Youtube Sentimental Analysis - Preliminary Results

1. Problem Statement

My project consists of creating a website which allows users to enter a link to a YouTube video to predict the overall sentiment toward the video by looking at its comments. The sentiments would be classified into five classes: very positive, positive, neutral, negative, and very negative.

2. Data Preprocessing

Although I am analyzing Youtube comments, I decided to change my dataset to an IMDB movie reviews dataset containing five classes as labels (very negative, negative, neutral, positive, and very positive). I changed the dataset since my previously selected dataset did not contain labels related to the comments. My current dataset includes three columns: text (the review), label, and label_text. It comes with three json files, one for training, one for validating, and one for testing. The training dataset has 8544 reviews, the validation dataset contains 1101 reviews, and the testing dataset contains 2210 reviews. I did not change any data in the dataset as everything was already in order, all columns were relevant in my case, and there were no missing data. Since the training dataset needed better distribution, I oversampled it using imlearn's RandomOverSampler. I used Google's implementation of BERT English uncased preprocessor to preprocess the data as I will use the BERT model for my project. To preprocess the data, I used the BERT preprocessing module from TensorFlow. It tokenizes the reviews and assigns them input_ids, attention_masks, and input_type_ids.

3. Machine Learning Model

The model that I will use is the BERT model. I opted for the BERT model as it is pre-trained and could better interpret Youtube comments, as BERT can analyze different words and sentences using context.

a. I decided to use Keras to build a train for my model. I have been testing my model with various layers. However, currently, my model contains three input layers, one intermediate layer, and one output layer. The three input layers consist of the input_ids, input_masks, and input_type_ids. I added these three layers since they are required for the bert_layer. The bert_layer is needed to create the embeddings of the inputs. The embeddings are the vectors that capture the meaning of words from the context of a sentence. Following that, I pass the embeddings to a dense layer. The final dense layers contain five neurons since I have five classes, and I use the softmax activation function to predict the probability distributions of the sentiment labels.

b. I did have to do that step to separate the validation, training, and test sets, as my dataset already came with these three sets separated. To optimize the model, I am using Adam's optimizer, the same one used to train the pre-trained Bert model. I am currently still adjusting my hyperparameters. However, I adjust them to improve the model's accuracy while reducing the training time. I am changing the hyperparameters, the batch size, and the epochs needed to train the model.

c. To test the model, I plan on using the accuracy metric and running my model on the test set. However, currently, my categorical and validation accuracy hovers around 40%. Therefore, the model can either be underfitting or there might be unoptimized hyperparameters causing the low accuracy.

d. My challenges were understanding the basic concepts of transformers, BERT, and how TensorFlow worked. I solved that issue by reading online and watching some tutorials. As for technical challenges, I am currently having problems fitting my model as it has a very low accuracy in each epoch. Therefore, currently, I am trying to solve that by adjusting hyperparameters.

4. Preliminary Results

At this stage, I do not have graphs to evaluate my model as I have been trying to fit my model and have high accuracy when fitting the model. During the epochs I have run, the classification accuracy and validation accuracy hovered around 40%, which seems low. Therefore, my model needs to be more accurate and perform better.

5. Next steps

My next steps would be to optimize the performance of my model and potentially add more data to the train model and observe if it can increase the accuracy of my model. I will also adjust the hyperparameters to improve the accuracy as around 40% is a low accuracy.