DESTRUCTIVE LANGUAGE RATER

Athira Indrakumar
ECE MEng
University of Toronto
Toronto, Canada
athira.indrakumar@mail.utoronto.ca

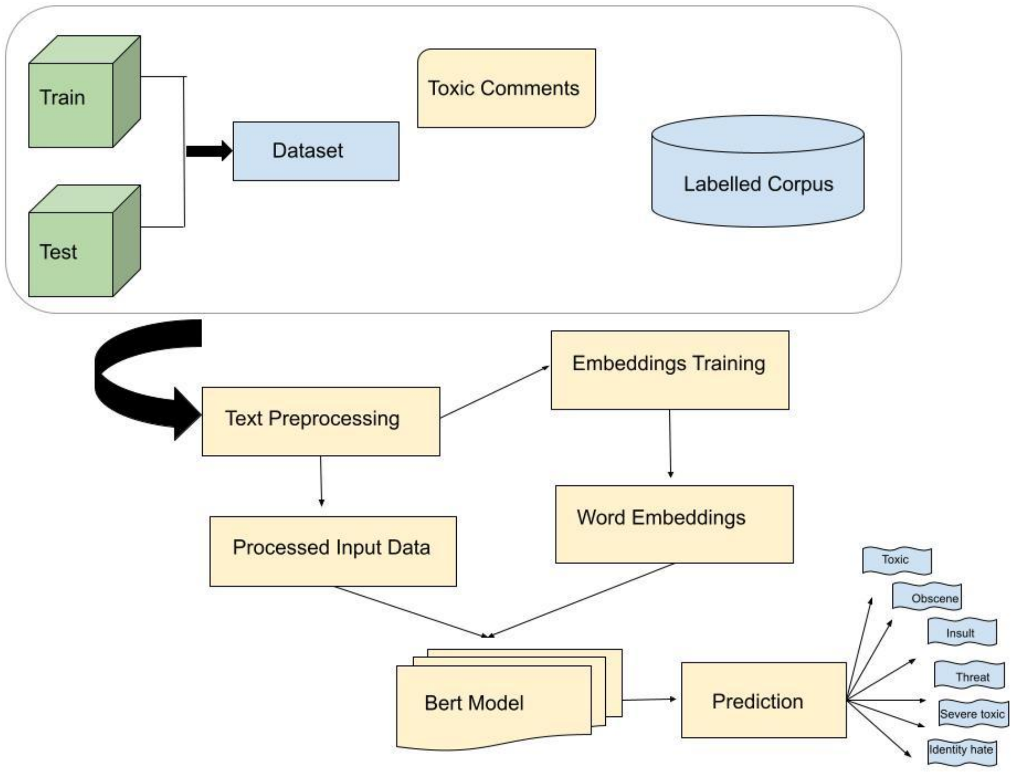
Vishnu Priya Rajendran
ECE MEng
University of Toronto
Toronto, Canada
vishnupriya.rajendran@mail.utoronto.ca

1. INTRODUCTION

Nowadays, contributing content online has grown to be a primary factor. It has become the main connecting factor for people around the world. Every day, social networking sites post millions of pieces of material. The contents are a mixture of compliments, opinions, harmful material, etc. The quality of human existence is greatly impacted by the toxic material, which also has harmful direct and indirect effects. To protect the wellness and comfort of their users, all social media firms employ a variety of techniques to remove these harmful words from their platforms. It is now difficult to distinguish between harmful and normal material due to the vast volume of data[1]. Our project's goal is to create a machine-learning model that can distinguish between different message types. Some of the types which we mainly consider classifying the language are toxic, obscene, severe toxic, identity hate, threat, and insult. We are using the BERT model where we use a pre-trained model for our project and also check the performance of the baseline model.

2. ILLUSTRATION

The below picture gives an overall idea of how the model is trained using the BERT model. The sentences are preprocessed before converting them to embeddings and these embeddings are fed to the model for multi-class prediction. We have 6 labels to which our sentences should be classified. Hence, we are using the input_ids and attention masks from the Bert tokenized sentences for multi classification.



3. BACKGROUND AND RELATED WORK

A. *BERT based model with transformers*
Hong Fan et al. [1] use the "Toxic Comment Classification Challenge" data to fine-tune the model and to evaluate the use of two other datasets that were collected from Twitter in two timelines. They compare four models BERT, Multilingual BERT, RoBERTa, DistilBERT and conclude on which model performs better. They conclude the BERT based model performs better followed by Multilingual BERT, DistilBERT and RoBERTa.

B. *Deep learning models for toxic classification*
Kevin Khieu et al. [2] uses deep learning models like LSTM, SVM, Multilayer Perceptron, and CNN to detect and classify toxic comments. They use metrics like accuracy, recall, precision, F1-Score and specificity to compare the model's performance. For both Multi-Label Classification and Binary Classification LSTM performs better than other models. However, for character level classification, their result shows that CNN performs better.

C. *Convolutional Neural Networks for Toxic Comment Classification*
Spiros and the team [3] worked on the study of recent approaches for the classification of texts. They compared the performance of different models for toxic comment classification and

concluded with sufficient evidence that a CNN model can overtop other most established procedures. The results are motivating and promising for further work on CNN-based models for text classification in the future.

4. DATA AND DATA PREPROCESSING

We used a toxic comment classifier dataset from kaggle. Along with that, we retrieved a few data from social media platforms and labeled them accordingly to align with the current dataset. The dataset from Kaggle contains examples of Wikipedia comments that were rated to different forms of toxicity. The data is in the form of a CSV file with column headers id, commentText, toxic, severeToxic, obscene, threat, insult, identityHate, where the latter 6 column labels are of boolean values. After collecting our custom dataset, we labeled the sentences according to the above boolean headers. In total we have 160595 rows of data.

All the NaN values were removed as part of the preprocessing. The dataset is then balanced between the toxic and non-toxic data to handle overfitting. We are not balancing between the 6 labels as this results in a decrease in the number of samples. This is a multi-labeling classification task. We divided the dataset to train, validate, and test in a 60- 20-20 ratio with unique values to avoid overfit. For word-level analysis, we used padding for all the sentences with 0's padded till the length of the largest sentence in the dataset batch. Data loaders were used on these datasets to produce data in batches for training, validating, and testing according to the input batch size. Bert tokenizer was used to tokenize the sentences which produced input_ids and attention_masks and those were used for the multi-label classification.

5. ARCHITECTURE AND SOFTWARE

We used the BERT (Bidirectional Encoder Representations from Transformers) model "distilbert-base-uncased" for our project. This model is pre-trained on raw text without any labels by Masked language modeling (MLM) and Next sentence prediction (NSP). Our model takes a sequence of tokens as input and classifies the text. We first train the BERT model with our train dataset and use validation dataset to validate the working of the model. The parameters we set are displayed in the table below.

HYPERPARAMETERS	**VALUE**
Optimizer	Adam
Learning Rate	0.001
Loss Function	Mean Square Error

Epoch	25

<u>Other details</u>

Batch Size	32
Number of layers(transformer blocks)	12
Hidden layer parameters	768
Number of attention heads	12
Total number of parameters	110M

From the available data, training, test and validation datasets are created using BERT Tokenizer which converts each sentence to input ids, attention masks and labels. These are single array tensors which are fed to the training model. Based on the datasets, data loaders are also created using the configured batch size. We created a custom model on top of the pre-trained BERT model where we use a linear layer and a sigmoid activation function is also used. To the linear layer, we pass the number of classes/labels (in our case, 6 in number) to call the pretrained BERT model with the BERT hidden config size. Finally, from the predictions, we are assigning scores to each label in the dataset for the sentences. We have set a threshold value of 0.5. We consider a sentence non-toxic if all the label values are less than 0.5. If all the values are greater than 0.5, we fetch the highest probability and that the label corresponding to that value is considered the sentence class.

Additionally, we are using Gradio to interface our results. Our gradio model has 3 fields - comment text, BERT model probability and baseline model probability. Based on the sentence provided, it will be tokenized and the scores corresponding to that sentence is fetched from both the models. According to the above logic, the sentence will be classified to toxic, non-toxic or to any of the 6 labels. Some examples related to the gradio interface can be seen in section 8 (Qualitative Results).

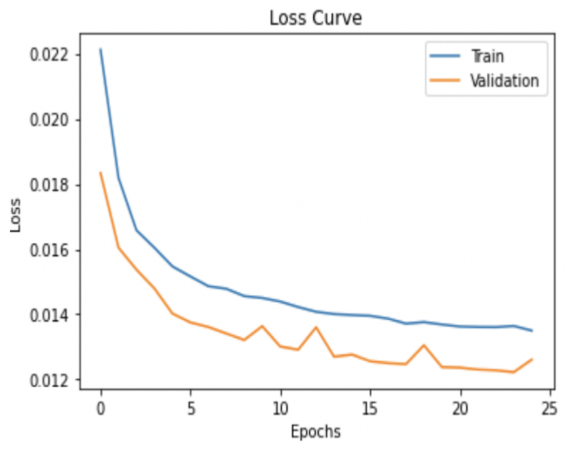
6. <u>BASELINE MODEL</u>

Our baseline model is a machine learning model that converts the input words to vectors. We are using glove embeddings(pre-trained word vectors) to convert the tokens to embeddings. The glove object contains the index (also called word token) for most words in the data set. For those words that do not occur in GloVe's vocabulary range, we substitute it for the last word in the vocabulary. The model computes the average of those word embeddings in a given sentence.

This is fed to a fully connected layer which produces a scalar output with sigmoid activation to represent the probability that the sentence is in each toxicity class. Based on the probabilities obtained, the sentences are classified into different toxicity classes. We are using the same hyperparameters as Bert model for training and tuning with appropriate batch sizes. We chose this baseline model as it is simple and to have a good standard to measure our model's performance and accuracy.

7. QUANTITATIVE RESULTS

Please find the loss curves and accuracy curves for BERT and baseline models below.



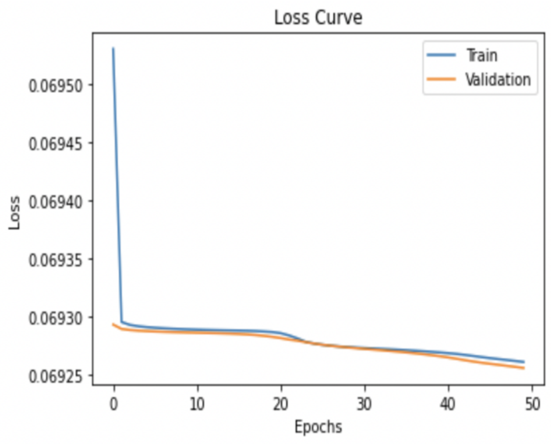
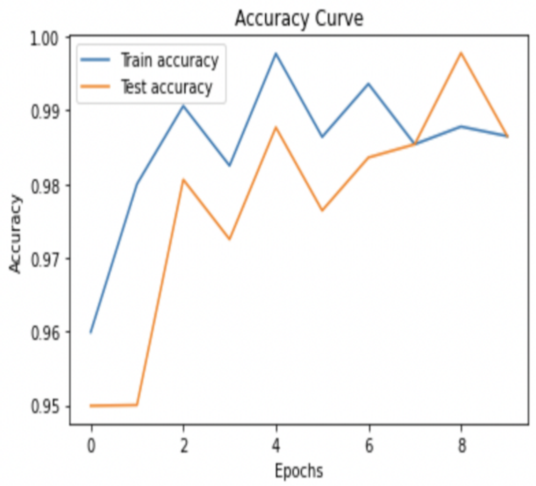


Fig: Train v/s Validation loss of bert model Fig: Train v/s Validation loss of baseline model



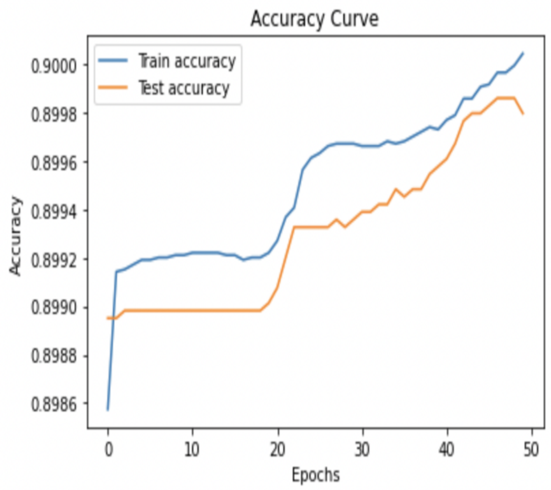
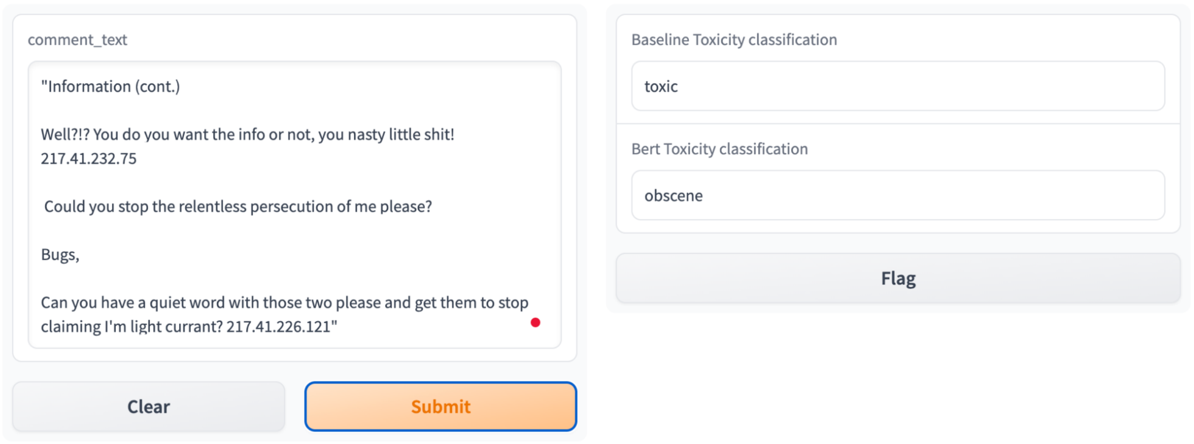


Fig: Train v/s Test accuracy of bert model Fig: Train v/s Test accuracy of baseline model

Based on the model's performance, it was seen that the BERT model had a higher accuracy and lower loss values than the baseline model. Please see the average values below:

	Train Accuracy	Test Accuracy	Train Loss
Baseline Model	90%	89.99%	0.069
Bert Model	98.24%	97.85%	0.014

8. QUALITATIVE RESULTS


comment_text

"Information (cont.)

Well?!? You do you want the info or not, you nasty little shit! 217.41.232.75

Could you stop the relentless persecution of me please?

Bugs,

Can you have a quiet word with those two please and get them to stop claiming I'm light currant? 217.41.226.121"

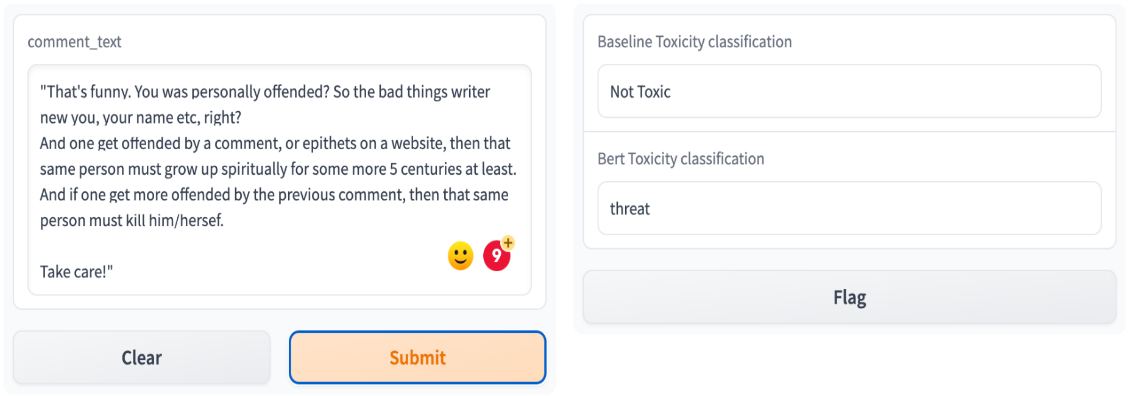
Baseline Toxicity classification

toxic

Bert Toxicity classification

obscene

The above is an example where the sentence was classified correctly using the baseline and the BERT model.


comment_text

"That's funny. You was personally offended? So the bad things writer new you, your name etc, right?
And one get offended by a comment, or epithets on a website, then that same person must grow up spiritually for some more 5 centuries at least.
And if one get more offended by the previous comment, then that same person must kill him/hersef.

Take care!"

Baseline Toxicity classification

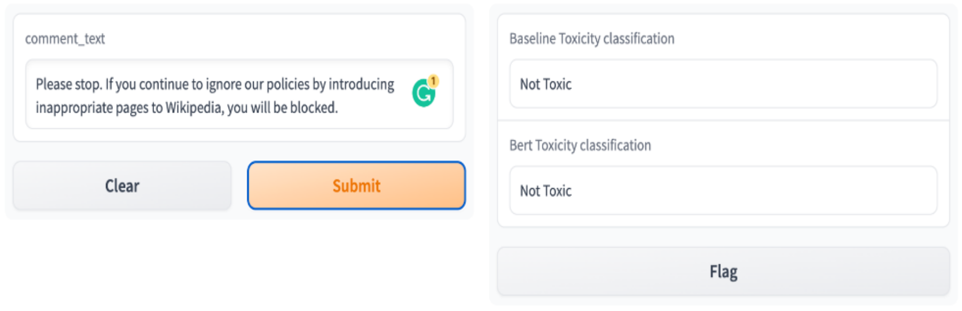
Not Toxic

Bert Toxicity classification

threat

The result below depicts a wrong baseline classification where a threat sentence was wrongly classified as non-toxic using the baseline model while the BERT model correctly classified the

sentence to threat. We consider this as an example where the BERT model's performance and efficiency is greater than the baseline model.
Both the models classified almost every sentence correct when tried to test. There are some sentences which involve wrong classification as shown below. We inferred that this is due to the presence of polite words like please in the sentence, and also due to comparatively less training dataset for the model.


comment_text

Please stop. If you continue to ignore our policies by introducing inappropriate pages to Wikipedia, you will be blocked.

Baseline Toxicity classification

Not Toxic

Bert Toxicity classification

Not Toxic

9. DISCUSSION AND LEARNINGS

The Bert model performs better than the baseline model both in terms of accuracy and loss. Balancing a multilabel dataset was a bit harder as improper split of train dataset led to misclassification by the model. As Bert model takes a lot of time(around 5 hours) to complete one epoch it was challenging to run the model for more epochs in our system. As we ran both the models we inferred that using transformers produce better train and test accuracies than normal machine learning models using glove embeddings. This can be because of the number of transformer blocks(12 in number) and the larger number of hyperparameters than the baseline model we used. Also, we could observe that even though the model performed well, for some sentences, the model classified it to be toxic even though those being other labels (like obscene or insult). We inferred from this behavior that the number of toxic sentences in the dataset was much higher than any other labels and this could be one of the reasons for this behavior. For future work, more data belonging to other labels should be added to the dataset to avoid this and increase model's efficiency.

10. INDIVIDUAL CONTRIBUTIONS

We are a team of 2 people. We divide the task among ourselves based on workload and schedule. We believe we make a good team and equally contribute to the project. We both discussed and got inputs from each other before starting on any concept or before making changes. We kept each other up to date on the progress. We planned on the timeline for each task and made sure to follow the timeline.

• Athira Indrakumar -Base code for Baseline mode, Code for Gradio, Report check and correction, code check and changes.
• Vishnu Priya Rajendran - Data collection and labeling, Base code for Bert model, Code check and changes, Report creation.

11. REFERENCES

[1] Hong Fan, Wu Du, Abdelghani Dahou, Ahmed A. Ewees, Dalia Yousri, Mohamed Abd Elaziz, Ammar H. Elsheikh, Laith Abualigah, Mohammed A. A. Al-qaness. Social Media Toxicity Classification Using Deep Learning: Real-World Application UK Brexit.
[2] Kevin Khieu, Neha Narwal. Detecting and CLassifying Toxic Comments.
[3] Spiros V. Georgakopoulos, Sotiris K. Tasoulis, Aristidis G. Vrahatis, Vassilis P. Plagianakos, Convolutional Neural Networks for Toxic Comment Classification, Department of Computer Science and Biomedical Informatics, University of Thessaly.