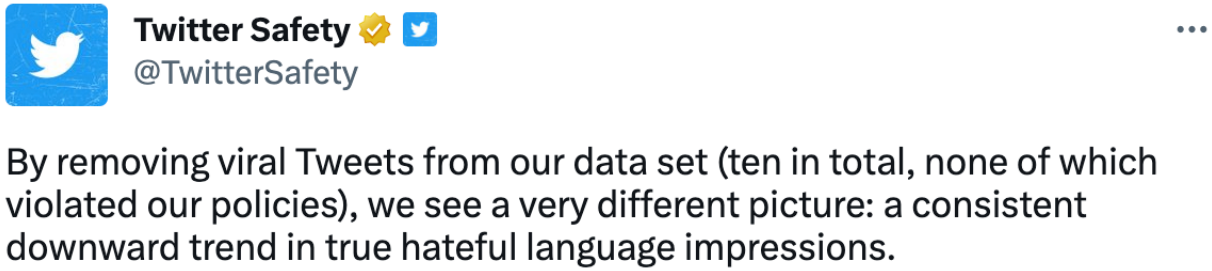
CS6301: Special Topics in Computer Science - Deep Learning for NLP

Contextual Toxicity Detection using BERT

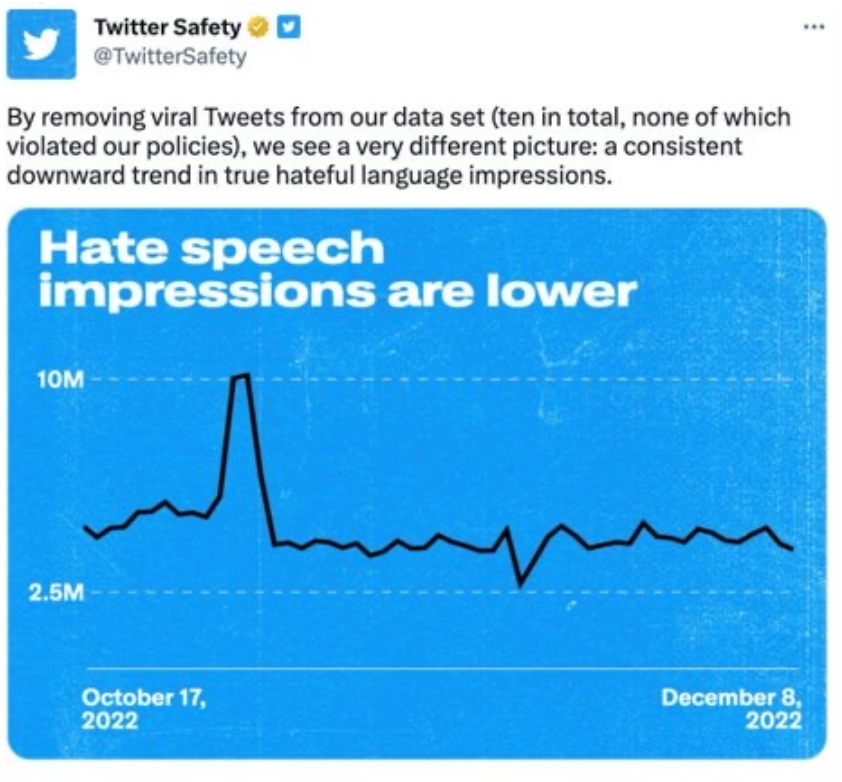


Ekaterina Lepekhina Sanatan Shrivastava Vishwesh Dave

Introduction

The ability to distinguish hate speech from offensive language is a major obstacle for automatic hate speech detection on social media.

- It is necessary to take into account the significant qualitative variations between several categories of potentially offensive language.
- For instance, a tweet that uses potentially offensive racial or gender slurs should not be treated the same as one that quotes rap songs that utilize such language.
- This distinction is frequently not made in existing work, and various forms of abusive language are frequently muddled.



Research Problem?

Through this project, we have tried working on the problem of hate speech and it's detection by analysis of the Pre-Trained BERT model with a neural-network mounted on the top of it.

Previous work done in the field:

Authors	Neural Network Approach	Year
Djuric et al.	Two-step strategy: Continuous bag of words model and binary classifier	2015
Waseem et al.	Multi-task learning framework to handle diverse datasets	2016
Gambck et al.	CNN model trained on various embeddings (word, character n-grams)	2018
Various (2018)	• Universal Language Model Fine-Tuning (ULMFiT), • Embedding from Language Models (ELMO), • OpenAI's Generative Pre-trained Transformer (GPT), • Google's BERT model	2018

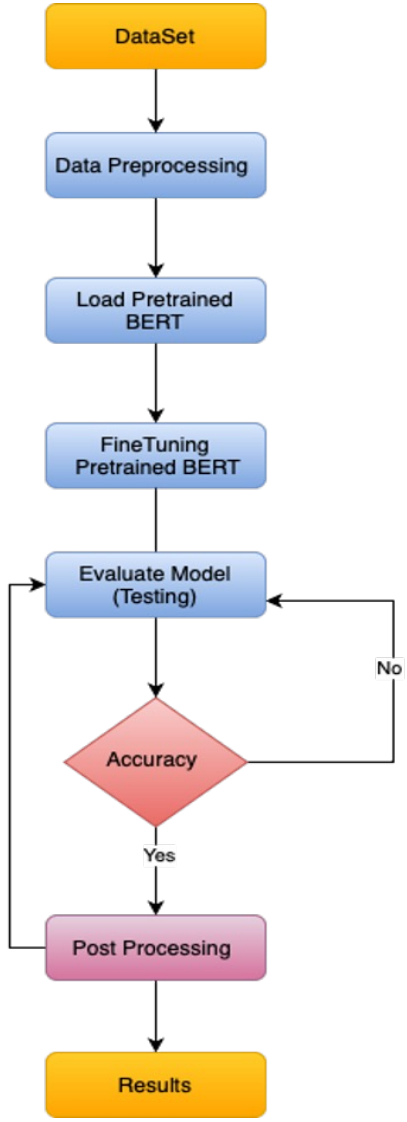
For many NLP activities, 2018 was a turning moment.

- With the aid of significant technological advancements like Google's BERT model, OpenAI's Generative Pre-trained Transformer, Embedding from Language Models (ELMO), Universal Language Model Fine-Tuning (ULMFiT) [9], and Embedding from Language Models (GPT) [18].

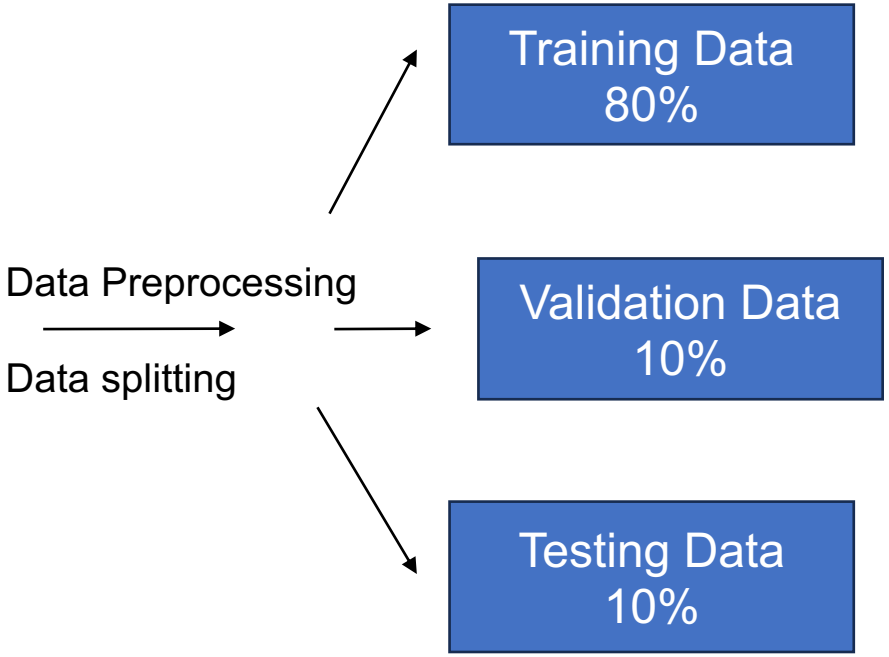
- Out of such models, we have analyzed Google's BERT model (Devlin et al.) for our project given that the pre-trained layers of the model can help us save a lot of computation and cost in training the model on a large text-corpora.

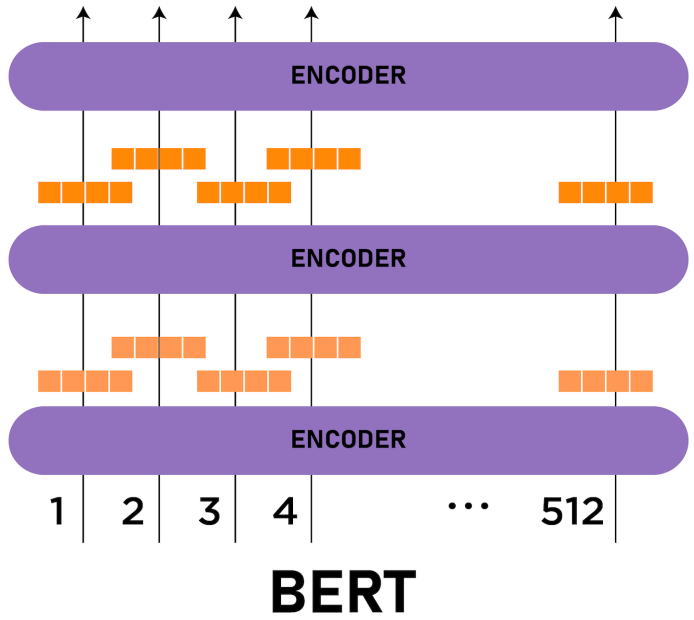
Methodology:

- To perform categorization of tweets, the methodology utilized in this study included fine-tuning a pre-trained BERT model on the "Hate Speech and Offensive Language" dataset.
- The dataset is first cleaned, then preprocessed by utilizing BERT's tokenizer and embedding layer to tokenize the text and map the tokens to the associated word embeddings.
- BERT-Base -Uncased Model is used for Pretraining.
- Different Classification models are then used to finetune the pretrained BERT model in order to reduce the discrepancy between predicted and actual labels.
- During training, data augmentation methods like token shuffle are used to further enhance the model's performance.
- Finally, validation and test set is used to assess the model's performance using metrics like accuracy, precision, recall, and F1 score.



Experimentations:





- We removed unnecessary characters, numbers, punctuation marks, and URLs from the tweets in the Davidson Hate Speech and Offensive Language dataset. We also converted all text to lowercase to ensure consistency in the data.

- We dropped the various irrelevant columns to our model like retweet count and hate, offensive and neither count as they do not contribute in classifying hate language.

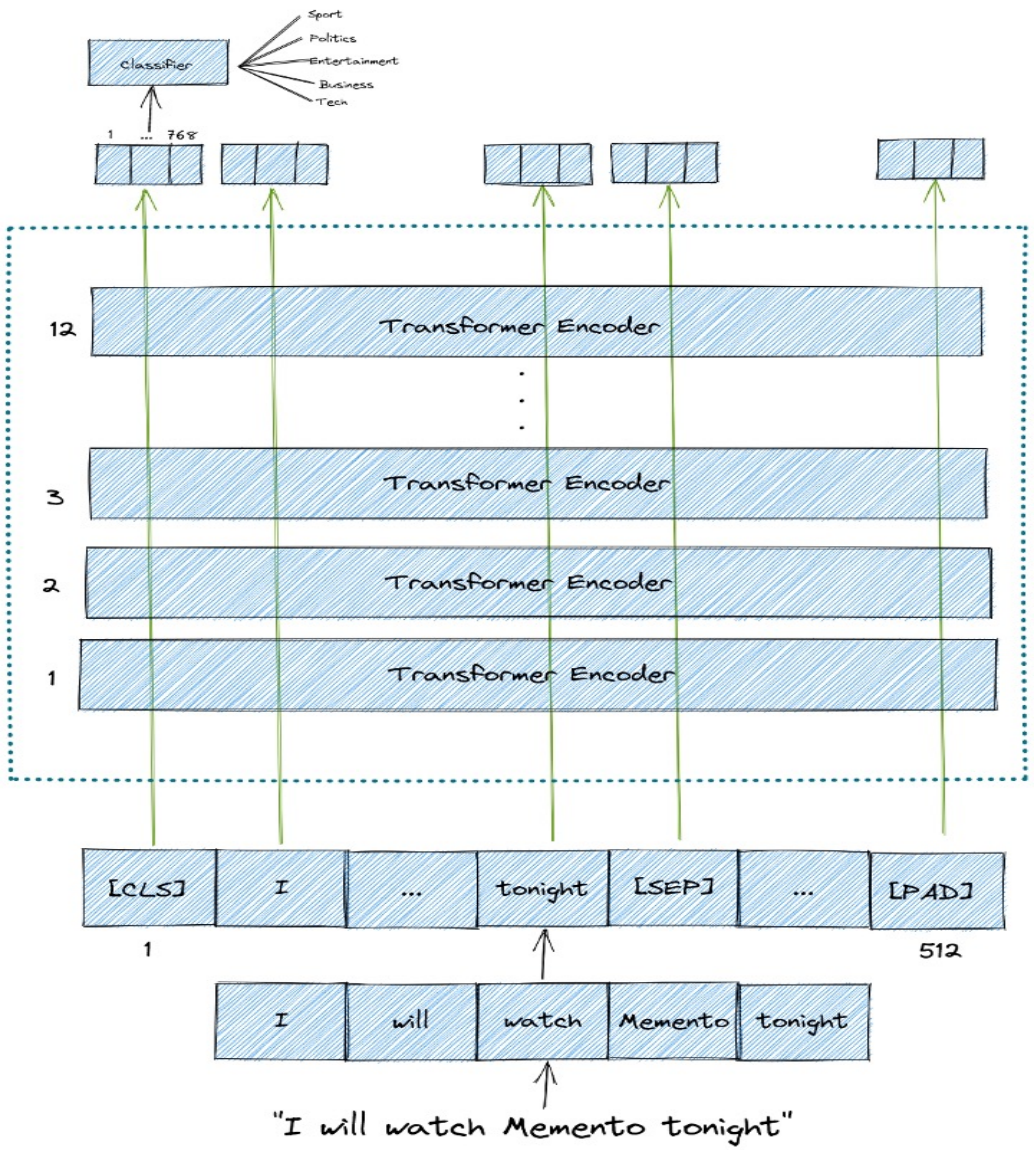
Experimentations:

- We split the dataset into training, validation, and testing sets using a 80:10:10 split ratio, respectively.
- We fine-tuned the 'bert-base-uncased' pre-trained BERT model on our dataset.
- We experimented with enhancing BERT-based model implementation with an additional neural network layer on top to improve performance.
- We experimented by taking different layers of BERT model into consideration from simple CLS token output to 11 layers to 13 (12 + CLS) layers.
- We experimented with the size of neural network from simple to complex.
- Also, we tried and tested our results on various batch sizes and epochs.

Implementation:

We have used various Classification models for Fine-tuning on top of BERT based pretrained model.

1. TF Auto Model For Sequence Classification
2. Simple Artificial Neural Network (ANN)
3. Simple Convolutional Neural Network



Implementation:

- The most accurate model among all the models implemented was the Artificial Neural Network (ANN).

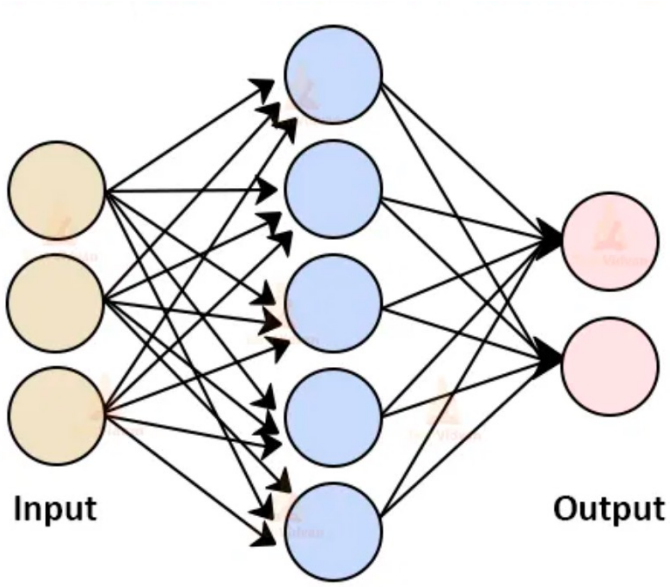
- The ANN model takes the hidden state output of all 13 layers of the pre-trained BERT model as input and implements a 2-layer deep neural network architecture with 512 neurons, ReLU activation function, dropout layer, and softmax activation function for classification.

- Binary cross-entropy loss function and Adam optimizer were used for training the ANN model.

- Early stopping was implemented to prevent overfitting.

- The TF Auto Model for Sequence Classification considers only the CLS token of BERT for classification.

- The CNN model takes the summation of the output of all layers of BERT and has 1 CNN layer on top of it.



Fine - Tuning

Dataset - Hate Speech and Offensive Language

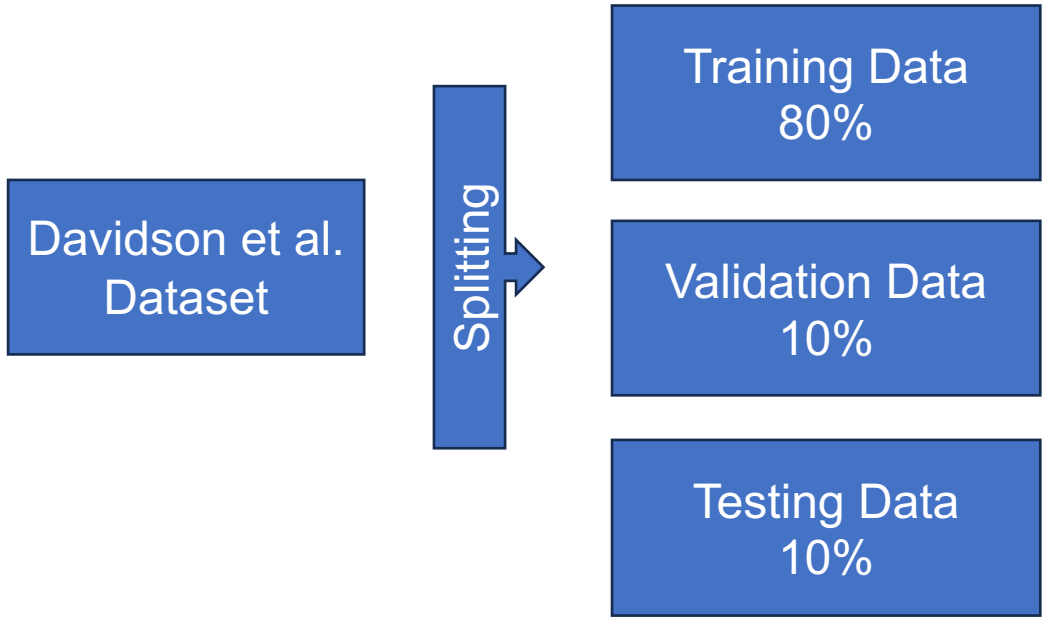
- "Hate Speech and Offensive Language" - **24,783** tweets labeled for toxicity detection.
  - Available at Automated Hate Speech Detection and the Problem of Offensive Language.

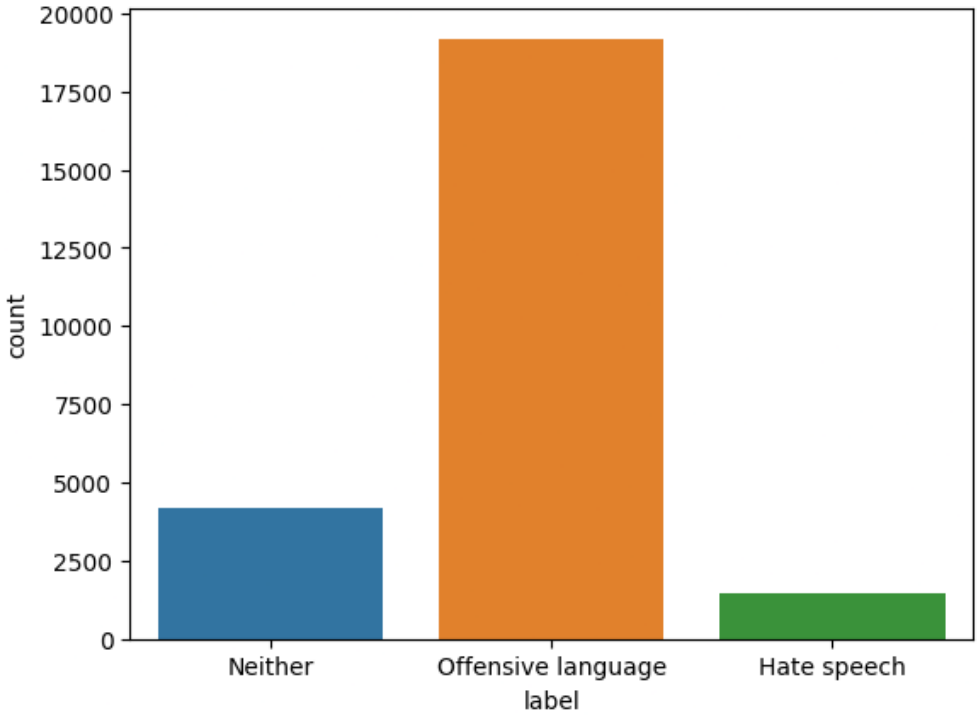
	Unnamed: 0	count	hate_speech	offensive_language	neither	label	cleaned
0	0	3	0	0	3	2	!!! RT : As a woman you shouldn't complain abo...
1	1	3	0	3	0	1	!!!!! RT : boy dats cold...tyga dwn bad for cu...
2	2	3	0	3	0	1	!!!!!!! RT Dawg!!!! RT : You ever fuck a bitch...
3	3	3	0	2	1	1	!!!!!!!!! RT _G_Anderson: _based she look like...
4	4	6	0	6	0	1	!!!!!!!!!!!! RT : The shit you hear about me ...

The dataset includes the following attributes:

- "tweet_text": The text of the tweet.
- "count": The number of times the tweet has been retweeted.
- "hate_speech": A binary indicator for whether the tweet contains hate speech.
- "offensive_language": A binary indicator for whether the tweet contains offensive language.
- "neither": A binary indicator for whether the tweet contains neither hate speech nor offensive language.

The annotations for the "hate_speech," "offensive_language," and "neither" features are not mutually exclusive, meaning that a tweet can be labeled as containing both hate speech and offensive language, or as containing neither, in addition to other combinations of labels.





- Hate speech is less common than ordinary offensive speech since it often calls for the use of specific terms that are directed at certain subjects.
- Weight balancing was used to address the problem of class imbalance.
- After the split, the number of training sets was 19826 (80%), validation sets were 2478 (10%), and test sets were 2479 (10%). The vocabulary size of this dataset is 48,312.

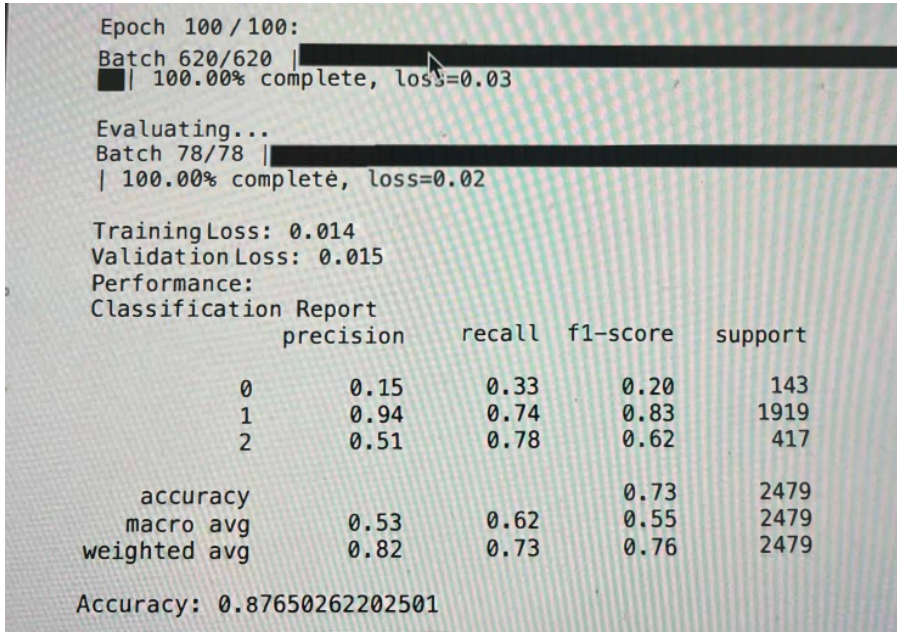
Results

- The overall Sparse Categorical Accuracy achieved for ANN model : 87%

- Offensive language class achieved the highest F1 score after 100 epochs, while 'hate speech' class only achieved 0.2 F1 score.

- The number of samples in "neither" class vs "hate speech" class is less drastic than the number in each of them vs "offensive language" class.

- F1 score for "hate speech" vs "neither" is noticeable for 100 epochs (similar difference for other epochs).



- It's easier for the model to differentiate between "no offensive language and/or hate speech" and "some presence of them" than between "offensive language" and "hate speech".

- More speed and resource utilization due to extra computations on combining ANN w/ BERT.
  - 20 epochs - 1.5hrs
  - 40 epochs – 2.5 hours using free-tier cloud resources
  - 100 epochs - 6 hours

Thank You!

Questions?