**Assignment Report**

Name: Prajwala Mugajji Shambulingappa

**Aim**: Goal in this homework is to find several pretrained classification models on ‘huggingface’ which all focus on a similar task (e.g., sentiment classification IMDB, sarcasm detection, AG News classification) and compare them. Also, choose models which are all trained in the same classification task.

**Observation Criteria and Analysis**:

1. Packages Used: torch, torchvision, NumPy, TensorFlow, datasets - load\_dataset, transformers - BertTokenizer, BertForSequenceClassification, RobertaTokenizer, RobertaForSequenceClassification, sklearn.metrics - f1\_score, precision\_score, recall\_score.
2. Dataset Used: ‘rotten\_tomatoes’ dataset from Huggingface. The dataset contains 5,331 positive and 5,331 negative processed sentences from Rotten Tomatoes movie reviews. The dataset includes a ‘text’: a string feature and a ‘label’: that has the classification label classifying if the data is a positive review or a negative review.
3. Task: sentiment classification of the rotten-tomatoes dataset.
4. For this task I have chosen to work with “BERT”. I chose to work with BERT because firstly, it is a bidirectional model, meaning that it can analyze the context of a word by looking both forward and backward in a sentence. Secondly, BERT is pre-trained on large amounts of data, which helps it to better understand the structure and meaning of language. Also, it uses a self-attention mechanism that enables it to focus on the most relevant parts of a sentence, helping it to capture important features for classification and improve accuracy.
5. Pretrained BERT model Used:
   1. ‘mrm8488/distilroberta-finetuned-rotten\_tomatoes-sentiment-analysis’: This model is a fine-tuned version of distilroberta-base trained on the rotten\_tomatoes dataset.
   2. ‘RJZauner/distilbert\_rotten\_tomatoes\_sentiment\_classifier’: This model is a fine-tuned version of distilbert-base-uncased on the rotten\_tomatoes dataset.
   3. ‘Ghost1/bert-base-uncased-finetuned\_for\_sentiment\_analysis1-sst2’: This model is a fine-tuned version of bert-base-uncased on the glue dataset.
6. For tokenization, I used AutoTokenizer. This automatically chooses an appropriate tokenizer for the given model and configures it based on the settings of the model. For BERT, tokenizer is configured to handle text data that has been preprocessed using the same tokenization scheme as the original pre-training corpus.
   1. The first tokenizer used creates a tokenizer object for the "distilroberta-finetuned-rotten\_tomatoes-sentiment-analysis" model, which was pre-trained on the RoBERTa architecture and then fine-tuned on the Rotten Tomatoes dataset.
   2. The second tokenizer used creates a object for the "distilbert\_rotten\_tomatoes\_sentiment\_classifier" model, which was pre-trained on the DistilBERT architecture and fine-tuned on the same Rotten Tomatoes dataset.
   3. The third tokenizer used creates a tokenizer object for the "bert-base-uncased-finetuned\_for\_sentiment\_analysis1-sst2" model, that can be used for classification. This model was fine-tuned on the Stanford Sentiment Treebank (SST-2) dataset.
7. Testing the pretrained model for sentiment analysis: As a part of testing, I defined a function called "predict\_sentiment" that takes in the pre-trained model, tokenizer, and text input defined above, and returns a predicted sentiment label for the input. Text that was passed as input is tokenized and fed through the model to generate a prediction based on the model's output probabilities. For each model, I iterated through the test dataset, extracting true labels for each text sample, and predicting a sentiment label. After which I compared the predicted labels to the true labels and calculated accuracy, F1 score, precision, and recall.
8. However, initially I trained the model using Bert-base-uncased, Bert-base-case and Roberta-base which gave me very weird accuracy and other evaluation metrics as it was only creating embeddings and not classifying. Hence, I switched to other models and after experimenting with some other models such as ‘roberta-base-rotten\_tomato’ model, ‘XLNet’ models and others. After many trials with multiple models the selected models performed comparatively better.

**Summary**:

1. Code that uses several pretrained classification models on ‘huggingface’ was implemented successfully.
2. For comparison, I used 3 models to classify the ‘rotten\_tomato’ namely, mrm8488/distilroberta-finetuned-rotten\_tomatoes-sentiment-analysis, RJZauner/distilbert\_rotten\_tomatoes\_sentiment\_classifier and Ghost1/bert-base-uncased-finetuned\_for\_sentiment\_analysis1-sst2.
3. Comparing the model performance for ‘rotten\_tomato’ dataset:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tokenizer | Model | Accuracy | F1 Score | Precision | Recall |
| distilroberta | distilroberta | 86.40% | 0.8639 | 0.864 | 0.8640 |
| distilbert | distilbert | 83.80% | 0.8379 | 0.838 | 0.8380 |
| bert-base-uncased | bert-base-uncased | 82.10% | 0.8209 | 0.821 | 0.8210 |

1. From the given table we can conclude that the pretrained model distilroberta is performing the best amongst all the models we compared with an accuracy of 86% and an F1 score of 0.8639. However, distilbert and bert-base-uncased are behind and have a good accuracy of 84% and 82% with an F1 score of 0.83 and 0.82 respectively.
2. The performance of these models can be improved by fine-tuning the model on the same dataset we used for testing them.