Leveraging BERT and Convolutional Neural Networks to Aid in Identifying Hate Speech Amongst Online Gaming Communities

Malav Modi (mhm114) malav.modi@rutgers.edu
Shyam Patel (spp128) spp128@rutgers.edu
Praveen Sakthivel (ps931) ps931@rutgers.edu

Abstract

Due to the overwhelming implications of the COVID-19 pandemic thus far, there has been a massive influx of hate speech in online gaming communities. Despite platforms like Twitch having moderators reviewing large amounts of comments at a time, human error is bound to misinterpret a multitude of comments. Therefore, we use a deep neural network model that can predict and classify various forms of hate speech, to help encompass a safe space for all viewers. We accomplish this by using a combination of Google's BERT Transformer and Convolutional Neural Networks to help identify and classify hate speech with high accuracy/certainty.

1 Introduction:

With the recent rise of COVID-19, billions of people around the world have taken the necessary steps to keep themselves safe during the pandemic. The most common measure people have taken is self-quarantine. As a result of quarantining oneself, people look to the internet to alleviate their boredom. As a result, online platforms such as Twitch or YouTube Gaming have experienced a massive spike in the number of daily users. This phenomenon has lead to an influx of toxic and hate speech that pervades communities and endangers the safe space for their users. To help alleviate the rise in hate speech, these platforms will employ human moderators that manually check any flagged message and determine whether it follows a given platform's community guidelines for acceptable behavior. The system is heavily reliant on users reporting suspicious comments and the speed at which moderators can review these comments. This problem could be greatly improved via automating parts of the shared responsibility between users and moderators to flag and remove any toxic comment.

With regards to the automation, it can be achieved with the use of neural networks and deep learning. We will seek to help alleviate the issue by using Google's BERT [1, 2] and Yoon Kim's proposed Convolutional Neural Network structure for sentence classification [4, 9] for the classification of toxic and hate speech in online communities. Bert's ability to accurately and efficiently tokenize sentences in will enable us to use Kim's CNN to its fullest extent Kim's CNN structure has been shown to provide accurate sentence classification [4]. Since the goal of our model is to determine whether a sentence is to be classified as toxic speech, we believe that Kim's CNN is a good fit with the problem we wish to solve.

2 Previous State of Knowledge

Multiple approaches have been taken towards successfully classifying toxic comments. Classification has ranged from simple binary classifications to more specific multi-class observations. To drive these classifications, single classifiers like Logistic Regression, Recurrent Neural Networks, LSTMs, GRUs and Convolutional Neural Networks have all been used. Some models have also incorporated

multiple classifier models that aggregate the results of all several classifiers. Most of these models incorporate previously created word embedding [8]. In the specific case of the Kaggle competition from which the dataset comes from, Naive-Bayes Support Vector Machines, LSTMs and GRUs were the most successful models.

One of the main difficulties in classifying hate speech is the variance in data sets. Hate Speech is subjective and has no universal definition. Thus variance exists across data sets where one instance may be considered hate speech in one data set but not hate speech in the other [5, 3]. Additionally, it is difficult to properly learn the context of words. Certain words like trash in the sentence "I need to throw out the trash" would not be considered hate speech whereas the sentence "You are trash" would be considered hate speech. As a result, many key words can be both hate speech and not hate speech depending on the context of the sentence. Other major difficulties include properly responding to words that do no exist in the vocabulary, recognizing sarcasm in phrases that may initially look positive and recognizing idioms and metaphors that have toxic meanings [8].

3 Main

For our project,we took an encoder-decoder approach to classifying hate speech. For our encoder we use transformers, specifically the state of the art BERT (Bidirectional Encoder Transformation from Transformers). For our decoder, we use a convolutional neural network to classify our embedded input. We recreate Kim's work on Convolutional Neural Networks for Sentence Classification to create a robust classifier for our model.

3.1 BERT

Bidirectional Encoder Transformation from Transformers (BERT) expands on the state of the art transformer by allowing the encoder to learn context from all words present in the sentence, not just the words preceding the target [7]. To achieve this, BERT utilizes the traditional transformer encoder but is trained unconventionally using 2 new methods: Masked LM and Next Sentence Prediction [6].

Masked LM In Masked LM, the entire sentence is provided to the transformer but 15% of the inputted words are randomly selected to be masked using a [MASK] token. The objective of the model is to determine the correct value of the selected words. Doing so allows the model to learn the relationship between all words in a sentence while still providing targets to the train against. The 15% of words selected are not always masked. Only 80% are eventually masked. 10% of the selected words are left unchanged and the remaining 10% are swapped with a random word from the corpus. This is done to allow the model to learn the proper relationships for unmasked words in the context of the whole sentence. Masking 100% of the words would lead to the network primarily learning relationships between [MASK] and other words in the sentence and not relationships between two unmasked words as well. The remaining 20% are evenly either replaced with a random word or left alone to accomplish two purposes. If the selected word were never replaced, the model would learn the relationship that the observed word was always the correct word. Similarly, if the selected word were always replaced with a random word from the corpus, the model would learn the relationship that the observed word was never correct. Forcing a 50% split ensures that the model will use context and attention to evaluate the selected word [6].

Next Sentence Prediction In Next Sentence Prediction, the transformer is fed a pair of two sentences and is tasked with determining whether the seconds sentence comes directly after the first sentence. 50% of the sentences fed to the transformer are sequential, for the remaining, the second sentence is randomly chosen from the corpus. Since the transformer takes in only one input sequence, the two sentences are merged into one sequence. The beginning of the sequence is marked by the [CLS] token and the end of each sentence is denoted by a [SEP] token. During training, both Next Sentence Prediction and Masked LM are used with the goal of maximizing accuracy in both [6].

Usage Google provides pre-trained versions of the BERT that were utilized to embed our input vectors. The 'bert-base-uncased' model in particular was used. Kim's discusses how encoders produce embedded word vectors that produce 'universal' features; features that can be used for a variety of classification purposes. Thus this drove the conclusion that the already trained BERT encoders would be able to produce word embedding that could successfully classify hate speech.

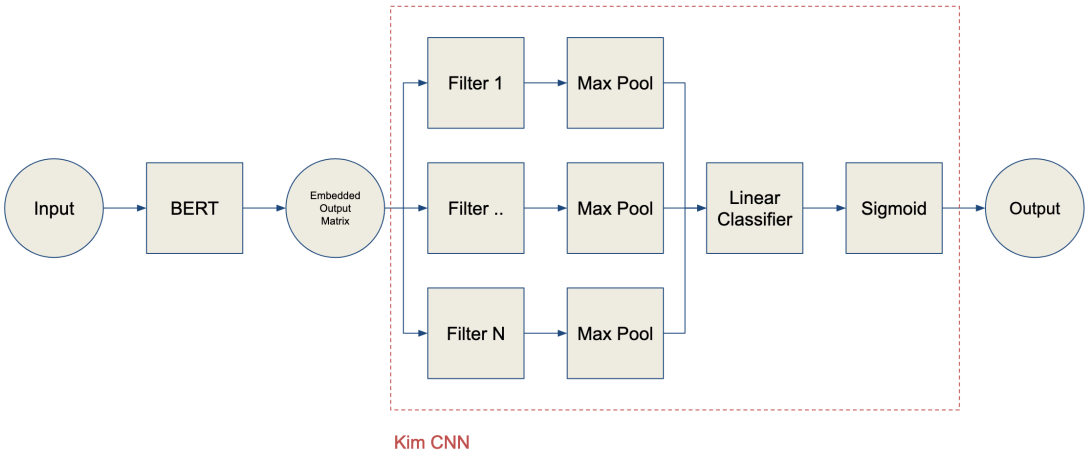

Figure 1: Model Architecture

3.2 Kim CNN

For our decoder, we opted to recreate the convolutional neural network discussed in Kim's Convolutional Neural Network for Sentence Classification. The network is comprised of multiple 2D convolutional neural networks with varying kernel dimensions that are applied to the original embedded input sequence in parallel. In our specific implementation, we used 3 convolutions with kernel dimensions of 2, 3 and 4. A dropout function is applied right after the initial convolutions. The outputs of these layers are reduced to a vector of length 3 consisting of the max value feature for each filter using the maxpool operation. These are passed into a fully connected layer to produce a vector of size 6 (the output dimension). Lastly this output vector is passed into a Sigmoid function to produce classification probabilities for all 6 labels. See figure 1 for a visualization of the entire model.

4 Experiment

4.1 Dataset

To train our model we used the dataset made available for the Kaggle Competition, Toxic Comment Classification Challenge. The dataset was loaded from a CSV and shuffled based on the random seed 69. 10,500 entries were used for training the model. Of the 10,500 entries, 7,500 were reserved for training and 1,500 were reserved for validation and test sets each.

4.2 Training

Our model was trained over 27 epochs with a batch size of 25 using the Adam Optimizer and BCE Loss. The SGD Optimizer and BCEWithLogits Loss were also tested as well, but better model performance was achieved using the combination of the Adam Optimizer and BCE Loss. During the initial runs our model overfit to the training data set (most likely due to the overall small size of the dataset). To target this issue, multiple different learning rates were tried. A learning rate of 1e-5 coupled with a dropout rate of 0.1 provided the best results. The slower learning rate prevented the model from heavily overfitting during training and the loss difference between validation and training was reduced to negligible amounts.

5 Evaluation

During testing, the model was able to achieve an accuracy of 96.7% in correctly predicting toxic classifications. To observe individual accuracy across the 6 labels, An ROC curve was constructed and used to calculate a score for each individual label. The model was able to accurately classify every label with at least 93% accuracy. See Table 1 for a list of full scores.

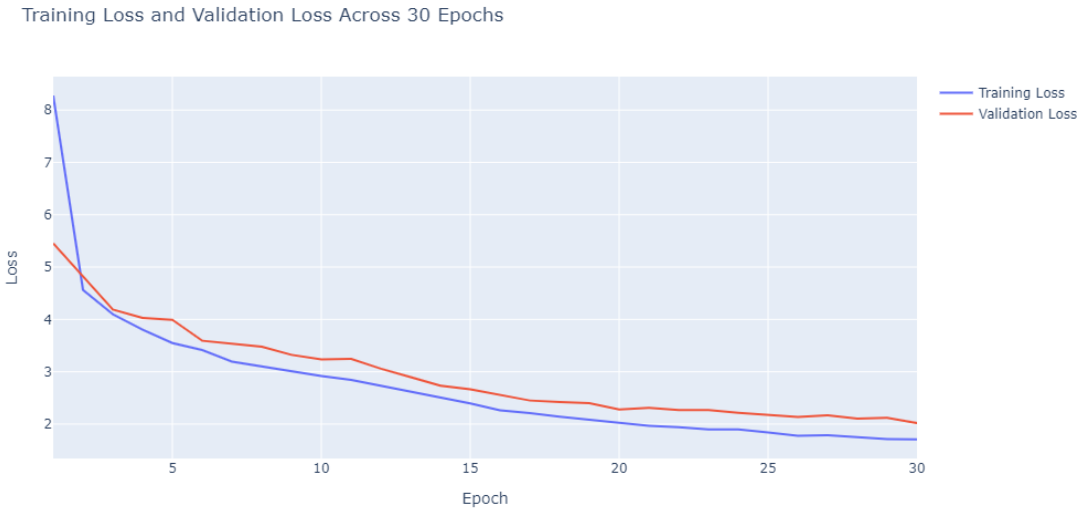

Figure 2: Training and Validation Loss across 30 Epochs

Table 1: ROC-AUC Scores

Label	Accuracy
Insult	0.973251
Obscene	0.970923
Identity Hate	0.960268
Toxic	0.957884
Threat	0.953939
Severe Toxic	0.939353

6 Conclusion

Our proposed model was successfully able to detect and accurately classify toxic comments. Comparatively, while our model does not exceed the performance of the top performing SVMs on the kaggle competition, it still performs better than other submissions such as logistic regression models. This model is to a degree inherently limited by the dataset it was trained on. The small size of the dataset reduces the versatility of the model as it may not be able to accurately classify toxic comments

```
----------------------------------------
Comment: also   yo being a vandal means im a better, holyer person than you @$$e$

Classification:
toxic  :  0.9999282360076904
severe_toxic  :  0.8186097741127014
obscene  :  0.9972718358039856
threat  :  0.3087213933467865
insult  :  0.958332896232605
identity_hate  :  0.4480263888835907

----------------------------------------
Comment: shut up you decaying brained piece of raped s%%t.  go tell your mental diarrea to your mom

Classification:
toxic  :  0.9970620274543762
severe_toxic  :  0.08733643591403961
obscene  :  0.9336186051368713
threat  :  0.010995832271873951
insult  :  0.6815534234046936
identity_hate  :  0.029576383531093597
```

Figure 3: Classification of example test cases

for a variety of purposes outside of gaming. In particular it may struggle with short comments that are only a few words long, due to the fact that the majority of the sentences present in the dataset are long. With the level of subjectivity present in classifying toxicity, it is difficult to produce a universal training set of toxic comments. Regardless, the performance of this model architecture on this particular dataset is indicative of the point that if presented with a larger more diverse dataset, the model could accurately classify toxic comments for a variety of purposes.

For future work, it would be interesting to observe the effect of additional kernels for the Kim CNN. In interest of maintaining a shorter and reasonable training time, only 3 kernels were chosen for this specific implementation. The introduction of more kernels along with the introduction of different kernel sizes could potentially allow for greater accuracy in classification at the cost of computational complexity. Additionally, multiple pre-trained editions of the BERT encoder exist. Using different versions of the BERT encoder may also allow for greater accuracy as well.

7 Division of Labor

The labor was evenly distributed amongst the group members. The exploration and manipulation of the dataset was done together by all group members. Shyam was responsible for the creating the Kim CNNs as well as some of the training. Malav was responsible for the BERT Encoding and visualization of results. Praveen was responsible for the validation function and Model Training. All members contributed equally to planning and writing of the report and presentation.

8 Additional Figures

Figure 4: Toxic Comment Wordcloud

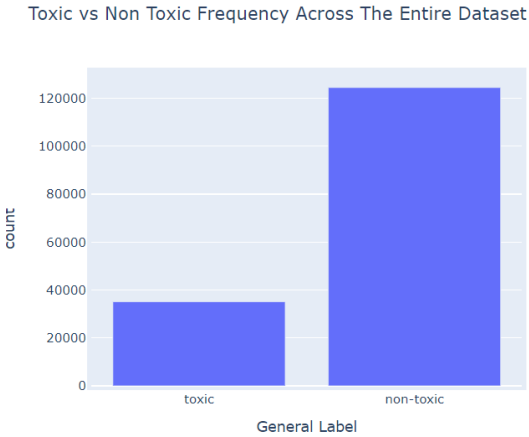

Figure 5: Label Frequency Distribution

References

[1] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[2] A. G. D'Sa, I. Illina, and D. Fohr. Bert and fasttext embeddings for automatic detection of toxic speech. In *2020 International Multi-Conference on: "Organization of Knowledge and Advanced Technologies" (OCTA)*, pages 1–5, 2020.

[3] P. Fortuna, J. Soler, and L. Wanner. Toxic, hateful, offensive or abusive? what are we really classifying? an empirical analysis of hate speech datasets. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 6786–6794, 2020.

[4] Y. Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, Oct. 2014. Association for Computational Linguistics.

[5] S. MacAvaney, H.-R. Yao, E. Yang, K. Russell, N. Goharian, and O. Frieder. Hate speech detection: Challenges and solutions. *PloS one*, 14(8):e0221152, 2019.

[6] Rani Horev. *BERT Explained: State of the art language model for NLP*, 2018. Available at `https://towardsdatascience.com/bert-explained-state-of-the-art-language\-model-for-nlp-f8b21a9b6270`, Uploaded: November 10th, 2018.

[7] I. Tenney, D. Das, and E. Pavlick. Bert rediscovers the classical nlp pipeline. *arXiv preprint arXiv:1905.05950*, 2019.

[8] B. van Aken, J. Risch, R. Krestel, and A. Löser. Challenges for toxic comment classification: An in-depth error analysis. *arXiv preprint arXiv:1809.07572*, 2018.

[9] Y. Zhang and B. Wallace. *A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification*, 2016.