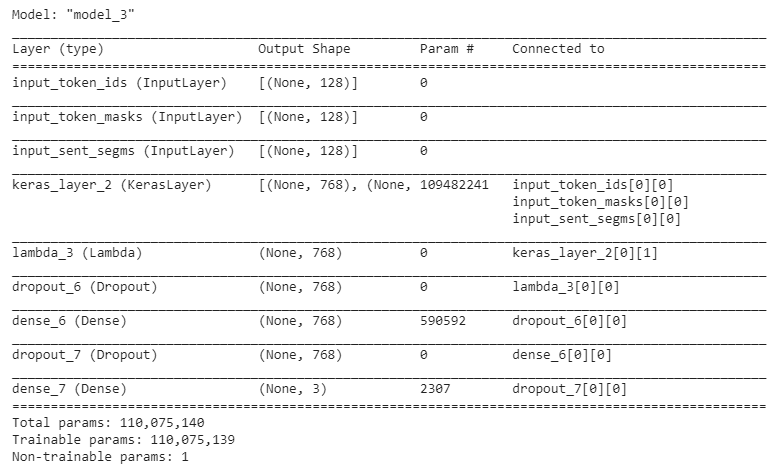
# Model Report

Michael Dowd, Napier Number: [40451589](mailto:%34%30%34%35%31%35%38%39@%6c%69ve%2e%6ea%70%69e%72.%61%63%2euk)

BERT is a neural network pre-trained on the Wikipedia and BooksCorpus data on the tasks of predicting missing words and sentence-matching. Pre-trained BERT models can be used simply as feature generators with excellent results. However the authors of the BERT paper achieved the best results on various NLP tasks after fine-tuning the model over a small number of epochs, this is the approach I took for this exercise.

First, the pre-trained weights were downloaded as a keras layer from tensorflow hub using this link: <https://tfhub.dev/tensorflow/bert_en_uncased_L-12_H-768_A-12/1>. I then appended additional layers to the model to augment it for the abusive language classification task. I tried a number of configurations for the additional layers, however ultimately I achieved the best results using a configuration drawn from an example on the bert-for-tf2 github repo: <https://github.com/kpe/bert-for-tf2/blob/master/examples/gpu_movie_reviews.ipynb>. The model architecture is shown below, note the two dropout layers to avoid over-fitting:



The BERT paper recommends some training parameters which the authors used to achieve good results during fine-tuning. Following these guidelines, I trained the model with an Adam optimiser and a small learning rate of of 2e-5 to minimise *Categorical Cross-Entropy*. I also used a batch size of 32. Finally the paper recommends training for just 2, 3 or 4 epochs, I implemented early stopping on the minimum value the validation loss, which occurred at epoch 5.

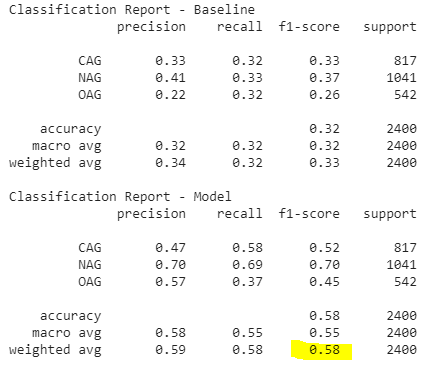
During training, 25% of the *training* data was held out for validation and early-stopping. From the charts and output below you can see that the validation precision and recall essentially level out after a number of epoch. The validation loss function, categorical cross-entropy, decreases to a minimum at epoch 5 and increases thereafter while training loss continues to decrease, suggesting that after epoch 5 the model is overfitting to the training data.

|  |  |  |
| --- | --- | --- |
|  |  |  |

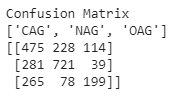
|  |  |
| --- | --- |
|  |  |

For evaluation, a test set of 20% (or 2400 samples) of the total cleaned data was held apart from the training and validation set. First, baseline predictions were generated by randomly predicting labels for these test samples. As expected, given 3 labels, a random predictor achieves a weighted average precision, recall and f1-score of 33%.

*The trained model significantly outperforms the baseline and achieves a weighted average f1-score of 58%.* Interestingly the model achieves the best results on the Non-Aggressive samples (NAG, f1-score: 70%), second best results on the Covertly Aggressive samples (CAG, f1-score: 52%) and the worst results on the Overtly Aggressive samples (OAG, f1-score of 45%).



The confusion matrix provides more detail:



* Of the 817 CAG samples, 228 were incorrectly predicted as being NAG while 114 were incorrectly predicted as being OAG
* Of the 1041 NAG samples, only 39 of them were incorrectly predicted as being OAG, while 281 were incorrectly predicted as being CAG.
* Of the 542 OAG samples only 78 of them were incorrectly predicted as being NAG, while 265 of them were incorrectly predicted as being CAG.

The model clearly has some difficulty differentiating overt aggression from covert aggression. Also, for non-aggressive comments the model is much more likely to mis-label them as covertly aggressive than overtly aggressive. This is inline with our understanding of the labels – covert aggression is a subtle concept even for a human to understand in some cases.