LiveDataLab Leaderboard Competition
Team BABL

Team Information:

a. Raoul Larios (NetID:rlarios2) - Captain
b. Lydia Brothers (NetID:lydianb2)
c. Asritha Bobbala (NetID:asritha2)
d. Hammad Ali (NetID:hali53)

Project Overview:

Our goal with this project was to set up a leaderboard competition to prompt students to train a performant model on a dataset of tweets that are labeled as either a negative, neutral, or positive sentiment. This is a multi-classification problem on twitter text data where depending on the context of the text, the sentiment will be classified as one of the 3 options. The dataset used is the Twitter Tweets Sentiment classification dataset located here: https://www.kaggle.com/datasets/yasserh/twitter-tweets-sentiment-dataset. Students will compete to achieve the highest model performance.

Implementation Documentation:

All of the code for this project is located within the tweet_classifier_model.py file. The program trains a BERT-base-uncased model on the Twitter Tweets Sentiment classification dataset. It first loads the dataset from a CSV using pandas and splits the dataset into training and test datasets. Then the pre-trained BERT model and tokenizer are loaded. The dataset is then pre-processed through tokenization and the DataLoader is created for both the train and test sets. The training set uses the 'selected-text' column of the dataset which contains portions of the full tweet content('text' column) that capture the sentiment. The test set uses the 'text' column. Labels for both sets are converted to integers: negative as 0, positive as 1, and neutral as 2. Parameters are then defined for fine-tuning, such as the learning rate, number of epochs, and type of optimizer. The model is then trained and finally the performance is evaluated on the train and test sets. The final output looks like the following:

```
PS C:\Users\raoul\Documents\GitHub\BABL_CS410> python3 .\tweet_classifier_model.py
Some weights of BertForSequenceClassification were not initialized from the model checkpoint at
You should probably TRAIN this model on a down-stream task to be able to use it for predictions
C:\Users\raoul\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.9_qbz5n2kfra8p0\LocalCa
implementation of AdamW is deprecated and will be removed in a future version. Use the PyTorch
arning
  warnings.warn(
Fine-tuning BERT on tweets... Starting Epochs

C:\Users\raoul\Documents\GitHub\BABL_CS410\tweet_classifier_model.py:74: UserWarning: To copy
lone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).
  labels = torch.tensor(batch['label'])
Finished Fine-tuning. Elapsed Time: 48.13612804015477 minutes and 8.167682409286499 seconds

Evaluating BERT on tweets... Starting Predictions

C:\Users\raoul\Documents\GitHub\BABL_CS410\tweet_classifier_model.py:109: UserWarning: To copy
clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).
  labels = torch.tensor(batch['label'])
Finished Predictions. Elapsed Time: 3.3222720901171368 minutes and 19.3363254070282 seconds

Accuracy: 0.6445333818446425
Classification Report:
              precision    recall  f1-score   support

           0       0.80      0.39      0.52      1562
           1       0.88      0.56      0.68      1705
           2       0.54      0.89      0.67      2230

    accuracy                           0.64      5497
   macro avg       0.74      0.61      0.63      5497
weighted avg       0.72      0.64      0.63      5497

PS C:\Users\raoul\Documents\GitHub\BABL_CS410>
```

In our initial Project Plan we had intended on directly connecting this coding competition to LiveDataLab. Due to complications with staff and lack of setup instructions, we were unable to obtain LiveDataLab credentials and Microsoft Azure credits to conduct the competitions. Instead we developed a model and provided the sentiment of tweets we analyzed and trained our model on and illustrate that this code could be augmented by students in a competition on LiveDataLab similarly to past Programming Assignments in this class. We did receive feedback from TAs that it was fine to proceed with this as a project without direct LiveDataLab integration.

If we were to implement our competition on LiveDataLab, the leadership board would have a column for the participant's id and one column for their F1 score. The participant would have to submit their results of the test set as a csv file to LiveDataLab. The higher the accuracy score, the higher up they are on the board.

Instructions for students:

In this competition, students will develop a classifier on a multiclass dataset and participate in a classification competition. The classifier will be evaluated using accuracy on the Twitter Tweets Sentiment Classification Dataset.

The tweet_classifier_model.py file contains some starter code for a BERT model to be applied to a training dataset and generate predicted labels for the test set. This will serve as the baseline model performance that students will aim to improve upon.

Step 1: Clone this repository
Step 2: Train a competitor model or fine-tune the existing model on the provided training dataset

Students are free to use any competing classifier and fine-tune hyperparameters such as learning rate and the number of epochs, etc.

Code and performance metrics will be collected by the TA and then ranked by model performance using accuracy. If you provide a performance better than the baseline model provided, you will be given credit for the assignment.

Project Setup:

To run this project, first clone this repository. This should include the archive folder which contains the Tweets.csv dataset required for model training.
Then navigate to the tweet_classifier_model.py file. Ensure the following are installed

1. pip install transformers
2. pip install torch
3. pip install pandas

Finally, perform the python3 tweet_classifier_model.py command to run the file. Make edits to the hyperparameters and model form as needed to improve performance.

Name	**Contribution**
Raoul Larios (rlarios2)	Report, Code, Video
Lydia Brothers (lydianb2)	Report, Code, Debugging
Asritha Bobbala (asritha2)	Report, Code, Research
Hammad Ali (hali53)	Report, Code, Commenting code