TOXIC COMMENTS DETECTION AND TOXICITY SCORE USING BIDIRECTIONAL GRU

Nikitha V
Dept. of Computer Science and Engineering
PES University, Karnataka, India
nikithav678@gmail.com

Harshitha S
Dept. of Computer Science and Engineering
PES University, Karnataka, India
Harshithas2173@gmail.com

Marada Likitha
Dept. of Computer Science and Engineering
PES University, Karnataka, India
maradalikitha281@gmail.com

M P Deepti
Dept. of Computer Science and Engineering
PES University, Karnataka, India
deeptimp2003@gmail.com

Prof. Kamatchi Priya L
Dept. of Computer Science and Engineering
PES University, Karnataka, India
priyal@pes.edu

***Abstract*– Although social media has made it possible for everyone to express their thoughts to a wider audience, it has also served as a venue for offensive, provocative, rude, and cyberbullying remarks, as well as harsh language and cruel behaviour. The study addresses online harassment, cyberbullying and abuse by automating the process of classification of toxic comments on social media. Our approach, which makes use of the Bidirectional GRU and CNN models, offers a reliable way to classify comments into a number of categories, including identity hatred, obscene, insult, severe toxic, and threat. Based on the assessment criteria, our approach provides a better performance in precisely detecting, classifying and identifying dangerous, poisonous remarks. Our method offers a superior performance in accurately detecting, categorizing and identifying harmful toxic comments, as measured by the evaluation metrics. Furthermore, our approach adds a toxicity score, which gives a numerical representation of the level of toxicity in comments.**

Keywords: **Bidirectional GRU, Toxicity score**

I. INTRODUCTION

Social media can be both a boon and a bane in our time. Comments on social media platforms have significant influence over individual's mental well- being and perspectives. Flagging such comments would save social media users from the negative impact of toxic comments and foster a healthier online social environment. In our project we developed a deep learning model that can automatically detect and classify toxic comments in online settings. Toxic comments can be classified into six classes of toxicity, namely- toxic, severe toxic, obscene, threat, insult and identity hate. By identifying and flagging these comments, online platforms can improve the quality of discussions and create safer environment for users. Our project also implements a toxicity score given based on the severity of the toxic comment.

Research has looked into hate speech, offensive language, online harassment. In the papers we have explored a variety of approaches have been used to tackle this problem. In [1], the authors M. Z. Islam et al. proposed a CNN model with LSTM on the Wikipedia Talk dataset and achieved an accuracy of 95.1%. The advantage is that noisy and unstructured data can be handled well due to the intense preprocessing done. The drawback faced is the unavailability of a diverse and huge dataset. Another paper [2] proposed the CNN, LSTM, Stacked ensemble model which helped achieve a high accuracy. The dataset used is the Wikipedia Talk dataset, and the shortcoming faced was the need for better and more accurate annotations. G. Mishra et al. [3] used the CapsuleNet model on the Jigsaw Toxic Comment Classification dataset and achieved an accuracy of 95%. The major advantage in this model is the ability to handle varying length inputs and improve pattern recognition and understanding contexts. The drawback is limited data availability. S. S. Rana et al. [4] used Transformer model on the Jigsaw dataset and achieved an accuracy of 95.7%. Transformers can learn long range data and capture deep patterns in text. The disadvantage is the complexity and need for additional resources. In [5], D. Kim et al. proposed the bidirectional GRU model with adversarial training. The advantage of using this model is the robustness towards adversarial attacks and handle training on limited data, as they used the Jigsaw Toxic Comment Classification dataset.

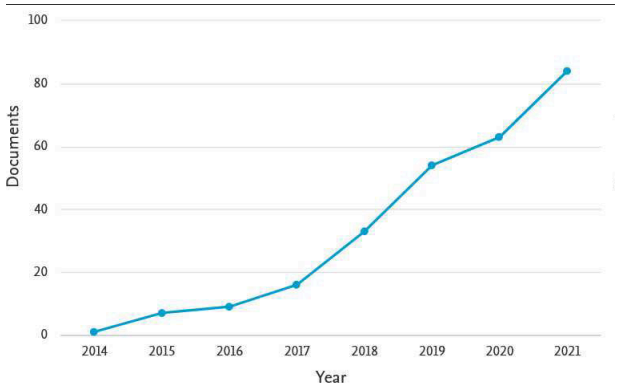

Fig. 1. Illustrates study related to toxic language detection over the years

In the further sections we will explore our literature survey, proposed methodology, implementation, result analysis, conclusion and further research. We shall also look into the dataset used in the implementation section.

II. LITERATURE SURVEY

Numerous machine learning models have been used for this use case, from CNNs to Transformers and each with varying results. We reviewed a large number of papers and the summary of some of them are as follows.

In [6], the authors Foh et. al., proposed two deep learning models and compared their performance.. The dataset used was a custom dataset made with keywords from hatebase.org which has Twitter data and then annotated by CrowdFlower. After preprocessing, the dataset consisted of a total of 24783 tweets, with 1430 rows annotated as hate speech, 19190 annotated as offensive tweets, 4163 tweets labeled as neither hate nor offensive speech. They preprocessed the data by removing '@' and hashtags were split into multiple words to improve the representation of the text The models used were CNN and bidirectional LSTM, both integrated with BERT and fastText embeddings. For binary classification, the Bi-LSTM classifier achieved the best performance with a macro-average F1-measure of 0.945 using BERT fine-tuning, while the CNN classifier achieved sn F1-measure of 0.942 using fastText embeddings. For multi-class classification, the Bi-LSTM classifier achieved the best performance with an F1-measure of 0.838 using BERT fine-tuning, while the CNN classifier achieved an F1-measure of 0.835 using fastText embeddings It was also observed that BERT fine-tuning performed much better than the feature-based methods for the Twitter data. The disadvantage in this paper is that the line between hate and offensive speech was hard to draw.

Bidirectional LSTM networks are presented in the study by Salehgohari et al.[7] as a method for identifying abusive language in social media. By utilizing natural language processing (NLP) techniques like word embedding, such as Word2Vec and GloVe, the study successfully presents text data. CNN was also used as a comparison tool with conventional machine learning models. With an accuracy of 94.52 on test data, this methodology was applied to classify abusive remarks into multiple classifications, including severe toxic, toxic, toxic, obscene, insult, threat, and identity hatred, with the goal of improving online safety and reducing cyberbullying.

In the paper [8] the authors Li,Hao, et al., implemented three models Naïve Bayes SVM, BiLSTM and BERT. Naïve Bayes SVM model was used as a baseline model. However this model achieved F1 score 20% lower than EM score due to imbalanced data which led to model's tendency to predict comments as nontoxic. LSTM model was trained for 4 epochs using Adam optimizer. The learning rate was adjusted at the end of each epoch using a learning rate scheduler. The loss function used was cross entropy loss. LSTM model has achieved a much higher F1 score and EM score as compared to the Naïve Bayes Model. This was possible as LSTM remembered the relationship between words which was ignored in our baseline model. Weighted Loss LSTM model was tried out to solve the imbalanced data problem by setting the weight of prediction loss for toxic comments greater than the weight prediction loss for non-toxic comments. This hyperparameter was very helpful to accurately fine-tune the model trained on an imbalanced dataset. LSTM achieved an overall F1 score of 77.95% and EM score of 95.37% and LSTM with weighted loss pair achieved an F1 score of 81.12% and EM score of 95.21%. BERT was trained for one epoch and achieved results on par with LSTM. Batch size was set to 32 due to GPU memory limitation and learning rate used was 2 x 10-2. It achieved an F1 score of 77.37% and EM score of 95.73%. BERT with weighted loss pair achieved an F1 score of 81.18% and EM score of 95.55%. As a conclusion, BERT was seen to perform better than all the other models in spite of being trained on just one epoch.

In the paper [9], by the authors Taleb, et. al , various classifiers and NLP techniques were used to detect and classify those toxic comments which provides a safer environment for the users. The dataset used was "The Toxic comment classification challenge" which was taken from kaggle which includes 159,571 instances with comments and several labels. The proposed methodology includes two main parts. The first part determines a classifier which classifies the comments and the next part includes detecting the toxic content based on the predictions of the classifiers. Models such as Naive Bayes, logistic regression, Random Forest, XGBoost, CNN, LSTM, GRU and BERT were used with Bag of Words (BoW), TF-IDF vectoring and various text representations to determine the toxic comments. Naive Bayes and SVM achieved an accuracy of 87.57%, LSTM with Adam optimizer and BERT attained accuracy of 95.54%, Ensemble models had an accuracy of 95.14%, LSTM with glove and FastText Embeddings achieved an accuracy of 98%, GRU with glove and FastText Embeddings had reached an accuracy of 93% .

However there were many challenges and drawbacks in these models. These includes bias in data, complexity and interoperability of deep learning models. Data privacy concerns and adaptability issues of the models to the new form of toxic comments were found. The

Occurrence of false positives and negatives which had an impact on the overall effectiveness of the toxicity detection of the systems.

III. PROPOSED METHODOLOGY

From the literature review, we observe that a large number of models have been implemented for this use case, most of them being LSTMs and BERT models. BERT models are computationally intensive and may require GPUs for fast training. While LSTM has shown promising results, it is also computationally intensive and requires higher memory, though lesser than BERT, and is more prone to overfitting. Finding middle ground with less intensive models and high accuracy is the aim of this paper. The following paragraphs describes the methodology proposed by this paper:

A. Data preparation:

Having a clean and well processed dataset is essential to train the model effectively. The dataset used is the Jigsaw Toxic Comment Classification dataset. It consists of Wikipedia comments which were annotated by humans and was provided as part of a Kaggle challenge. It consists of 6 classes - toxic, severe_toxic, threat, identity_hate and obscene. These classes are annotated with 0/1 indicating if they belong to that class ot not. The dataset has certain rows of comments unfilled or filled with an empty string (" "). This issue is addressed by replacing all such comments with the string "no comment", providing a general form for such discrepancies.

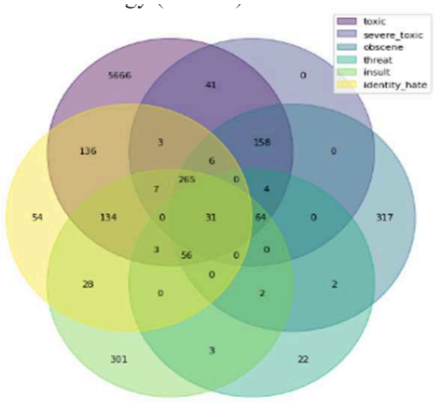

Fig. 2. Distribution of the dataset

B. Word Embeddings:

Word embeddings are a type of representation where words are converted to a vector form for the machine learning models to understand. Each word will be mapped to a unique vector and words semantically and unsemantically similar to each other will be in the same area in the vector space.

Pre-trained embeddings are available online to enable faster embeddings as it is a complex training process which requires intensive resources like GPUs. Some of the pretrained embeddings available are fastText model by Facebook, GLoVe (GLobal Vectors for word representation), BERT models by Google etc. The one used in this paper is the GLoVe embedding, specifically the glove.840B.300d which contains 840 billion tokens and 300-dimensional vectors.

GLoVe works by building a co-occurrence matrix, which is a square matrix in which each element Aij represents how often word i occurs in the context or surroundings of word j. An objective function is used to learn these word embeddings and capture relationships. It works behind the idea that the dot product of vectors that co-occur frequently is proportional to the logarithm of their count of co-occurrence. The result is a set of vectors which capture semantic patterns and relationships between words.

C. Bidirectional Gated Recurrent Unit (GRU):

GRU is an optimization of Recurrent Neural Network (RNN) that is optimized to avoid vanishing gradient and also have selective memory. It employs a gating mechanism to enable this selective memory and control flow of information through the neural network. A bidirectional GRU processes input sequences in both directions (forward and backward) by using two separate GRUs, one for each direction. This enables the model to capture intricate patterns in both past and future context.

Assume x(t) to be the input text at some time t, h(f) to be the hidden state in the forward processing GRU and h(b) to be the hidden state in the backward processing GRU. It has two main gates:

Update gate(z): It decides how much of the old hidden state (h(t-1)) to keep and how much of the new candidate state ($\tilde{h}(t)$) to use. It is computed as:

$$z = \sigma(W(z) * x(t) + U(z) * h(t-1))$$

Where W(z) and U(z) are the weights and $\boldsymbol{\sigma}$ is the sigmoid function.

Reset gate(r): It decides how much of the previous (old) hidden state to forget. It is computed as:

$$r = \boldsymbol{\sigma}(W(r) * x(t) + U(r) * h(t-1))$$

With the help of these gates, a GRU model can selectively read and update the hidden state based on the current input and the previous hidden states, and captures dependencies in sequences more effectively.

D. Convolutional Neural Network (CNN):

It is a neural network architecture generally used for extracting features from images, text and video data. It uses a set of filters or kernels to pass over the data and perform convolution operation. This

results in a feature map representing all the extracted features. These kernels represent a set of learnable weights that are updated during back propagation based on the loss calculated.

The model proposed uses a 1-dimensional CNN to extract local features from the output of the GRU. The kernel size for this task is typically small, therefore, we chose a kernel size of 2 for 64 filters. The loss function used is the binary cross entropy loss, which is commonly used for binary classification tasks. It compares the predicted and actual value (lying between 0 and 1) and computes the following:

$$BCE(y, y') = -[y*\log(y') + (1-y)*\log(1-y')]$$

Where y represents the actual value and y' represents the predicted value.

Since the dataset has 6 classes, the BCE loss is calculated separately for each class, and all classes are later combined. The total loss is the average of individual BCE losses for each class.

$$BCE(y, y') = \sum_i BCE_i(y, y')$$

The features are passed into an output layer with Sigmoid activation function to predict the values for the 6 classes. The optimizer used is the Adam optimizer which helps in faster convergence and requires less hyperparamter tuning. Sigmoid function is commonly used in classification tasks and has a smooth derivative, making it suitable for optimizers like Adam. In order to prevent overfitting and achieve a generalized model, we incorporated spatial drop-out, which kills a fraction of neurons during training, allowing all neurons to be active at some point and ensuring all features are recognized.

E. Toxicity Score:

The toxicity score is a single-value representation for all 6 classes. All the classes have been given equal priority and weights. It is calculated by summing up the predicted probabilities of toxicity across all categories (toxic, severe toxic, threat, obscene, insult, and identity hate) and normalizing the sum to a range of 0 to 1. Mathematically,

$$S=\sum_{i=1} Pi/n$$

$$S_{normalized}=S-S_{min}/(S_{max}-S_{min})$$

This toxicity score gives an overall estimate of how toxic a certain comment is.

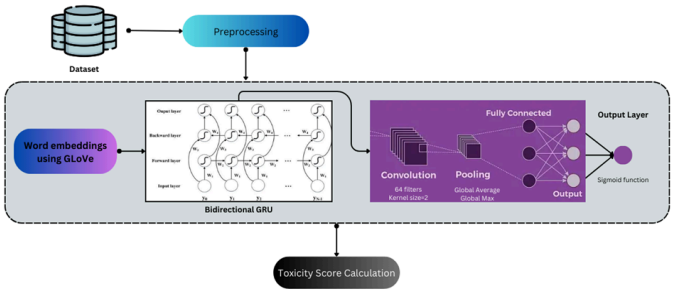

Fig. 3. Architecture of the model

IV. RESULTS

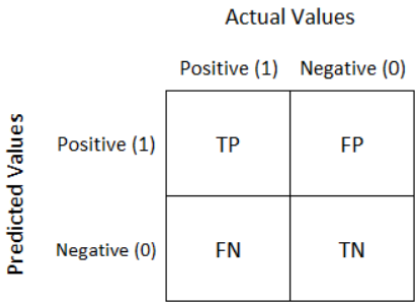

Fig. 4. Confusion matrix

Accuracy is calculated by dividing the number of correct predictions by total number of predictions made. The correctness of the model can be gauged by it.

$$\text{Accuracy} = \frac{(TN+TP)}{(TN+FP+FN+TP)}$$

Precision is the proportion of the true positive predictions (correctly predicted positives) to the total number (predicted+actual) of positive predictions(both true positives and false positives). It represents the model's ability to accurately identify positive samples.

$$\text{Precision} = \frac{TP}{(FP+TP)}$$

Recall is the ratio of all correctly identified positive instances to the total number of actually positive samples in the dataset. It assesses the model's ability to accurately detect all positive samples, even those incorrectly classified as negative.

$$\text{Recall} = \frac{TP}{(TP+FN)}$$

F1_score is the harmonic mean of precision and recall, offering a single unified metric combining both to consider the precision-recall tradeoff accurately.

$$F1 = 2 * \frac{precision*recall}{precision+recall}$$

Train-validation-test split	60-20-20
Accuracy	98.38%
Loss	4.6%
Precision	0.808
Recall	0.70
F1_score	0.75

Table 1 - Model Results

For the external test dataset containing 159,000 rows, we achieved an accuracy of 97.7%. This was used to test the credibility of the model on new and unseen data.

Overall, we can conclude that the model is performing well in classifying the toxic comments.

V. CONCLUSION AND FUTURE WORK

In this study, we've trained a deep learning model to detect and classify comments on social media into six various categories such as severe_toxic, toxic, insult, obscene, threatening and identity hatred by utilizing toxic comment dataset for training our Bidirectional GRU revealed notable outcomes.

Based on our research we've come up with the following conclusion: Firstly, our model demonstrated commendable performance for text classification giving an accuracy of 98.38%, with a precision of 0.808, recall of 0.7 and F-score 0.75 on the test data showing it's ability to accurately classify and identify toxic comments. Secondly, our model not only classifies toxic and non-toxic phrases but also provides proportion of the toxicity within the comments as well as toxicity score. Thirdly, in order to get another confirmation, we tested the model on a separate test dataset of 153165 rows. In this dataset, we got an accuracy of 97.7% which indicates the model is not overfitting, but works well overall for unseen data too. Although there is still need for improvement in recall and F1-score, our results generally highlight the model's efficacy in correctly classifying harmful comments.

Suggestions for future work include refining the Bidirectional GRU model, enlarging the dataset to improve representation, researching ensemble learning, adding user-specific features, and examining pre-training effects to improve performance.

VI. REFERENCES

[1] Islam, M. Z., Bae, J., & Lee, S. (2019). Deep Learning-Based Toxic Comment Classification. Symmetry, 11(4), 616. doi:10.3390/sym11040616

[2] Khan, N., Chandra, P., & Sharma, A. (2020). A deep learning-based framework for detection and classification of abusive language in social media. Multimedia Tools and Applications, 79(23), 16455-16476.

[3] Mishra, G., Pandey, S., & Srivastava, S. (2018). A study of capsule networks for toxic comment classification. In Proceedings of the 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI) (pp. 2015-2020).

[4] Rana, S. S., Kumar, A., Dave, M., & Sharma, D. (2020). Toxic comments classification using transformers. In Proceedings of the 2020 International Conference on Inventive Computation Technologies (pp. 978-983).

[5] Kim, D., Kim, J., Kang, J., & Kim, Y. (2020). Robust and interpretable toxicity detection with adversarial training and gated recurrent units. IEEE Access, 8, 55763-55776.

[6] A. G. D'Sa, I. Illina and D. Fohr, "BERT and fastText Embeddings for Automatic Detection of Toxic Speech," 2020 International Multi-Conference on: "Organization of Knowledge and Advanced Technologies" (OCTA), Tunis, Tunisia, 2020, pp. 1-5, doi: 10.1109/OCTA49274.2020.9151853.

[7] A. Salehgohari, M. Mirhosseini, H. Tabrizchi and A. V. Koczy, "Abusive Language Detection on Social Media using Bidirectional Long-Short Term Memory," 2022 IEEE 26th International Conference on Intelligent Engineering Systems (INES), Georgioupolis Chania, Greece, 2022, pp. 000243-000248, doi: 10.1109/INES56734.2022.9922628.

[8] Li, Hao, Weiquan Mao, and Hanyuan Liu. "Toxic comment detection and classification." In CS299 Machine Learning. Standford University, 2019.

[9] Taleb, Mohammed, Alami Hamza, Mohamed Zouitni, Nabil Burmani, Said Lafkiar, and Noureddine En-Nahnahi. "Detection of toxicity in social media based on Natural Language Processing methods." In 2022 International Conference on Intelligent Systems and Computer Vision (ISCV), pp. 1-7. IEEE, 2022.

[10] L. Dipietro, A. M. Sabatini and P. Dario, "A Survey of Glove-Based Systems and Their Applications," in IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 38, no. 4, pp. 461-482, July 2008, doi: 10.1109/TSMCC.2008.923862.

[11] Zhongguo Wang and Bao Zhang. 2021. Toxic Comment Classification Based on Bidirectional Gated Recurrent Unit and Convolutional Neural Network. ACM Trans. Asian Low-Resour. Lang. Inf. Process. 21, 3, Article 51 (May 2022), https://doi.org/10.1145/3488366

[12] Serkan Kiranyaz, Onur Avci, Osama Abdeljaber, Turker Ince, Moncef Gabbouj, Daniel J. Inman, 1D convolutional neural networks and applications: A survey, Mechanical Systems and Signal Processing, Volume 151, 2021, 107398, ISSN 0888-3270, https://doi.org/10.1016/j.ymssp.2020.107398.