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), RNN 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 help 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 preocessing, I worked on split contradiction words and removing stopwrods. The contradiction word might affect the progress of tokenization, therefore, we decided to split them into their subwords, make them can be recognized by tokenizer as other words. I also tried to remove stopwords from the text at begining, and compared the training and validation accuracy and loss from the training with 5 epochs. There is no big difference between with stopwords and without stopwords, since we decided use BERT 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 using informal language, it makes preprocessing harder, for example, how to deal with emoji and emoticon, and text language like ‘BTW’,’Ty’, 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 suit for classifier that takes as input a list of strings and outputs a 2d array of prediction probabilities. Lime creating some perturbed samples around the neighbourhood of data point of interest, then use linear model to classify the prediction of new samples.

III. Results

For LIME implementation, I custimized a LIME pipeline to fit our model architecture. At beginning, the pipeline always return size errors. In order to fix this problem, I create a tokennizer encoder for LIME pipeline, which adding special tookens, split into words and with max_length of 200. But I got other error which indicate there are some difference in input and output expectation of the way of our model process, it might because 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 model and the table below shows the result. The table below shows the training and validation accuracy and loss of training process with 20 epochs. As we discussed before, the LSTM head model is overfitting, which make MLP is 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

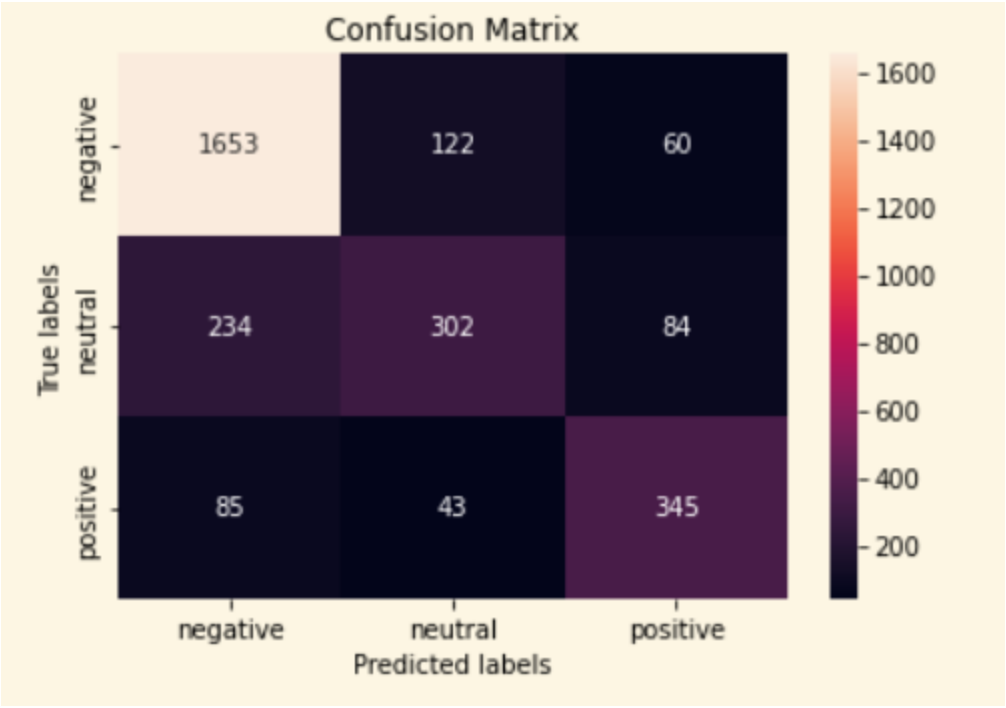

Figure: Confusion Matrix

For the post-hoc analysis, figure and table above shows some insights of 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 not very good on predicting neutral sentiment. And neural network tweets were more likely be predicted as negative tweets. According to EDA, the number of negative tweets is way larger than the other two, and neutral tweets has the least number of tweets. This very imlanced data, and not enough of data might somehow affect the model to leran the context of sentence 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 sample to analysis the attributions of tokens which assigned by model. It shows that sometime the model had the right attributions and made right prediction, like it gave word ‘good’ as positive attribution and made positive sentiment for the third sample.
But it also made wrong decision on assigning word attributions and wrong predictions, or assigned wrong attribution to words and made 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 txt classification task on twitter airline sentiment data using BERT based with MLP head model. We fine-tuned the model by adjusting some of 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 on predicting negative sentiment. By implement model interpretation process, we are able to know a bit more clear on how does model make decision on classification of this data, by assigning and caculate these attributions to words. And we can see the model is not always made 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. Also, I have gained depth in the model architechture and tokenizer encoder structure.

For the improvement of this project, If we have more time I would ike to focus on three parts, first is to do some improvement on model itself, and second is to improve the data preprocessing. The data we used is informal language, including slangs, text language, emoji/emoticon and other words that not in dictionary. For the improvement, should make a pipeline to processing these problems. The last thing I would like to improved is model interpretation. I would like to apply different model interpretation method such as LIME, on the model and make comparison between two interpretation methods. From the comparison, we might find out different interpretation from new method. Also, while I am searching for model interpretation, I came across some questions other people has about model interpreter randomly assign attribution to words. While taking look at the result from our model interpretation, same question raised in my mind, because it assigned words with wrong attribute, and assign attribute to meaningless text like ‘http:/’.It might be wrong, but I believe there are methods that we could try to test on 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://www.geeksforgeeks.org/fine-tuning-bert-model-for-sentiment-analysis/
https://stackoverflow.com/questions/73911673/pytorch-dataloader-with-huggingface-transformer-getting-error-unable-to-creat
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
https://www.researchgate.net/publication/337510909_Improving_BERT-Based_Text_Classification_With_Auxiliary_Sentence_and_Domain_Knowledge