the BERT network is used to add the pinyin previous knowledge of the error location. In comparison to the classic language error correction model, the Character-Phonetic BERT model enhances the effect by roughly 5%; in comparison to the Bert-Finetune model without incorporating the pinyin information of the mistake site, the Character-Phonetic BERT model improves by 2.1 percent. Again, P. Malik [11] proposed a model with the idea of classifying text into three classes named toxic, non-toxic or unclear. For this purpose, they used different Machine Learning and Deep Learning algorithms such as LSTM,CNN, LR, XGBoost etc on their model and from these, the highest result were obtained from CNN, LR and XGBoost. They used a dataset which was the combination of two different datasets containing toxic conversations. Moreover, Bert and fastText were used as the embedding methods along with NLP for the preprocessing of the data.

III. PROPOSED METHOD

A. Dataset

From our own experience from online video games and from going through several YouTube videos and articles regarding the toxicity of the most famous online video games such as League of Legends, Dota2, Valorant and Overwatch we discovered that the hate speeches and toxic languages used in these games by players are not so different from those used in other social platforms like Twitter, Facebook and YouTube. We found a dataset on Kaggle that had a good mixture of Toxic and non-Toxic tweets from twitter and decided to use it to train our models [4]. The dataset contains 54,313 unique tweets among which 32,592 are non-toxic texts and 24,153 contains toxic language. So, 57.4 percent of the texts are non-toxic and 42.6 percent are toxic as we can see in fig 1.

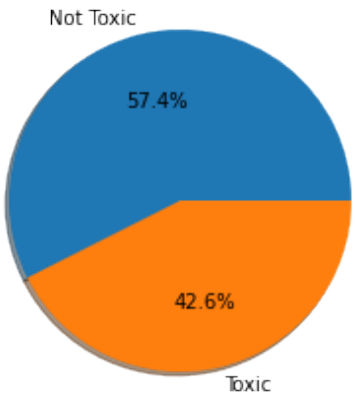

Fig. 1. Pie chart showing the ratio of two classes of texts

Most of the texts in the dataset contains 0 to 200 characters with a few containing 200-300 characters. The non-toxic messages in the dataset are mostly around 100 characters while some toxic messages exceed 200 characters as we can see from fig 2.

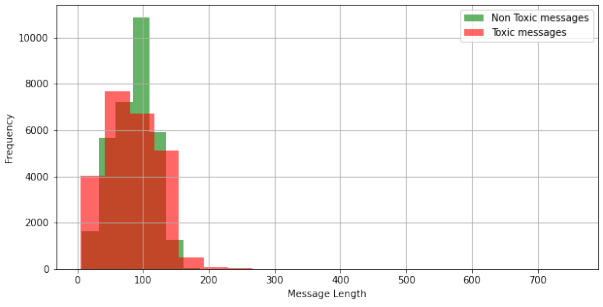

Fig. 2. Graph showing the length of texts of both classes

The dataset contains 3 columns, an unnamed id column that holds no value in our calculations, a column labelled Toxicity that contains 0 or 1 denoting non-toxic or toxic texts respectively and finally the tweet column that contains the

text data based on which our models will be trained. Fig 3 represents the overview of the dataset structure.

	Unnamed: 0	Toxicity	tweet
0	0	0	@user when a father is dysfunctional and is s...
1	1	0	@user @user thanks for #lyft credit i can't us...
2	2	0	bihday your majesty
3	3	0	#model i love u take with u all the time in ...
4	4	0	factsguide: society now #motivation

Fig. 3. Overview of the dataset structure

Afterwards, the entire dataset was split into two sets, a training set that contained 90% of the data and a validation set that contained 10% of the data.

Then, for training our Naïve Bayes model we removed stop words, special characters and punctuation from the texts. As the dataset was formed of tweets from twitter there were abundance of special characters like '@' which is used to tag a user. As these kinds of characters bared no meaning to the context they had to be removed before training our bag of words model.

However, for data preprocessing for the BERT model was much simpler. We only needed to remove the terms with '@' and remove trailing spaces etc.

```
Original:   @user when a father is dysfunctional and i
Processed:  when a father is dysfunctional and is so s
```

Fig. 4. An example of original and preprocessed text data.

B. TF-IDF and Naïve Bayes

- Naïve Bayes algorithm is based on Bayes' Theorem. It is unique in its approach to problems as it assumes that a feature in a class has no relation to any other feature in that class. So, it assumes that every feature that is to be classified is independent of each other. Then, every feature is considered to have the same importance [12].

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (1)$$

- In order to train our Naïve Bayes model, the data in the tweets column of our dataset was vectorized using TF-IDF. TF-IDF is used to assess the relevance of a word in a document within a number of documents. Its value increases by the number of times a word is present in a document but the number of documents in which the word is present offsets its value.
- After that, we used the AUC score and cross-validation in order to find out the hyperparameter. We used stratified KFold to shuffle the dataset before cross validation.
- We used multinomial Naïve Bayes in our model because it shows the best performance in case of data which can be converted into some form of count, in this case a count of word in text. In this case, we tuned the alpha value which is the hyperparameter for Multinomial Naïve Bayes and figured out the alpha for which our model would achieve the highest AUC score.

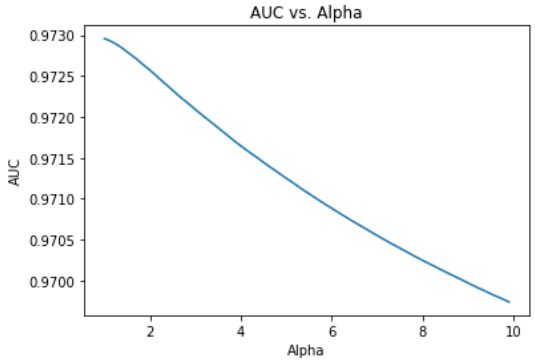

Fig. 5. Finding out the best Alpha (1.0)

- Finally, we assessed the performance of our model on the validation set. We calculated the AUC and Accuracy of the model created using TF-IDF and Naïve Bayes.

C. BERT

BERT is a unique approach to the transformer model as instead of reading the text from left to right, the transformer encoder taken in all of the words at the same time. So, BERT can be called a bidirectional or even a non-directional process. Semi-supervised Sequence Learning, ELMo, and ULMFit are examples of recent work in pre-training contextual representations that BERT relies on. BERT, on the other hand, is a deeply bidirectional and unsupervised language representation, that has been pre-trained using a plain text corpus, unlike previous models. Usually, BERT takes longer to train and is more accurate than its counterparts because it undergoes a sophisticated process before taking in sequences of words. In a sequence of words, 15% of them are masked, i.e., the original values of the words are hidden from the model and the model tries to guess the actual value of the words that were masked by reading the contextual meaning provided by other words that were not masked. Due to this reason, BERT takes longer to converge than other directional models.

- Our BERT model before training had to complete a number of tasks. Such as, the sentences in our dataset had to be tokenized, a special kind of token had to be added before and after every sentence, every sentence had to be truncated or padded to max length, the created tokens had to be mapped to their respective ID and a dictionary of outputs had to be created.
- After specifying the max length of the sentences, we tokenized the data.
- A feed forward neural network with one hidden layer was used as the classifier for this purpose. The BERT classifier model reads the final hidden layer of [CLS] tag.
- We used 16 as the batch size, 5e-5 as the learning rate and the number of epochs for training was 2. At the end

```
Original:   @user when a father is dysfunctional and is so selfish
Token IDs:  [101, 2043, 1037, 2269, 2003, 28466, 2389, 1998, 2003,
Tokenizing data...
```

Fig. 6. Tokenizing data for BERT

of each epoch, we trained our model and assessed its potency on our validation set.

Epoch	Batch	Train Loss	Val Loss	Val Acc	Elapsed
1	20	0.583503	-	-	8.09
1	40	0.303252	-	-	7.58
1	60	0.255561	-	-	7.53
1	80	0.202346	-	-	7.59
1	100	0.231847	-	-	7.55
1	120	0.186604	-	-	7.59
1	140	0.210556	-	-	7.56
1	160	0.104222	-	-	7.60
1	180	0.205645	-	-	7.58
1	200	0.220272	-	-	7.60
1	220	0.209740	-	-	7.62
1	3140	0.147314	-	-	7.55
1	3160	0.200525	-	-	7.59
1	3180	0.187956	-	-	7.57
1	3191	0.074526	-	-	4.11
1	-	0.174866	0.160803	95.19	1249.31

Fig. 7. An epoch during training our BERT model.

- Finally, after completion of training we tested the potency of the model on the validation set. We calculated the AUC score and accuracy for the BERT model as well.

IV. Results and Discussion

After using TF-IDF and Naïve Bayes algorithm to train our basic model, we tested it on the validation data. The AUC score and accuracy of the model was calculated based on its performance while predicting the classes based on the validation set. The AUC score was 0.9721 and the accuracy was 91.47 percent.

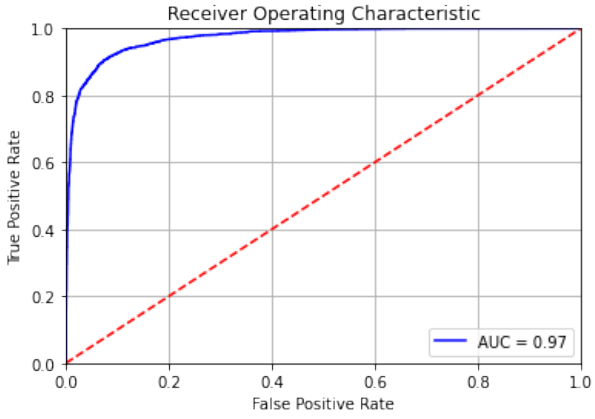

Fig. 8. AUC and Accuracy for TF-IDF and Naïve Bayes model

Afterwards, after training our BERT model, we tested it on the same validation data as well. The AUC score and accuracy were calculated for this model as well based on its performance while predicting toxicity from the tweet, column of the validation set. The AUC score in this case was 0.9898 and the accuracy was 95.91 percent. Ultimately BERT gave a better accuracy and AUC score in comparison to TF-IDF and Na ıve Bayes.

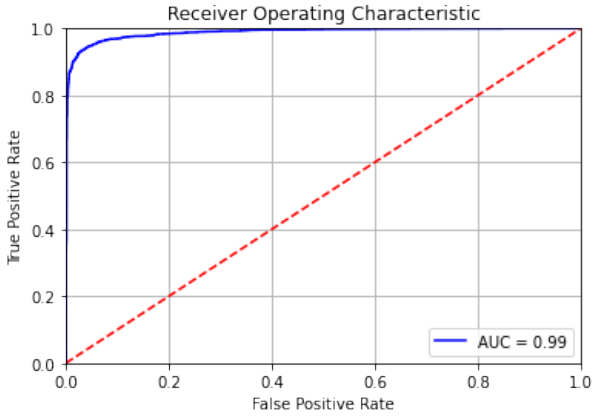

Fig. 9. AUC and Accuracy for BERT model

After the evaluation of both models we have found that the BERT model clearly yields the better results. From the AUC-ROC curves we can see that both the models have achieved outstanding True positive and True negative rates.

Here, in I we have compared the accuracy of all the algorithms implemented

TABLE I
Comparison between the two models

Model	AUC	Accuracy	Precision	F1 Score
TF-IDF and Naïve Bayes	0.9721	0.9147	0.8872	0.9016
BERT	0.9898	0.9591	0.9091	0.9357

From the comparison we can see that the BERT model which was created by adding one more hidden layer with BERT yielded more than 4 percent increased accuracy when compared to the generic TF-IDF model.In the AUC score we can see a slight improvement of 0.01 points. Even though we used a small batch size and ran only 2 epochs, a notable improvement is evident when compared to the TF-IDF Naïve Bayes model. Therefore, we can come to the conclusion that the BERT model is definitely a better choice when predicting the dataset that we used and also potentially the best choice for predicting any kind of toxic language on any dataset.

V. Conclusion

This paper proposes two models for detecting toxic chats and texts in video games and potentially in any kind of social platform. Here, we have used TF-IDF and Multinomial Naïve Bayes to develop a model at first which represents the classic forms of NLP. Later on, we have developed another Neural Network based model that included BERT and both the models were trained and tested using the same training dataset and validation set respectively. While both models yielded very satisfactory results, BERT clearly outperformed the classic model and hence proved to be the best model for solving the

problem statement of this paper. Therefore, we will continue to use BERT in order to perfect a model that can effectively filter out toxic languages in video games. In the future, this work can be improved by forming datasets from actual video game chats and not from other social platforms. The future scope of this research includes but is not limited to: toxic language detection in video game voice chats, hate speech detection in YouTube videos, Detection of depression from video game chats etc.

REFERENCES

[1] J.Clement, "Gaming Market Value Worldwide 2012-2023," Video game market value worldwide 2012-2023, Statista.com, November 2021.

[2] 'Free to Play? Hate, Harassment and Positive Social Experience in Online Games 2020, adl.org, 2020.

[3] A Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser and Illia Polosukhin, "Attention Is All You Need," 2017.

[4] Ashwin U Iyer, "Toxic Tweets Dataset," Kaggle.com, 2021.

[5] A. S. Saksesi, M. Nasrun and C. Setianingsih, "Analysis Text of Hate Speech Detection Using Recurrent Neural Network," 2018 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC), 2018, pp. 242-248, doi: 10.1109/ICCEREC.2018.8712104.

[6] H. Şahi, Y. Kılıç and R. B. Sağlam, "Automated Detection of Hate Speech towards Woman on Twitter," 2018 3rd International Conference on Computer Science and Engineering (UBMK), 2018, pp. 533-536, doi: 10.1109/UBMK.2018.8566304.

[7] M. U. S. Khan, A. Abbas, A. Rehman and R. Nawaz, "HateClassify: A Service Framework for Hate Speech Identification on Social Media," in IEEE Internet Computing, vol. 25, no. 1, pp. 40-49, 1 Jan.-Feb. 2021, doi: 10.1109/MIC.2020.3037034.

[8] M. Aquino et al., "Toxic Comment Detection: Analyzing the Combination of Text and Emojis," 2021 IEEE 18th International Conference on Mobile Ad Hoc and Smart Systems (MASS), 2021, pp. 661-662, doi: 10.1109/MASS52906.2021.00097.

[9] S. R. Gomes et al., "A comparative approach to email classification using Naive Bayes classifier and hidden Markov model," 2017 4th International Conference on Advances in Electrical Engineering (ICAEE), 2017, pp. 482-487, doi: 10.1109/ICAEE.2017.8255404.

[10] M. Tan, D. Chen, Z. Li and P. Wang, "Spelling Error Correction with BERT based on Character-Phonetic," 2020 IEEE 6th International Conference on Computer and Communications (ICCC), 2020, pp. 1146-1150, doi: 10.1109/ICCC51575.2020.9345276.

[11] P. Malik, A. Aggrawal and D. K. Vishwakarma, "Toxic Speech Detection using Traditional Machine Learning Models and BERT and fastText Embedding with Deep Neural Networks," 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), 2021, pp. 1254-1259, doi: 10.1109/ICCMC51019.2021.9418395.

[12] "Naïve Bayes Classifier Algorithm," javatpoint.com.