Deep Learning REPORT

BS(CS) Spring 2024

Name: Abdullah
Memon 20K-0283

Name: Abdul
Ahad Shaikh
20K-0319

Teacher: Miss Sumaiyah Zahid

Department of Computer Science

**FAST-National University of Computer &
Emerging Sciences, Karachi**

Sentiment Analysis Using Transformers

Objective:

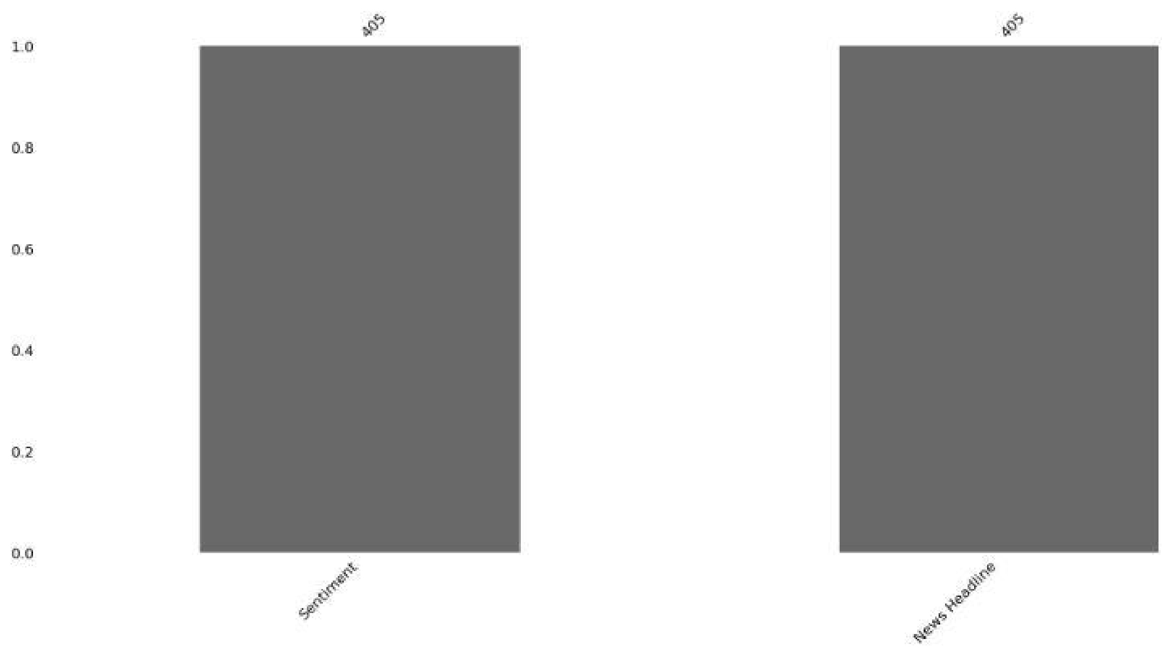
The objective of this project is to implement sentiment analysis using the BERT (Bidirectional Encoder Representations from Transformers) model to accurately classify textual data into positive, neutral, or negative sentiment categories. The project aims to develop a robust deep learning framework capable of analyzing the sentiment of textual input across various domains with a focus on assigning sentiment scores ranging from 1 to 5 for both positive and negative sentiments. Through this endeavor, the project seeks to enhance understanding and interpretation of sentiment within textual data, facilitating applications in sentiment analysis for diverse real-world scenarios.

Problem Statement:

Sentiment analysis is the process of understanding and categorizing textual data into positive, neutral, or negative sentiments and it plays a pivotal role in various applications across industries. However, existing sentiment analysis models often struggle with nuanced understanding and accurate classification. To address this challenge, our project aims to implement sentiment analysis using the BERT (Bidirectional Encoder Representations from Transformers) model. The objective is to develop a robust deep learning framework capable of accurately classifying textual data into sentiment categories, with a focus on assigning sentiment scores ranging from 1 to 5 for both positive and negative sentiments.

DATA USED

The dataset comprised both sentiments and news headlines, such as current events and public perceptions. The headlines encompassed a wide array of topics, ranging from significant global developments to localized news items. To ensure consistency, special symbols were removed from the headlines during preprocessing. Subsequently, sentiment analysis was integrated to discern the emotional nuances associated with each headline, categorizing them as positive, negative, or neutral.



Methodology:

- Importing Libraries:
  ktrain, scikit-learn, TensorFlow, spaCy, Pandas, NumPy
- Data Loading and Preprocessing:
  The dataset is loaded from a CSV file, initial data exploration is done, missing values and duplicates are removed, and text preprocessing is performed using spaCy.

- Text Tokenization and Encoding:
  The text data is tokenized and encoded using the BERT tokenizer by ktrain, and then divided into training and testing sets.
- Building and Training the Model:
  A BERT-based text classifier model is built using ktrain's text_classifier function.The model is trained using the training data, with hyperparameters.
- Making Predictions:
  The trained model predicts sentiment classes in sample text and a dataset, maps predicted labels to human-readable categories, and generates ratings based on predictions' confidence.

Results:

BERT

```
begin training using onecycle policy with max lr of 2e-05...
Epoch 1/3
58/58 [==============================] - 532s 9s/step - loss: 1.0973 - accuracy: 0.4409
Epoch 2/3
58/58 [==============================] - 521s 9s/step - loss: 0.8863 - accuracy: 0.6052
Epoch 3/3
58/58 [==============================] - 491s 8s/step - loss: 0.4837 - accuracy: 0.8184
<keras.src.callbacks.History at 0x7882d0163280>
```

Simple Sentiment Prediction with Custom Labels

```
import ktrain
from ktrain import text

label_map = {'Sentiment_0': 'negative', 'Sentiment_1': 'neutral', 'Sentiment_2': 'positive'}

sample_text = "I love this product! It's amazing."

predictor = ktrain.get_predictor(learner.model, preproc)


numeric_prediction = predictor.predict(sample_text)
print(numeric_prediction)
sentiment_name = label_map[numeric_prediction]


print("Predicted sentiment:", sentiment_name)
```

```
Sentiment_2
Predicted sentiment: positive
```

Making Predictions with Probabilities and rating for first 10 samples from dataset

```
--------------------
Sample 7 - Text: 'yit say acquisition strategy expansion central eastern european market'
  Predicted sentiment: positive
  Probabilities [negative, neutral, positive]: [0.006264152936637402, 0.01958436705172062, 0.9741514325141907]
  Rating: positive - 5
====================
Sample 8 - Text: 'basware product sale grow strongly financial period percent'
  Predicted sentiment: positive
  Probabilities [negative, neutral, positive]: [0.01064243447035551, 0.010126035660505295, 0.9792314767837524]
  Rating: positive - 5
====================
Sample 9 - Text: 'second quarter firstquarter growth net sale quarter say magnus rosen ramirent ceo'
  Predicted sentiment: positive
  Probabilities [negative, neutral, positive]: [0.018228748813271523, 0.16941744089126587, 0.812353789806366]
  Rating: positive - 4
====================
Sample 10 - Text: 'transaction strengthen position design brand good say fiskars president ceo heikki allonen point group relatively overlap operation'
  Predicted sentiment: positive
  Probabilities [negative, neutral, positive]: [0.008959456346929073, 0.02638513594865799, 0.9646554589271545]
  Rating: positive - 5
====================
```

Making Prediction with probabilities and rating for sample text

```
Predicted sentiment: positive
Probabilities [negative, neutral, positive]: [0.10033182799816132, 0.2216554582118988, 0.6780127286911011]
Rating: positive 2
```

DistilBERT

```
begin training using onecycle policy with max lr of 2e-05...
Epoch 1/3
58/58 [==============================] - 869s 15s/step - loss: 1.0378 - accuracy: 0.5043
Epoch 2/3
58/58 [==============================] - 836s 14s/step - loss: 0.8276 - accuracy: 0.5764
Epoch 3/3
58/58 [==============================] - 835s 14s/step - loss: 0.4507 - accuracy: 0.879
<keras.src.callbacks.History at 0x7882c251fb50>
```

Conclusion

DistilBert Performs better as the Dataset size is small and distilbert uses lesser parameters compared to BERT hence is less prone to overfitting. However, both Models perform well on the dataset

REFERENCES:

https://huggingface.co/docs/transformers/en/model_doc/bert

https://huggingface.co/docs/transformers/en/model_doc/distilbert