Airline Tweet Sentiment Analysis: Use of Transformer Architecture and Captum Interpretability in Text Classification

Yaxin Zhuang Individual Report

I. Introduction

In our project, we are aiming to fine-tune BERT based model with MLP or LSTM head to perform text classification on twitter airline sentiment data, and apply several model interpretation methods to explain the model in the text classification task.

After evaluating the training results of training and validation data, we decided to focus on the BERT model with the MLP head. Because with the same number of epochs(20), LSTM head model is overfitting with higher training and validation accuracy, and higher model complexity than the optimum. Compared to LSTM head model, MLP head model achieved more reasonable training and validation accuracy score and loss, making MLP a better choice of model interpretation.

In this project, I am mainly working on using the LIME approach to interpret the model, and also working on data preprocessing.

II. Description of individual work

- Split contradiction words
- Experiment on model interpretation using LIME(Local Interpretable Model-Agnostic Explanations)
- Confusion matrix interpretation

In this project, I am mainly working on using the LIME approach to interpret the model and also helping with data preprocessing and confusion matrix interpretation for post hoc analysis. For the paper and presentation, my work is primarily related to data description, model description, experiment setup, and model result.

For the data processing, I worked on split contradiction words and removing stopwords. The contradiction word might affect the progress of tokenization, therefore, we decided to split them into their subwords, make they can be recognized by the tokenizer in other words. I also tried to remove stopwords from the text at the beginning and compared the training and validation accuracy and loss from the training with 5 epochs. There is no big difference between stopwords and without stopwords, since we decided to use the pre-trained model, it is better to keep all stopwords to provide enough context information like negation words(not, nor, never) which are also considered to be stopwords. Also, since Twitter uses informal language, it makes preprocessing harder, for example, how to process emojis and emoticons in special characters, numbers, hyperlinks, etc

For the model interpretation, we planned on using the LSTM head, because of the overfitting issue of LSTM head model, we decided to use MLP head model in the end. LIME is suited for a classifier that takes as input a list of strings and outputs a 2d array of prediction probabilities. Lime creates some perturbed samples around the neighborhood of the data point of interest, then uses a linear model to classify the prediction of new samples.

III. Results

For LIME implementation, I customized a LIME pipeline to fit our model architecture. In beginning, the pipeline always returns size errors. In order to fix this problem, I create a tokenizer encoder for the LIME pipeline, which adds special tokens, split into words, and with a max_length of 200. But I got other errors that indicate there is some difference in input and output expectation of the way of our model processes, it might be because of the one-hot encoding preprocessing we used for our model. In the end, we decided to use integrated gradients for model interpretation

For the fine-tuning process, we compared the result from two models and the table below shows the result. The table below shows the training and validation accuracy and loss of the training process with 20 epochs. As we discussed before, the LSTM head model is overfitting, which makes MLP a better decision for our project.

	MLP Head	RNN Head
train_loss	*0.34*	*0.10*
train_accuracy	*0.77*	*0.94*
val_loss	*0.31*	*0.41*
val_accuracy	*0.78*	*0.81*

Classification Metric Results

Classes/Metrics	**Precision**	**Recall**	**F1-score**
Negative	0.84	0.90	0.87
Positive	0.71	0.73	0.72
Neutral	0.65	0.49	0.56

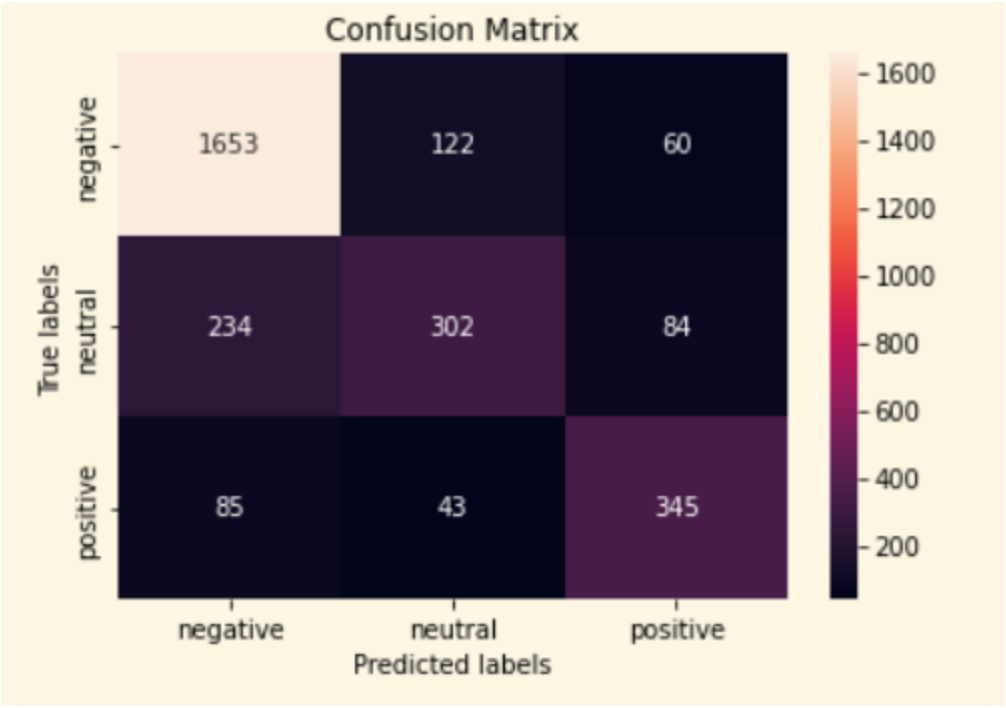

Figure: Confusion Matrix

For the post-hoc analysis, the figure and table above show some insights into the result. There are 1835 actual labels negative, and the model predicted 1972 negative labels. The model correctly predicts 1653 negative labels from 1972 predicted negative labels and correctly classifies 1653 negative labels from 1835 actual labels. For neutral classification, there are 620 actual labels, and the model makes 489 neutral predictions. The model correctly classifies 302 labels from 620 actual labels and 489 predicted labels. For positive classification, the model correctly classifies 345 labels from 473 actual labels and 467 predicted labels. From the table and matrix, we can conclude that the model is pretty good at predicting negative sentiment, and performance is not very good at predicting neutral sentiment. And neural network tweets were more likely to be predicted as negative tweets. According to EDA, the number of negative tweets is way larger than the other two, and neutral tweets have the least number of tweets. This very imbalanced data and not enough data might somehow affect the model to learn the context of sentences and prediction.

Legend: Negative Neutral Positive

True Label	Predicted Label	Attribution Label	Attribution Score	Word Importance
negative	negative (0.81)	0	0.35	[CLS] americana ##ir th ##x for losing my bag how hard is it to care for a bag w priority on it ? why do u con ##t to not care for ep s ? [SEP]
negative	negative (0.92)	0	0.69	[CLS] americana ##ir i even went to ticket counter and got no help [SEP]
positive	positive (0.86)	2	2.88	[CLS] jet ##bl ##ue good to hear th ##x for being responsive [SEP]
positive	positive (0.67)	2	3.35	[CLS] usa ##ir ##ways customer service at its best ! rachel s took great care of us at the ph ##x airport http / / t co / h ##g ##7 ##ve ##q ##hg ##hy [SEP]
neutral	negative (0.79)	0	0.03	[CLS] jet ##bl ##ue 2 55 tomorrow from ric to bo ##s looking good or am i better res ##ched ##ulin ##g ? [SEP]
negative	negative (0.88)	0	0.75	[CLS] southwest ##air really ? all other carriers are staffed and you ve got a triple loop ##ed one and no employees in sight in ok ##c [SEP]
negative	neutral (0.71)	1	2.53	[CLS] southwest ##air have you considered adding the we ll call you back when we have someone free feature to your support line ? [SEP]
negative	negative (0.88)	0	0.72	[CLS] southwest ##air is your b ##wi s ##j ##d service seasonal ? wasn t part of extension called int ##l desk they didn t know want to fly in sept on sat [SEP]
negative	negative (0.92)	0	1.02	[CLS] americana ##ir it is now going to be reported to the police due to the sexual ass ##ult sad that you didn t care [SEP]
neutral	neutral (0.67)	1	2.01	[CLS] southwest ##air vin ##dict ##ive t ##k larry david works for southwest ? [SEP]

For the model interpretability, the figure above shows we use ten samples to analyze the relationship between attributions of tokens assigned by the model, and predictions generated by the model. It shows that sometimes the model had the right attributions and made the right prediction like it gave word ‘good’ as positive attribution and made positive sentiment for the third sample.
But it also made the wrong decisions on assigning word attributions and wrong predictions, or assigned the wrong attribution to words and made the right decision. For example for the first sample, the model assigned ‘not ’ as positive attribution and predicted the sample correctly as negative sentiment.

IV. Summary and conclusion

In conclusion, this project is mainly on performing text classification tasks on twitter airline sentiment data using BERT based with the MLP head model. We fine-tuned the model by adjusting some hyper-parameters like batch size, max length, epoch, and learning rate. By doing the training and post hoc analysis, we know the model accuracy is 79%, with better performance in predicting negative sentiment. By implementing the model interpretation process, we are able to know a bit more clearly how the model makes a decision on the classification of this data, by assigning and calculating these attributions to tokens. And we can see the model is not always make the right decision and require more room for improvement.

In this project, while implementing all the work this project required, I feel I have gained a deeper understanding and am able to connect what I have learned from class, also it provides a great chance for me to do exploration and application. While searching on LIME, I am getting to know more about model interpretation, and how it works to interpret models in NLP context. And I successfully applied LIME interpreter to make a linear model interpretation for my other class’s project. Also, I have gained depth in the model architecture and tokenizer encoder structure.

For the improvement of this project, If we have more time I would like to focus on three parts, the first is to do some improvements to the model itself, and the second is to improve the data preprocessing. As mentioned earlier, we removed the emotion and emoticon which would likely be used to convey emotions. This might affect the model's performance. Also, we didn’t remove hyperlinks which might affect the performance as well. The last thing I would like to make improvements to is model interpretation. I would like to apply different model interpretation methods such as LIME, to the model and make a comparison between the two interpretation methods. From the comparison, we might find out different interpretations of the new method. Also, while I am searching for model interpretation, I came across some questions other people have about model interpreters randomly assigning attribution to words. While taking look at the result from our model interpretation, the same question was raised in my mind, because it assigned words with the wrong attributes, and assigned attributes to meaningless text like ‘http:/’.It might be wrong, but I believe there are methods that we could try to test the result.

V. Code Calculation

Code_from_internet.py : 207 lines
Code_edited.py :160 lines
Code_added.py: 80 Lines
Code from inter nett : (207-160)/(207+80)*100 = 16.37

VI. References

https://www.kaggle.com/datasets/crowdflower/twitter-airline-sentiment?datasetId=17&sortBy=voteCount
https://medium.com/aiguys/bert-explainability-5b54cff01407
https://captum.ai/tutorials/Bert_SQUAD_Interpret
https://github.com/bentrevett/pytorch-sentiment-analysis/issues/87
https://github.com/pytorch/captum/blob/master/captum/attr/_utils/visualization.py
https://www.theaidream.com/post/google-bert-understanding-the-architecture
https://www.oreilly.com/content/introduction-to-local-interpretable-model-agnostic-explanations-lime/
https://coderzcolumn.com/tutorials/machine-learning/how-to-use-lime-to-understand-sklearn-models-predictions#lime_tabular_ex1