Individual Report for Airline Tweet Sentiment Analysis

Pavani Samala

1. Introduction

In this project, we fined-tuned a BERT model and used Integrated Gradients to interpret why our model was inclined to make the predictions it did. First, we performed EDA to understand the data we were working with and preprocessed it by removing special characters and punctuation and expanding contractions. Before fine-tuning, we compared the results of the BERT model being run with a MLP head as well as a RNN head and saw better results for MLP. Hence, we fined-tuned a BERT-MLP and experimented with the number of epochs, learning rate, and freezing and unfreezing layers. Finally, we conducted post-hoc analysis and used layer gradients to check which words were positively or negatively impacting the model's decision.

2. Background and Description of Individual work

For my contribution, I worked on EDA and interpretability. The EDA portion was meant to give us insight into how our data was distributed and show us any interesting trends that we could make sense of. I created plots to determine if the dataset was imbalanced and see how the sentiments were distributed when they weren't mapped to the classes. I also looked at the polarity of tweets per airline to see if one airline had more of a certain sentiment. Lastly, I looked at the length of the tweet with respect to the sentiments.

SHAP and LIME were two tools that I tried to implement to understand our model's predictions (Note: at the time of implementing SHAP and LIME, we planned on using an LSTM head, but in the end, we chose MLP). SHAP begins by using prior knowledge as a base value and adding the present data's features to understand its impact. It calculates SHAP values for each feature, telling us how much each feature impacts the model's decision. LIME works by generating new data points around a given instance, then uses the model to classify the new data points and fits it using a linear model. Unfortunately, SHAP is not compatible with LSTM models and LIME was not useable with the one-hot encoding preprocessing. Therefore, we moved on to implementing Integrated Gradients.

Much of the other work I did was analyzing post-hoc results and determining which tweets the model excelled in classifying and which tweets it had trouble classifying.

4. Results

As mentioned before, prior to the fine-tuning phase, we needed to decide between using an RNN or MLP head in conjecture to the BERT model. The table below shows the BERT model running using a RNN and MLP head. It is evident that the RNN head is causes the model to overfit, making MLP a better choice.

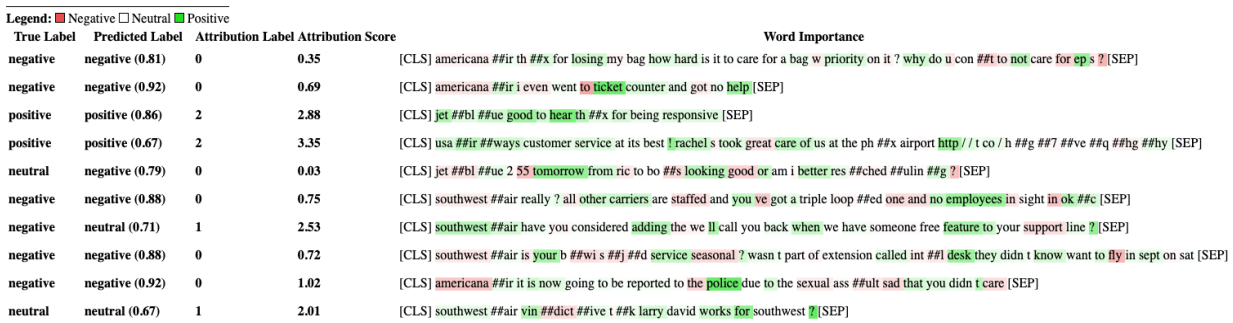
	MLP Head	RNN Head
train_loss	0.3430536785354353	0.10062554912412003

train_accuracy	0.7700128424657534	0.9436001712328768
val_loss	0.31268693341149223	0.4129580193095737
val_accuracy	0.7860243055555556	0.8116319444444444

After running our experiment, we calculated precision, recall, and f1-score metrics to evaluate the performance of our model. The classification metrics are summarized in the tables below. We were able to see that the negative connotated tweets were easier for the model to classify compared to the positive and neutral connotated tweets. The f1-score also followed this trend and was highest for negative connotated tweets compared to positive and neutral connotated tweets. The overall accuracy of the model on the test data was 79%. This metric was backed by the training and validation accuracies which were 77% and 78%. This shows us that there are no signs of overfitting and we believe the model performed well overall.

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

In terms of interpretability, the figure below shows how Integrated Gradients attributed certain words and how they positively or negatively impact the model's prediction. There are times when the model correctly understood the context of a word such as "good" and "responsive" and predicted the tweet as positive sentiment. However, there were also times when it incorrectly understood the context. For example, words like "great" and "sad" did not impact the model's prediction of positive and negative sentiments, even if the tweet was correctly classified.



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]

5. Summary and conclusions. Summarize the results you obtained, explain what you have learned, and suggest improvements that could be made in the future.

In conclusion, the BERT model combined with an MLP head worked well with a 79% test accuracy. The model classified more negative sedimented tweets correctly compared to positive and neutral sedimented tweets. After analyzing the results from layer gradients, we could see that the model was correctly understanding the context of particular words when making a prediction but failed to do so consistently.

6. Code

EDA: I took 133 lines of code, modified 5, and added 6, so (133-5/133+6) x100 = 92%

SHAP: I took 7 lines of code, modified 3, and added 0, so (7-3/7+0) x100 = 57%

LIME: I took 8 lines of code, modified 5, and added 2, so (8-5/8+0) x100 = 37.5%

7. References

https://www.yourdatateacher.com/2021/05/17/how-to-explain-neural-networks-using-shap/

https://captum.ai/tutorials/Image_and_Text_Classification_LIME