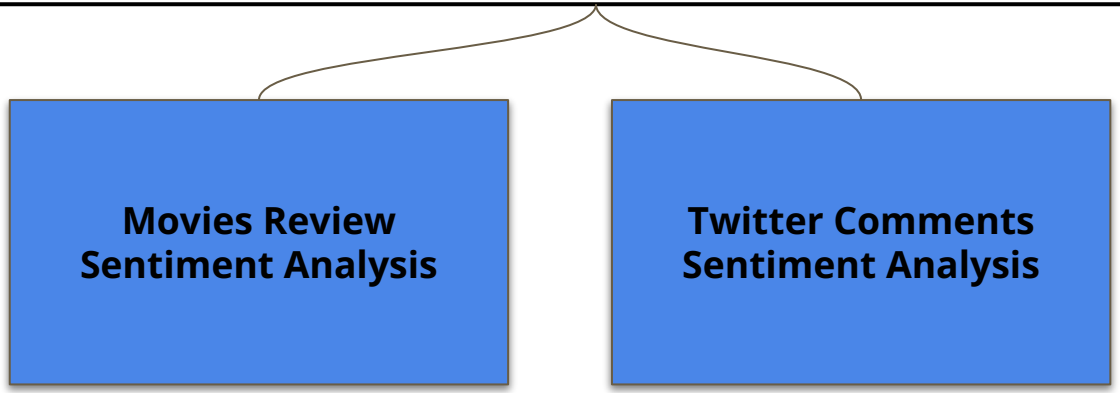
Sentiment Analysis on Movie Reviews and Twitter Comments

Gökay Gülsoy 270201072
Merve Nur Ozan 270201071

Introduction

- Our task is a very well known problem in NLP known as sentiment analysis.
- We are applying sentiment analysis problem to two different cases



Our task is actually binary classification problem (being positive or being negative)

Datasets

- Movie reviews Imdb dataset consists of 25,000 training and 25,000 test review data
- Twitter comments dataset consists of 1,600000 comment
- Datasets consists of mainly two classes positive or negative
- As Twitter dataset is large we have used GPU for training in Colab environment

Methodology

- We have trained an relatively older model neural network, LSTM.
- We also trained an BERT model which is one of the newer models that provides better accuracy and smaller loss values.

- Our LSTM model uses encoder as the first layer for encoding the input words to indices, then embedding layer takes the output of the encoding layer to create word embeddings. Embeddings are then given to LSTM unit. Output of LSTM unit is given to Dense layer with size 1 which outputs the classification result.

Methodology

- In our BERT model we have used Input layer as the first layer which takes thetext in appropriate shape and then text is given to preprocess layer in order to preprocess text. Outputs of preprocessing layer are then given to BERT encoder, then dropout layer with dropout ratio 0.1 is added in order to provide regularization.

Metrics Chosen

- Binary accuracy
- AUC
- F1 score
- Mean squared error

- Most prominent difficulty that we have faced was the training time for the BERT models, We have used A100 GPU to train the models for making training considerably faster.

Experiments and Results

LSTM Model Experiment Results

Experiments	loss	Binary accuracy	F1 score	Validation loss	Validation binary accuracy	Validation f1 score
Stacking LSTM with 32 layer	0.4458	0.8045	0.8078	0.4558	0.8318	0.8319
Adding Dense layer with 128 units	0.4546	0.8045	0.8078	0.4558	0.8047	0.8084

Experiments	loss	Binary accuracy	F1 score	Validation loss	Validation binary accuracy	Validation f1 score
Changing activation function from relu to tanh	0.4778,	0.7849	0.7759	0.4812	0.8120	0.8194
Adding Dropout layer	0.6387	0.6988	0.6562	0.5454	0.7026	0.6170
Increasing hidden layer size to 128	0.5141	0.7499	0.7292	0.4774	0.7974	0.8094
Decreasing batch size to 32	0.4374	0.8112	0.8082	0.4823	0.7969,	0.8083

BERT Model Experiment Results

Experiments	loss	Binary accuracy	F1 score	Validation loss	Validation binary accuracy	Validation f1 score
Increasing dense layer number by adding dense layer with 64	0.1519	0.9437	0.9428	0.4844	0.8552	0.8548
Increasing Dropout ratio from 0.1 to 0.5	0.1752	0.9336	0.9323	0.4848	0.8510	0.8513

Changing learning rate from 3e-5 to 2e-5	0.2116	0.9128	0.0.9104	0.4123	.0.8476	0.8467
Changing batch size from 32 to 64	0.2491	0.8921	0.8880	0.3773	0.8454,	0.8416

Discussion

- BERT models have better accuracy and validation score compared to LSTM based models.
- Experiments shown that changing different parameters that affect the performance of the model may not have uniform effect in most cases which means that there may not be straightforward relationship between metrics like accuracy, loss, or validation loss.

Conclusion

- Overall results have supported that state-of-the-art models like BERT overperformed older neural models such as LSTM in our case.
- We have learned essential deep learning workflow used in NLP which is data preprocessing, building model, training model, evaluating model, prediction on new data respectively.

- We could have conducted an experiment which uses larger BERT model we have currently used smaller BERT model.
- We could have conducted an experiment which trained for 10 epochs (default is 5 epochs in our model)