# Comment Toxicity Model Prediction

AJAY JATPROL and DEEKSHITTH VEGI

**ACM Reference Format:**
Ajay Jatprol and Deekshitth Vegi. 2024. Comment Toxicity Model Prediction. 1, 1 (April 2024), 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

## 1 INTRODUCTION

The aim of the project is to address the challenge of online toxicity by picking out and clustering toxic comments. The prevalent issue of negative online behaviour can effect a individual and communities.Thus, it is important to push-button the method of handling the comments and soothe such behaviours to establish a safe and better digital environment. This project work belongs to the field by creating and developing a deep learning model that productively find out different forms of toxicity in online comments, which could be connect to a content handling systems to push-button the process of filtering or flag unfitting content.

· Problem Statement: Online networks and platforms have become a key space for interaction Discussion. However, these platforms are often filled with the presence of toxic comments which can harm an individual and also can silence their voices. This model mainly focuses on developing a deep learning model to spot toxic comments and categorize such comments, which is an important step in abating online content to develop safer digital platforms.

· Relevance: With the analog-to-digital conversion, handling the caliber of action has become a superior concern. Toxic comments can also lead to discrimination, cyberbullies, and extended negativity, which leads to the internet being an uninviting place for the members who are using it. This project will take care of solving the issue by providing a push-button solution that helps human moderators.

· Contribution: The study will handle the research of employing the dl techniques and methods to get better accuracy and Productivity of toxic comment classification, which helps us to get better real-time comment moderation. [6]

### 1.1 Terminology

**Adam Optimizer:** It is a popular optimization in deep learning and machine learning it is also known as Adaptive Moment Estimation. It has 2 techniques AdaGrad and RMSprop and it helps to adjust the learning rates of each parameter and converge is faster and able to take care of distinct types of data.

**Balanced Accuracy:** The metric is mainly used to evaluate the performance of the classification model and when we are handling the imbalanced datasets it will consider both true positive and true negative rates of each class and show how well the model is performing after training.

**Bidirectional LSTM :** This can capture information in both directions, forward and backward.

**Gradio:**Gradio is a library in Python used for deployment and we can share the machine-learning models through web interfaces.

## 2 RELATED WORK

Former tests to solve the problem of toxic comments have used different natural language processing techniques and range from a simple bag of words model to complex and better deep learning methods like Recurrent neural Networks and Convolutional Neural Networks and the study of the jigsaw dataset challenge, has advance text vectorization have other sequence models which help to solve the motto of the project.

Background: The previous studies have mostly focused on feature engineering and other machine learning models like classical ones and they used SVM and naive Bayes for text classification, but this method will fall short in capturing the semantic nuances of language In this project at the initial step used with basic sequence model with 7 layers starts with embedding and next did the vectorization and then used bilstm and lstm gates for longer dependencies range and at last I use dense layer model.

Building Upon Past Work: The research is set in the evolution of the NLP technique and using the earlier works in the realm of neural networks and using the LSTM gates have shown the best results in understanding the dependencies in the text data.

### 2.1 Dataset Explanation:

Train.csv: The dataset is used to train the model and it contains several columns and the labels might indicate a comment is toxic, severely toxic, a threat, an insult, etc. Test.csv: The CSV file contains the test data and it has the same structure as the train.csv but it won't have label columns in it and it is used to evaluate the performance and the model. Test-labels.csv: It allows you to evaluate the accuracy of your model predictions by comparing them against these true labels. Train.csv is used to teach the model to recognize the patterns in the toxicity in comments and the test.csv is used to help the model to learn the patterns and test-labels.csv -how well the model did with the prediction.

The heat map shows a strong correlation in color scale and if the 2 categories have 1.0 means they are strongly correlated and if they have 0 means no correlation between them

The toxic category is perfectly correlated with itself because it has a 1.0 value and the threat and column insult have 0.74 it correlates high means the text classified as a threat is also classified as an insult. The severe toxic has a strong correlation relationship with all the other categories and this suggests that the text is classified as having one or more of the other categories. Identity hate has a weak correlation with the other categories and the classification is less likely for other categories.

Authors' Contact Information: Ajay Jatprol; Deekshitth Vegi.

ACM XXXX-XXXX/2024/4-ART
https://doi.org/10.1145/nnnnnnn.nnnnnnn

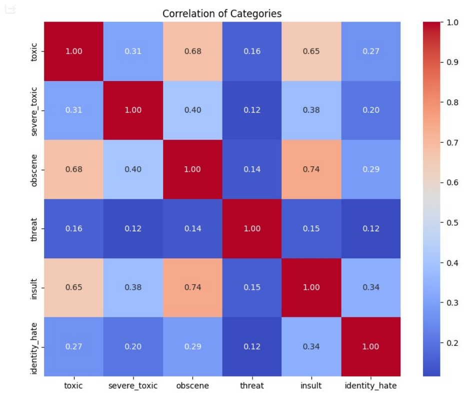


Fig. 1. Heat Map

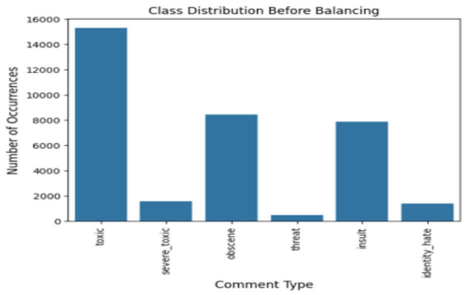


Fig. 2. Dataset Before Balancing

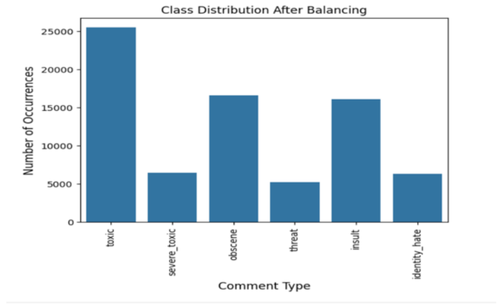


Fig. 3. Dataset After Balancing

## 2.2 Data preprocessing and Data balancing :

To prepare the data for analysis. started with replacing any missing comments with placeholder text and converting lowercase for consistency and then used heatmap to see the correlation values between the labels. ISSUES: The dataset has imbalance categories and toxic has a higher data value than other labels so I resolved this issue, by increasing the number of samples for underrepresented categories and a process known as oversampling so that each category has the same number of samples and if the dataset has not balanced then the model will go underfitting and get best accuracy which is not true. The imbalance class in the dataset leads to a classification problem using the oversampling technique. In the initial step the target (`y_train`) datasets and feature (`X_train`) datasets are reset to deal with data manipulation.the feature dataset (`X_train`) has only single column named as `'comment_text'` to ease manipulation.The code calculates the median of the total number of samples per class in all categories present in the target dataset (`y_train`) and stores it as `target_samples`.

For every class in the target dataset, we take some samples randomly with relate them with the existing sample until the no of samples matches with the `target_samples` Then oversampled data are stored in a list `oversampled_data`, The original data is concatenated with the oversampled data to create an oversampled training dataset to train the model. Finally, the oversampled datasets are shuffled using the `sklearn. utils.shuffle` function to mix the data properly for model training.

## 2.3 Dataset Split

**Training Set:** The set is the biggest portion of the dataset and data that the model learns to make predictions and pattern identifying, In the project we used 80 percent for training. The dataset has 155,000 samples: The train set would contain 124,000 samples. The test set would contain 31,000 samples.

**Test Set:** The test set is used to evaluate the model performance and generalize the capability of the trained model and we can see how the model will perform in real-time data in the project we took 20 percent as test data.

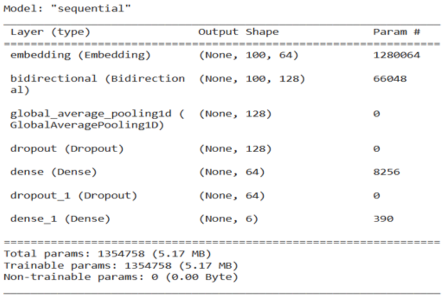
**Validation Set:** The validation is used to fine tune the model parameters and validation dataset can be separate but in our case we used same data used for the test data and it helps model to make decisions without biases and the percentage is 20 .model can learn from one data train and fine tune with parameters against another validation and performance is evaluate by unseen data test and it helps to over fit the model on new data.[2]

## 3 METHODS

Study Design: The Methodological process of the model involves: 1. Preparing the dataset 2. Designing the model architecture 3. Training and validating data 4. The dataset was pre-processed for consistency in text and to address class imbalance. The model architecture for the deep learning model used TensorFlow and Keras and below is the model summary and model in detail.

- **Text Vectorization:**strings are converted into vocabulary indices.
- **Embedding:**vocabulary indices are converted to vectors.
- **Bidirectional LSTM:**used to capture data in both forward and backward.
- **GlobalAveragePooling1D:** Used for dimensionality reduction by taking the average over the time dimension.
- **Dropout:** This layer is used to prevent overfitting by regularization.
- **Dense:**fully connected layer that outputs a vector with real.
- **Dense:** This is the final layer with sigmoid activation.

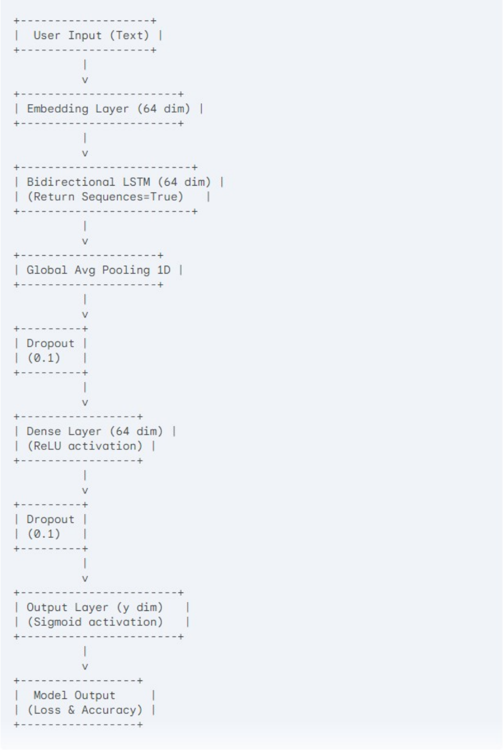
The optimizer used is adma and the learning rate is 1e-4 and the loss function is binary cross-entropy and it is suitable for binary classification.



```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 100, 64)           1280064

 bidirectional (Bidirection  (None, 100, 128)          66048
 al)

 global_average_pooling1d (  (None, 128)               0
 GlobalAveragePooling1D)

 dropout (Dropout)           (None, 128)               0

 dense (Dense)               (None, 64)                8256

 dropout_1 (Dropout)         (None, 64)                0

 dense_1 (Dense)             (None, 6)                 390

=================================================================
Total params: 1354758 (5.17 MB)
Trainable params: 1354758 (5.17 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Fig. 4. Model summary

The model is saved and then loaded in a Python library known as gradio and it is a simple interface for the users to interact with the trained model on the web. when we type the toxic comment in the input comment box then you will get the result of the toxicity of the comment in the result box along with the probabilities of the toxic labels.



```
+------------------+
|  User Input (Text) |
+------------------+
          |
          v
+----------------------+
| Embedding Layer (64 dim) |
+----------------------+
          |
          v
+------------------------+
| Bidirectional LSTM (64 dim) |
| (Return Sequences=True)    |
+------------------------+
          |
          v
+-------------------+
| Global Avg Pooling 1D |
+-------------------+
          |
          v
+---------+
| Dropout |
| (0.1)   |
+---------+
          |
          v
+----------------+
| Dense Layer (64 dim) |
| (ReLU activation) |
+----------------+
          |
          v
+---------+
| Dropout |
| (0.1)   |
+---------+
          |
          v
+----------------------+
| Output Layer (y dim)   |
| (Sigmoid activation)   |
+----------------------+
          |
          v
+----------------+
|  Model Output     |
| (Loss & Accuracy) |
+----------------+
```

Fig. 5. Model Architecture

## 4 BASELINE MODEL

To comparing the performance of the proposed Rnn model withe the baseline model we considered onevsrest classifier because it can deal with the multi-labeled classification that involve train single classifier per class and the model is well suited for the the text classification task.

## 5 RESULTS

### 5.1 Proposed Model vs Baseline Model

The baseline model onevsrest classifier has achieved an accuracy of 91.92 train and for test data 0.90 and precision and F1 -scores differ based on classes and 1,3,5 need improvement. Proposed LSTM Model(RNN): the model has an accuracy of 95 for train and for the test data I got around 95 and it has similar patterns in classes 1,3,5.

Rnn model has a slight edge over the baseline model and works better with test data.

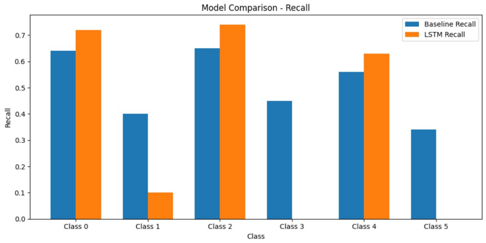


Fig. 6. precision

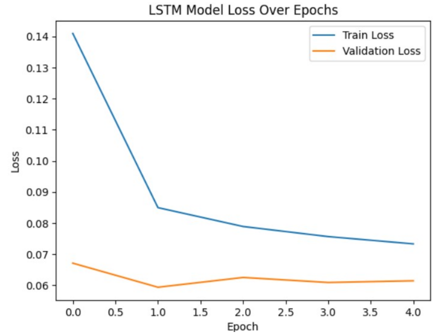


Fig. 7. Comparison

### 5.2 Baseline model

The baseline model performance is best on some classes in the dataset and we imported the model from multi-class and we got the accuracy of 0.90 which is best. below is the classification report where we can see the model performance.

The F1-scores in classes 1,3,5 suggest that model is having issue with the complex patterns that are hard to learn for the model.To resolve that issue we use the proposed model Rnn which can understand the complex patterns in the dataset.

Table 1. Classification Report for Baseline Model

| Class | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| 0 | 0.90 | 0.64 | 0.75 | 3056 |
| 1 | 0.49 | 0.40 | 0.44 | 321 |
| 2 | 0.91 | 0.65 | 0.76 | 1715 |
| 3 | 0.45 | 0.45 | 0.45 | 74 |
| 4 | 0.81 | 0.56 | 0.67 | 1614 |
| 5 | 0.60 | 0.34 | 0.43 | 294 |

Table 2. Classification Report for Proposed Model

| Class | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| 0 | 0.83 | 0.72 | 0.77 | 3056 |
| 1 | 0.62 | 0.10 | 0.18 | 321 |
| 2 | 0.82 | 0.74 | 0.78 | 1715 |
| 3 | 0.00 | 0.00 | 0.00 | 74 |
| 4 | 0.73 | 0.63 | 0.68 | 1614 |
| 5 | 0.06 | 0.00 | 0.00 | 294 |

### 5.3 Proposed Model

The validation accuracy is high which suggests that the model is effective for identifying the toxic comment but it also leads to model overfitting.

The model used for toxicity detection was trained for 5 epochs and in the process of training the loss for the model is reduced from 0.1299 to 0.0745 in the process we can observe that performance is increased and the accuracy for the model is raised to 0.9538 over the last epoch and the model ability to make correct predictions also gotten better and in the case of validation the loss is decreased to 0.9030 to 0.0861 and in the case of the accuracy is remained high, the range is 0.9906 to 0.9907. The balanced accuracy for the classes like toxic and obscene is getting high while for the remaining ones like threat and identity hate is lower accuracy and is around 0.5 to 0.52 and the average balanced accuracy for the overall performance of the model is decent.

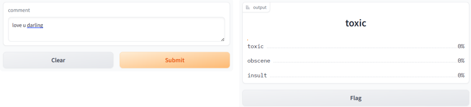
The validation remains constant and does not fluctuate

- The model learned to generalize the data quickly.
- The learning rate might be too small but not much changes in loss value.
- model capacity is appropriate for complexity of task and it is neither over-fitting nor under fitting.

- **Embedding Layer** It has $(\text{max\_features} + 1) \times \text{output\_dim}$ parameters, where `max_features` is the max no of tokens in the vocabulary and `output_dim` is the dimensional of the output space. In this one, it has $(20000 + 1) \times 64 = 1,280,064$ parameters.
- **Bidirectional LSTM Layer (Bidirectional LSTM):** It has $4 \times ((\text{input\_dim} + \text{output\_dim}) \times \text{output\_dim} + \text{output\_dim})$ parameters, where `input_dim` is the dimensional of the input space and `output_dim` is the dimensional of the output space. In this one, it has $4 \times ((64+64) \times 64+64) = 66,048$ parameters.
- **Global Average Pooling1D** The layer will reduce the dimension of the input by taking input as average over time dimensions.no parameters for training
- **Dropout Layers (Dropout):** The dropout layers randomly take some input fraction units to make it zero and it will help us to prevent over-fitting and no new parameters are introduced in this layer.
- **Dense Layers (Dense):** The layers are known as fully connected layers that connect the input data. every dense layer has $\text{input\_dim} \times \text{output\_dim} + \text{output\_dim}$ parameters, where `input_dim` is the dimensional of the input space and `output_dim` is the dimensional of the output space. In this layer, the first dense layer has $(128 \times 64) + 64 = 8,256$ parameters, and the second dense layer has $(64 \times 6) + 6 = 390$ parameters.

The total no of parameters in the trained model is adding all the parameters of all layers in the sequential model $1,280,064 + 66,048 + 8,256 + 390 = 1,354,758$.

### 5.4 Test case


comment

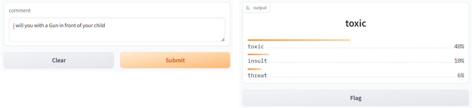
love u darling

output

toxic

toxic 0%
obscene 0%
insult 0%

Fig. 8. test-1


comment

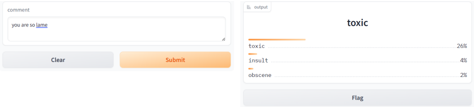
i will you with a Gun in front of your child

output

toxic

toxic 48%
insult 10%
threat 6%

Fig. 9. test-2


comment

you are so lame

output

toxic

toxic 26%
insult 4%
obscene 2%

Fig. 10. test-3

## 6 DISCUSSION

Model Evaluation: the model has performed extremely well and demonstrated excellent performance on the validation set and high accuracy can lead to model overfitting and class imbalances. The limitations of the study can include potential bias in the oversampling method and the need for a more diverse dataset to enhance model generality. The model shows great performance with an increase in accuracy after we balanced the dataset but we highlight the importance of class imbalance in the dataset and we need to examine the model in real-time for better results we can also use the model like Bert's transformer-based model.[3]

## 7 REFERENCES

(1) **1** Wulczyn, E., Thain, N., & Dixon, L. (2017, April). Ex machina: Personal attacks seen at scale. In Proceedings of the 26th international conference on world wide web .

(2) **2** 2.Jigsaw Toxic Comment Classification Challenge. Kaggle. (n.d.). Retrieved from https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge

(3) **3** Adadi, A., & Berrada, M. (2018). Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). IEEE access, 6, 52138-52160.

(4) **4** .Patil, A. P., Mohammed, A., Elachitaya, G., & Tiwary, M. (2019, October). Practical Significance of GA PartCC in Multi-Label Classification. In TENCON 2019-2019 IEEE Region 10 Conference (TENCON) . IEEE.

(5) **5** Park, J.H., Fung, P.: One-step and two-step classification for abusive language de-tection on twitter. In: Proceedings of the Workshop on Abusive Language

(6) **6**.Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., & Joulin, A. (2017). Advances in pre-training distributed word representations. arXiv preprint arXiv:1712.09405.

https://www.overleaf.com/project/66270cbaa6c01eae79fac98b