Toxic Emotion Detection Using Multimodal Learning

Abhay Jolly, Harry Preet Singh, Karan Pathania, Neil Mukesh Shah

ABSTRACT

Content consumption on the internet has soared recently due to its widespread availability and consequently, users on the internet are required to identify whether this content is positive or negative in nature. Unfortunately, one might not realize the negativity associated with a particular piece of content until after they have consumed it. In this paper, we propose a method to classify toxic content using multi-modal deep learning and ensemble learning. While there have been several kinds of research into classifying toxic or non-toxic text, we have tried to combine audio and text modalities to classify toxic content based on their respective multimedia features. As there has been limited work done in combining the two modalities, our paper introduces a weighted classification method - weighted on the performance of individual models on single modalities, to combine them, hence extending the efficiency of detecting toxic content on the internet.

1 INTRODUCTION

Internet users are vulnerable to content which is anti-social in nature. The significance of identifying such content prior to the interaction is vital as it apprises the user of the mental risk they would be undertaking if they choose to interact with it. Through this paper, we aim to create a tool that can be used to monitor and measure the toxicity associated with multimedia content on social media platforms. Current approaches tend to be focused on identifying emotions through text and we believe this approach is limited in its scope. The reason is that researchers in human-computer interaction (HCI) have deduced that audio or speech is another important factor while considering the speaker's emotion. A sentence can have different meanings and emotional connotations depending on the way it is said. For example, "I hate you!" can be considered humorous if the the context in which it is said is playful or joking in nature. Of course, it can also be said in a way where the meaning is literal - in an aggressive and hostile conversation.

One of the major issues that such a classification task entails is the limitation of toxic audio data which is available. As noted before, audio plays a vital role in detecting this emotion, and hence creating such a dataset is not only a long task but also the annotation is entirely subjective to the demographics of the annotators[1]. Hence, the data can turn out to be biased depending on the annotator's cultural background and multiple other factors.

Myriad of NLP methods have tried detecting different categories of toxic behavior such as hate speech, offensive, insults, etc. through neural networks, n-grams, and graph-based models[2]. For instance, the Hateful Meme challenge by Facebook was highly appreciated among researchers. Badjatiya et. al. and Gamback et. al. have also used RNNs to detect hate speech in tweets[3].

Although there are limited datasets available on toxic emotion specifically, there do exist other text datasets for subcategories of toxic behavior like OffensEval (2019), Waseem and Hovy (2016) (WH), Davidson (2017). OffensEval contains three levels of annotation for tweets collected by crowd sourcing that are offensive, towards groups/individuals and others.[4] WH also divides tweets into 3 categories namely racism, sexism, and neither, annotated by a third-party review with an inter-rater agreement of 0.84.[5] Lastly, Davidson (2017) categorizes tweets into hate speech, offensive, and neither, with annotation done by hatebase.org.[6]

2 APPROACH

We have approached to implement multi-modal learning by first training a text based LSTM-RNN model on toxic comments. We then evaluated the performance metrics for this model on a hold out text data. Then we used a dataset of toxic audios called HSDVD which we split into 373 and 123 samples for train and test respectively. We trained a CNN model and a Random Forest model on the this audio training set. We test all three models on audio test set to evaluate individual performance metrics. We combined the LSTM-RNN with CNN and then the LSTM-RNN with Random Forest to perform late fusion. We took the weighted fusion values by assigning weights on the ratios of their F1 score performance metric.

2.1 Data Collection

2.1.1 Text Data.

We acquired a data set for building the LSTM-RNN model through the Kaggle competition site[7]. The dataset consisted of 159,571 Wikipedia comments annotated manually by other users. The dataset contains comments that are either toxic or non-toxic and it also contains subcategories of toxic behavior which are 'toxic','severe_toxic', 'obscene', 'threat', 'insult', 'identity_hate'. Through our analysis, we found the data to be highly imbalanced as it contains 80% of non-toxic comments and 20% of toxic comments. We used this dataset to construct our LSTM model for textual analysis.

2.1.2 Audio Data.

We first constructed a dataset by extracting videos of toxic behavior on YouTube and Twitter. The examples of toxic behavior were found by searching for tags specific to racist, hate, and toxic videos. We tried having a similar distribution of 80% to 20% for the non-toxic and toxic classes but due to issues mentioned at the end, we ended up using a public dataset called HSDVD [8]. This data set was annotated and used YouTube audios which we used to train our models and the distribution can be seen in Fig 3. We had 491 samples of such audios of roughly 10 seconds. The data seemed balanced as it contained roughly 50% of toxic and nontoxic audios. Out of these 491 samples, we held out 123 to test all of our models. We also hard-coded the text encoded in the made audios so that we can test the text-based model. All audio models were trained on the remaining 378 samples. Examples of Audio Data set:

- Toxic Audio linked here
- Non Toxic Audio linked here

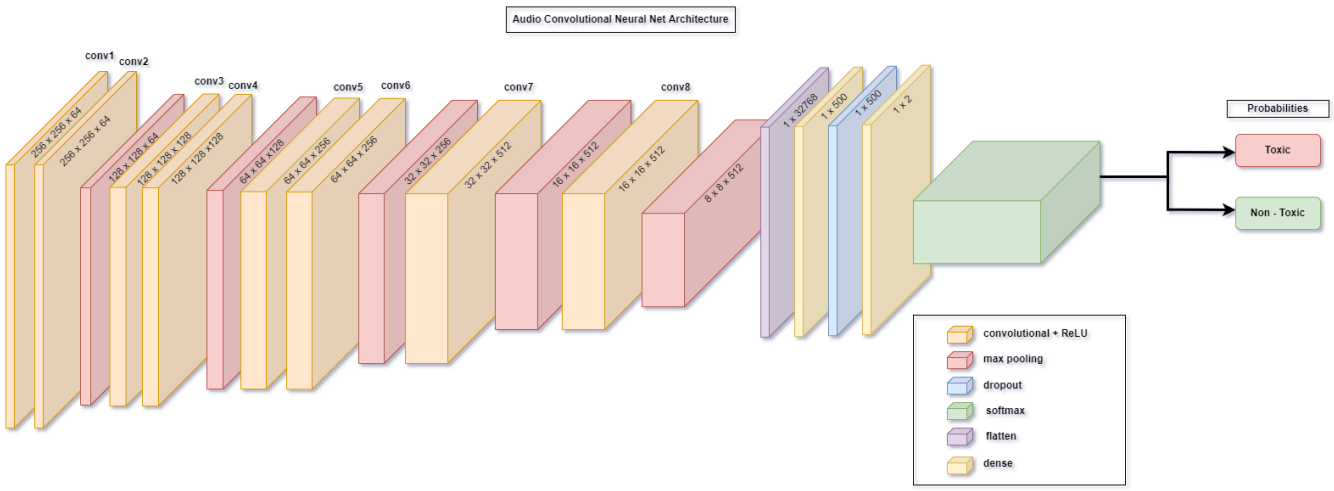

Figure 1: CNN Architecture for Audio Data

2.2 Models Used

2.2.1 Long Short-Term Memory - Recurrent Neural Network (LSTM-RNN).

First and foremost, we processed the text data by normalizing and removing characters which lead to inconsistent languages. The next step of lemmatization was performed to group together different inflected forms of a word so that they can be analyzed as a single item. Lastly, to perform machine learning on this text data we had to convert it to numerical data, and we used tokenization to convert it into its equivalent machine-readable form. To combat variable length in comments, we padded shortfall by zeros and trimmed the longer ones. We used fastText to create pre-trained word embeddings by assigning the vocabulary with the pre-trained word embeddings. For model creation, the reason we decided to go with the LSTM model is because of its efficiency with the Natural Language Processing. In the LSTM model, hidden layer updates are replaced by memory cells which makes them better at discovering long-range dependencies in data. We defined 8 layers in the model with the output layer being a 6 neuron layer giving the probability of each subcategory of toxic. We used the GridSearchCV to obtain the best hyper-parameters. After training, we only considered the maximum probability among the 6 subcategories of toxic behavior which was then ultimately used for fusion with the audio models.

2.2.2 Convoluted Neural Network (CNN).

The audio files for toxic and non-toxic were converted to mel-spectrograms. The reason for this was that the Mel scale depicts how the human ear works because it does not perceive frequencies on a linear scale [9]. The channel used was mono and the audios were averaged to five seconds and then used for spectrogram generation. Each spectrogram image has the size 256 x 256 for consistency. The converted files were divided into toxic and non-toxic folders. In Fig 4 and Fig 5 an example of toxic and non-toxic respectively shows some difference. This might be a result of different emotions similar to aggression and anger experienced by the person saying them

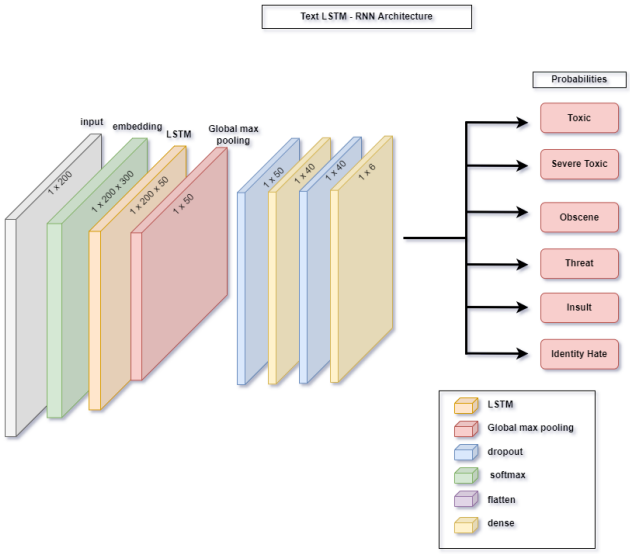

Figure 2: LSTM Architecture for Text Data

We decided to use CNN as it is a powerful algorithm for image processing and especially for object detection. [10]. We wanted to explore how CNN would perform in detecting toxicity from mel-spectrogram images of audio clips. We tried to take inspiration from VGG-19 CNN architecture which has 19 convolutional layers and just focuses on most essential features of convolutional neural networks. Since we had limited computational power, we were unsuccessful in running VGG-19 architecture. Therefore, we created a 17 layer convolutional neural network which was similar to VGG-19 and must be faster computationally for our machines.

We divided training data into 80/20 split for validation. We did not do any data augmentation or rotation as we are using mel-spectrograms, doing rotation or augmentation changes the meaning

of the image. We also shuffled train and validation data between each epoch to have minimum overfitting.

We had a total of 21,070,878 trainable parameters upon which we trained our mel-spectrogram images. The best hyper-parameters were found and the dataset was trained using the training mel-spectrogram images.

2.2.3 *Random Forest Classifier (RF).*

Random Forest is useful for such binary classifications tasks as it implements ensemble learning by calculating vote for prediction from individual decision trees. The higher the number of trees in the forest the more confident a prediction is.[11] Topped with the efficiency of computation and faster processing time, we decided to go with random forest to classify audio into toxic and non toxic on basis of their midterm features. To use random forest with audio, we first used pyaudioanalysis library in python to obtain midterm features from the audio files for all the categories. The total number of features for each audio totaled to 136. The midterm features contained data like mean spectral energy, entropy, MFCCs, etc. Then we performed standardization on this compiled data to train our Random Forest ensemble. .

3 EXPERIMENTS & RESULTS

3.1 LSTM-RNN

Running LSTM-RNN model on a 25% hold out data of Wikipedia comments gave a precision of 83.68% with AUC as 98.39% and a recall of 65.21% and a total F1 score of 73.30%. Interesting thing to note is that by manually testing this model we found that it detects toxic behaviour when there is an explicit word in the sentence whereas sentences which are toxic but don't contain a cuss word are not detected as toxic.

Running LSTM-RNN on the test audio dataset gave an F1 score of 72% which seems not surprising as the test audio data is also composed of sentences where abuses were not used. On the confusion matrix in Fig 7, we can see that the precision was a little better than recall, which was expected due to dataset imbalance although the value is still high.

3.2 CNN

Running the the CNN model on training data with 20 epochs resulted in 97.46% and 96.74% on validation. Running the CNN model on the test dataset gave an accuracy of 92% and F1 score of 0.92, which was expected due to this model's high accuracy on validation dataset as well. In Fig 6, precision is lower than recall which shows that the model predicts right more on quantity rather than giving quality results.

3.3 Random Forest

We tried PCA with 2 components for all the midterm features but due to high information loss given by explained variance of about 40%, we decided to not perform PCA. The parameters received from Random Search CV were used in our final model training for Random Forest. This model gave an accuracy score of 84% on the test dataset. The precision and recall for Random Forest can be seen in the Fig 8. The slightly higher value of precision than recall shows that this model gives less false negatives than false Positives.

Model	F1 Score	Weight
LSTM-RNN	72%	0.46
Random Forest	84%	0.54

Table 1: LSTM RNN and Random Forest Comparison

Model	F1 Score	Weight
LSTM-RNN	72%	0.44
CNN	93%	0.56

Table 2: LSTM RNN and CNN comparison

3.4 Weighted Fusion

Two models were considered for weighted fusion, they are as follows:

(1) LSTM-RNN and Random Forest
(2) LSTM-RNN and CNN

The purpose here was to perform late fusion for our two modals and find which of them provides better results. The weight assigned to each classifier was calculated using its accuracy with the test dataset and finding a ratio between the classifiers used. These weights were multiplied by the probability of toxicity received from predictions and the weighted toxic probabilities for the models were added.

A threshold value was used for predicting, the best value required a hyper-parameter tuning step in which a range of threshold values was used.

3.4.1 *LSTM-RNN & Random Forest.*

In Table 1, we can see the calculated accuracy and weights. Using these weights, we received a threshold value of 0.25, which gave an accuracy of 87% and F1 score of 0.87. The value for precision is higher and similar recall as seen in Fig 10. This shows that model is giving similar quantity and quality results. Therefore weighted fusion resulted in a better result for this model and we can conclude that audio and text together improves our result.

3.4.2 *LSTM-RNN & CNN.*

In Table 2, we can see the calculated accuracy and weights. Using these weights, we received a threshold value of 0.55, which gave an accuracy of 93% and F1 score evaluated was 0.93. In Fig 9 we can see the recall is again lower than precision, therefore even this model gives a better quality result than quantity. Therefore weighted fusion resulted in a little better result for this model and we can conclude that audio and text together improves our result.

4 DISCUSSION

The accuracy given by CNN was quite high compared to the Random Forest classifier. This could be due to the small dataset used for audio. Replacing our self constructed and annotated dataset for audio with the one found at [8] made a bigger difference than we expected, we did try using both but the results were not as impressive. One reason could be our search being specific to audio which

had some particular words, whereas the dataset found had audios which had some social cues of emotions like anger, aggression etc. Our text dataset, even though being a little unbalanced gave us better results than we expected, specifically when fused with other classifiers. The audio dataset can be expanded in the future, though a good prediction can be found with these models. This can be used to find toxic speech over social media, specifically in meta verse which is expanding and people have reported online abuse [12].

5 CONCLUSION

Moderating and monitoring the content on the Internet is a vital deterrent for spread of toxicity in the digital environment towards other. As concluded from our research, text alone cant' be the sole parameter in identifying any kind of human emotion, we have to incorporate audio or any other kind of modality that provides us with some knowledge about the emotional or physiological context of the speech. Further, we also believe that due to limited toxic data sets that are available, our research can certainly be extended to incorporate more wide scaled samples. Another improvement that can be done is to implement early fusion by combining embedding from the text and audio features to test if it gives better results.

REFERENCES

[1] J. H. Park and P. Fung, "One-step and two-step classification for abusive language detection on twitter," *Proceedings of the First Workshop on Abusive Language Online*, 2017.
[2] C. Wang, "Interpreting neural network hate speech classifiers," *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, 2018.
[3] M. G. Constantin, D.-S. Parvu, C. Stanciu, D. Ionascu, and B. Ionescu, "Hateful meme detection with multimodal deep neural networks," *2021 International Symposium on Signals, Circuits and Systems (ISSCS)*, 2021.
[4] M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, and R. Kumar, "Predicting the type and target of offensive posts in social media," *Proceedings of the 2019 Conference of the North*, 2019.
[5] Z. Waseem and D. Hovy, "Hateful symbols or hateful people? predictive features for hate speech detection on twitter," *Proceedings of the NAACL Student Research Workshop*, 2016.
[6] T. Davidson, D. Warmsley, M. Macy, and I. Weber, "Automated hate speech detection and the problem of offensive language," *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 11, pp. 512–515, May 2017.
[7] "Toxic comment classification challenge."
[8] A. Rana and S. Jha, "Emotion based hate speech detection using multimodal learning," 2022.
[9] A. Goni, "How to detect covid-19 cough from mel spectrogram using convolutional neural network," Jul 2021.
[10] P. Mishra, "Why are convolutional neural networks good for image classification?," Jul 2019.
[11] G. Louppe, "Understanding random forests: From theory to practice," Jun 2015.
[12] T. Wang, "Online abuse in the metaverse untangled," Feb 2022.

APPENDIX

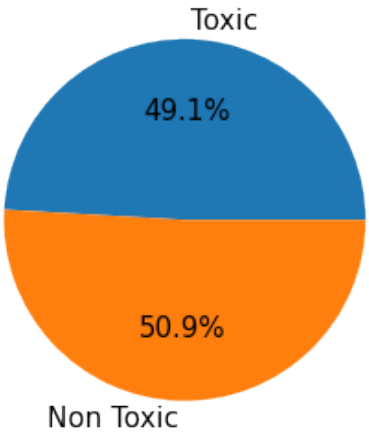

Figure 3: Audio Data Distribution

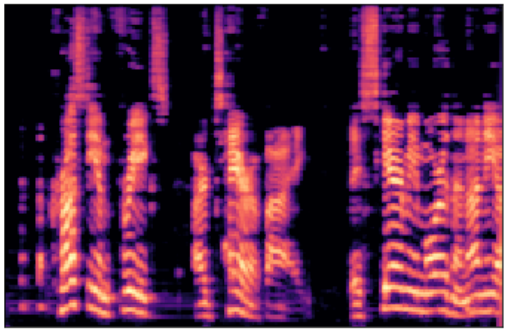

Figure 4: Spectrogram for Toxic

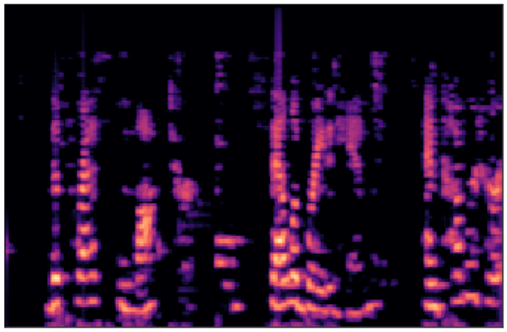

Figure 5: Spectrogram for Non toxic

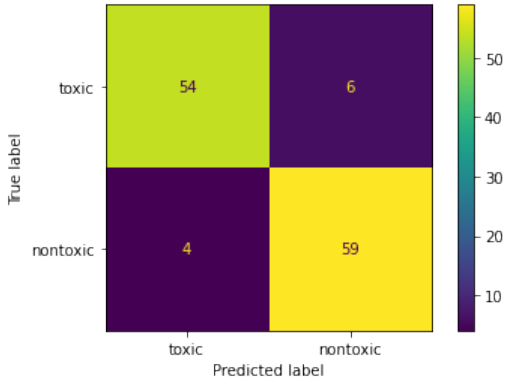

Figure 6: Confusion Matrix for CNN

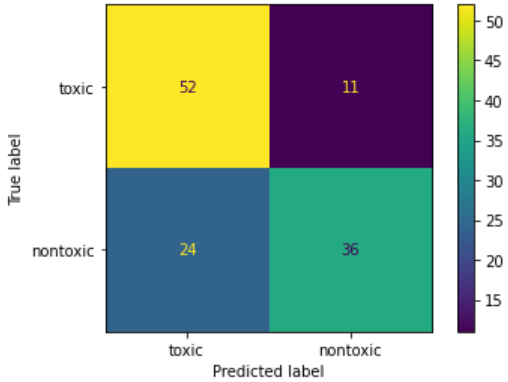

Figure 7: Confusion Matrix for LSTM-RNN

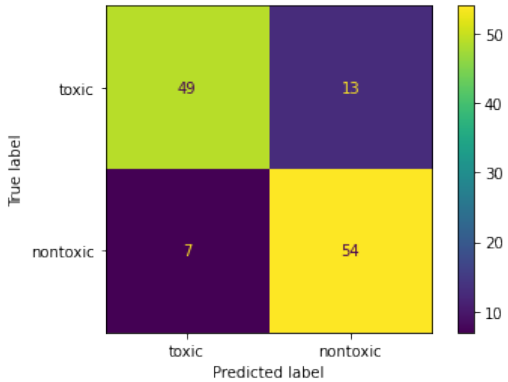

Figure 8: Confusion Matrix for Random Forest

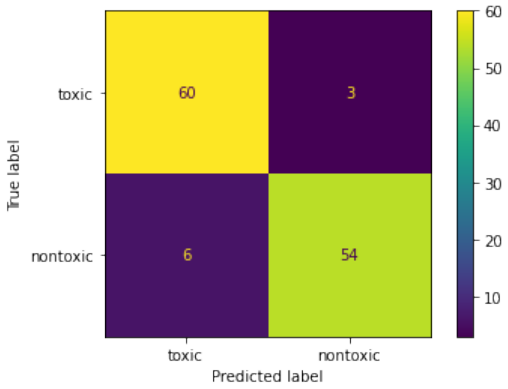

Figure 9: Confusion Matrix for CNN and LSTM-RNN

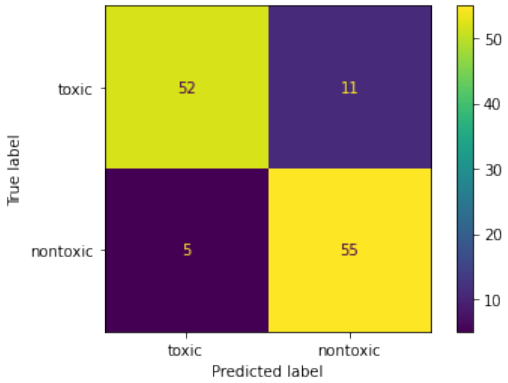

Figure 10: Confusion Matrix for Random Forest and LSTM-RNN