

Ryan Dimaranan (rtd180003)

Hannah Valena (hcv180000)

Text Classification Report

- Sequential Model
 - Using a Sequential model in Keras, we got a 50% accuracy on our dataset. This may be due to our dataset being a multi-class text classification problem because we are predicting tweets based on 3 classes, positive, neutral and negative. We also only ran 10 epochs during training due to time, but if we increased the number of epochs we might have received a higher accuracy.
- RNN & CNN
 - We tried to use a RNN and CNN layer to predict our dataset, but with our large dataset size and the limited RAM provided by Google Colab, we were unable to find a configuration that allowed us to train the model without it crashing or taking days to complete.
- Embeddings
 - As a baseline, we use the layers found in the example notebook for chapter 25
 - Several layers of Conv1D followed by pooling, ending in a softmax classification layer.
 - Training Embedding Layer
 - When training an embedding layer instead of using a pre-trained embedding, I got an accuracy of 88% with 15 epochs.
 - GloVe
 - When using the GloVe pre-trained embedding layer, we got an accuracy of 91% with 15 epochs.
 - Through this experiment, we found that using the GloVe pre-trained embedding gave better results than training my own embedding layer. In addition, using pre-trained embedding also leads to a faster training time since the weights are already determined. In our opinion, for most NLP tasks that require an embedding layer, using pre-trained embeddings would be the best option due to their performance and time efficiency.