BERT-Based Sentiment Analysis on X Data: Understanding Social Media Discourse

Jakeb Milburn

Abstract

This study presents a BERT-based approach for sentiment analysis on a curated dataset from X (formerly known as Twitter) comprising approximately 70,000 tweets, each labeled with an entity and one of three sentiment categories: positive, negative, or neutral. Leveraging the bert-base-cased model, we fine-tune it using transfer learning techniques to capture nuanced sentiment expressions in tweets. The dataset is preprocessed to ensure compatibility with BERT's input requirements. Our methodology involves encoding messages and their associated labels to predict sentiment classes accurately.

Experimental results demonstrate the effectiveness of our approach in achieving high accuracy in classifying sentiments across the three defined categories. This streamlined sentiment analysis framework provides valuable insights into public sentiment on social media, aiding decision-making processes and enhancing user experience.

Problem Statement

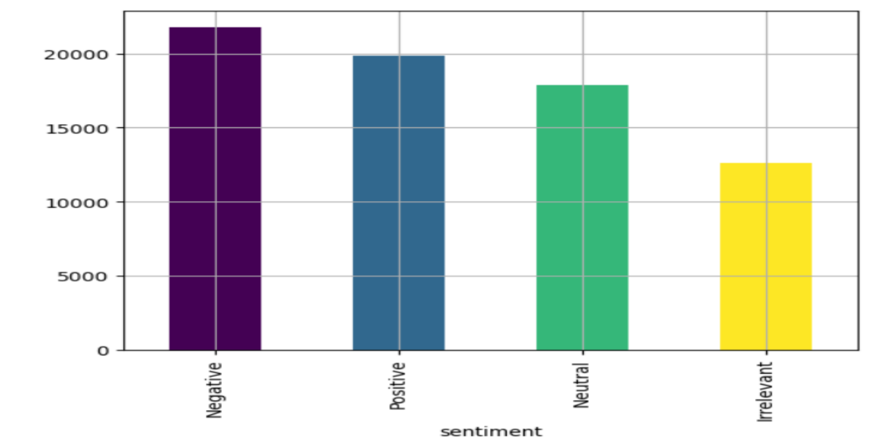
Despite the widespread use of social media platforms like X for expressing opinions, emotions, and sentiments, analyzing the vast amount of user-generated content poses significant challenges. Sentiment data can contain information useful in determining public opinion towards any topic or entity, especially when it comes from social media outlets that adapt to real-world events as they occur. However, manually labeling data generated constantly every day is extremely expensive and time-consuming. This study seeks to mitigate that cost by creating a framework that allows users to automatically classify social media sentiments. This opens the door for efficient real-world applications such as:

- Aided decision making for companies utilizing social media for advertising.
- Generating public opinion reports on political events.
- Analyzing trends in popular culture.

Methodology

Dataset Pre-processing

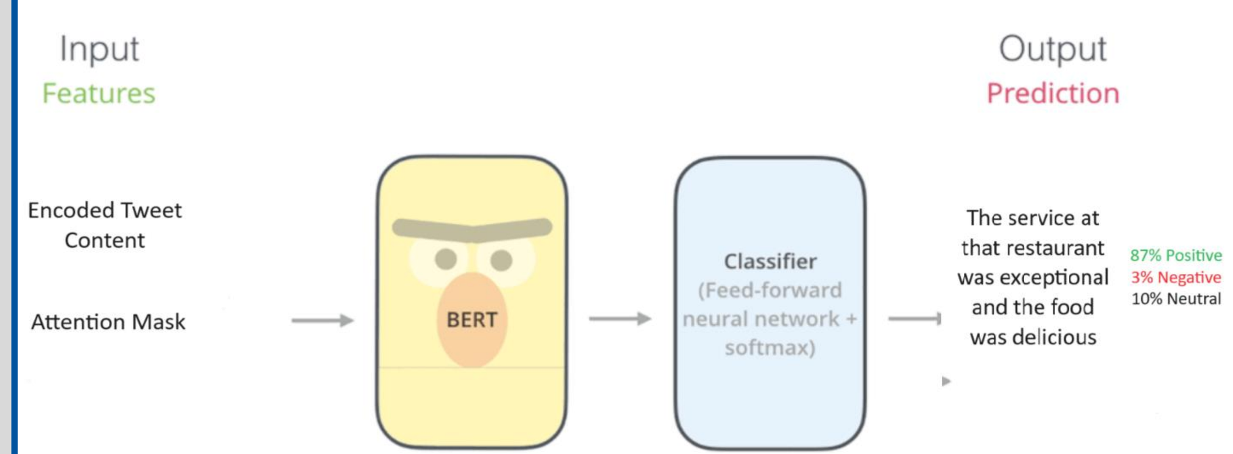
Preprocessing that data was required to achieve high accuracy with BERT. Each tweet originally had four labels: “Positive, Negative, Neutral, and Irrelevant”. “Irrelevant” and “Neutral” examples were indistinguishable without context regarding the topic of the tweet, so “Irrelevant” examples were relabeled as “Neutral”. The distribution of the data before reclass:



Numbers, special characters, and stopwords were then removed from each tweet and labels were converted to integers corresponding to classes. Data was split into 90% train, 5% test, and 5% validation. Encodings and attention masks were generated using the “encode_plus” method from the bert-base-cased tokenizer, which automatically handles adding [CLS], [SEP], and [PAD] tokens. We finally utilize a dataset class combined with a dataloader to return the tweet content, tweet encoding, attention mask, and label target tensors.

Fine Tuning Pipeline

The model architecture is a simple feed-forward network built on top of BERT like so:

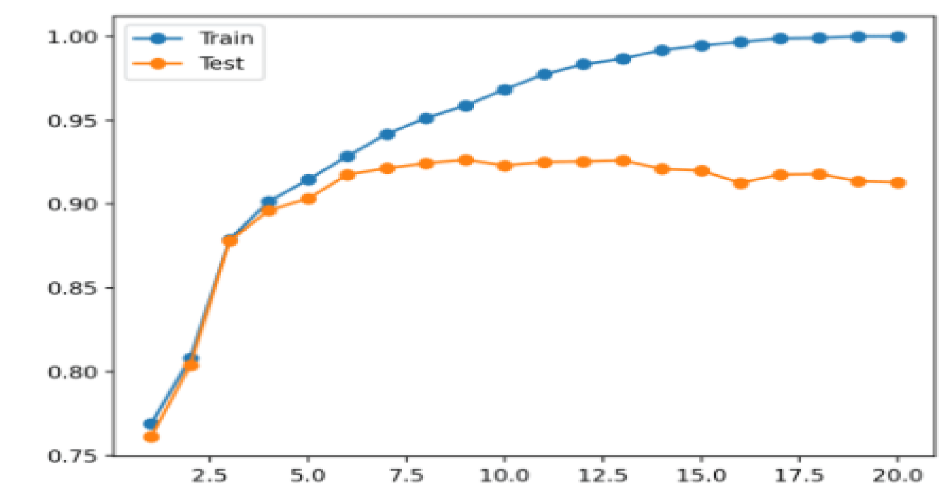


We initialize the epochs, initial learn rate, Adam optimizer, scheduler, and loss function. The inputs are fed to BERT to utilize the pretrained embeddings, A dropout with a p of .3 is applied for regularization and avoiding overfitting, then a linear transformation is applied to change the hidden size of the BERT model to our number of classes to get the output.
After the output is received, we apply Cross-Entropy Loss to the output and the target label to obtain the loss, then perform back-propagation with the loss and step with our optimizer and scheduler. The final predication can be acquired by taking the argmax of the output.

Results

Accuracy

With the initial learn rate set to 0.00002, batch size of 16, number of epochs at 20, and a linear learn rate scheduler, we were able to achieve an accuracy of 92.5% on the test set.



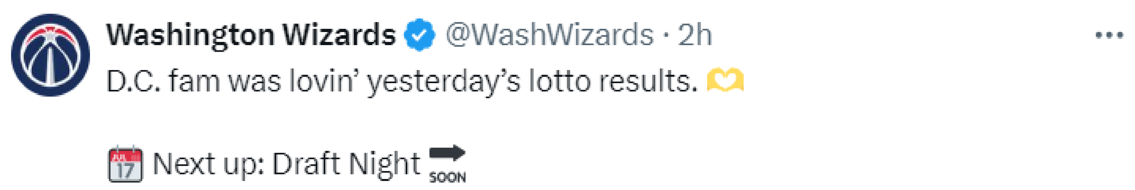
Other hyperparameters yielded good results, but we found this configuration to be the best. Accuracy was consistently over 90% regardless of what parameters were used.

The largest boost in accuracy occurred when we relabeled the “Irrelevant” category to be included in the “Neutral” category. The greatest accuracy for that run was only 88%, suggesting that the model has trouble differentiating between those categories, and shows almost a 5% increase in performance when they are combined.

An experiment done on the same dataset using a Random Forest Classifier⁰ showed an accuracy of 90.5%, compared to our accuracy of 92.5%, suggesting that leveraging BERT embeddings leads to an increase in performance.

Inference on Modern Tweets

Our data is from 2021. Without access to an up-to-date dataset, we can only make predictions on smaller amounts of data from recent times. Still, the model seems to generalize well on newer data.



Prediction: “Positive”

Prediction: “Negative’

Conclusions

Leveraging BERT fine tuning for X sentiment analysis proved extremely effective, boasting high accuracy in distinguishing between the three sentiment classes.

Limitations

- Language is always evolving, so words that are used in social media today might not have been around when the model was trained, leading to lower accuracy over time if the model is not up to date.
- Combining the “Irrelevant” and “Neutral” classes led to a class imbalance, potentially impacting results

Direction for Future Research

- Creation of a dataset with more sentiment labels than the three used could lead to more precise classifications when used to train the model.
- Find a way to convert emojis into text representing emotion for use in making model predictions.

References

- Speech and Language Processing
https://web.stanford.edu/~jurafsky/slp3/
- Twitter Dataset
https://www.kaggle.com/datasets/jp797498e/twitter-entity-sentiment-analysis/data
- BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding
https://arxiv.org/abs/1810.04805
- Sentiment Analysis with BERT using Huggingface
https://medium.com/@manjindersingh_10145/sentiment-analysis-with-bert-using-huggingface-88e99deeec9a
- Random Forest Sentiment Analysis⁰
https://www.kaggle.com/code/omaradel1221/nlp-sentiment-analysis