DeepSentiment: Tweet Sentiment Analysis using BERT & LLMs

Authors:

- Dhyey Mavani*, Amherst College '25*
- Carlsson May*, Williams College '25*

Professor & Project Advisor:

- Bálint Gyires-Tóth, *AIT Budapest & NVIDIA*

Abstract:

This project, DeepSentiment, leverages advanced deep learning techniques, particularly BERT (Bidirectional Encoder Representations from Transformers) and LLMs (Large Language Models), to analyze and predict sentiment from tweets. The primary aim is to perform sentiment analysis using BERT-based models, and observe the effectiveness of other contemporary LLM-based techniques such as few-shot learning in classification as well. The insights garnered from tweet sentiment patterns can enhance understanding and implementation of state-of-the-art deep learning techniques in diverse sentiment analysis applications such as financial markets, and behavioral economics.

1. Introduction

The exponential growth of social media platforms like Twitter has resulted in vast amounts of unstructured text data that encapsulate public opinions on various topics. Analyzing such data, especially determining the sentiment of texts, plays a crucial role in market analysis, political campaigns, public relations strategies, and more. Traditional sentiment analysis methods struggle with the nuances and informality present in tweets. Thus, this project explores the application of the BERT-based models, renowned for its contextual understanding capabilities, in interpreting and predicting sentiments expressed in tweets. Additionally, we aim to compare the accuracy of the fine-tuned version of the BERT-base model against other methods such as Naive Bayes, RoBERTa-base and few-shot LLM-based sentiment classification.

Upon researching, we saw that the previous solutions to the above-mentioned problem of our interest use models like BERTweet[2], RoBERTa[3], and Conditional Generative Adversarial Network (CGAN)[4].

2. Methodology

2.1 Data Collection and Preprocessing

Our dataset comprises tweets extracted from the TweetEval[1] benchmark, specifically focusing on the sentiment subset. The initial step involves preprocessing the raw text data to format suitable for training deep learning models. This includes removing noise like hashtags, URLs, and user mentions, standardizing text by correcting typos and expanding contractions, and tokenizing text into manageable units for model processing. The preprocessing scripts and operations are detailed in our GitHub repository at github.com/DhyeyMavani2003/DeepSentiment/tree/main under the `./code` directory in a file named `preprocess_tweeteval.py`.

Each `{train/val/test}_text.txt` file has one tweet per line in the original format, i.e., no preprocessing has been applied. Each `{train/val/test}_labels.txt` file has one label per line which maps to its corresponding tweet. The `mapping.txt` files contain tab-separated lines of the form

`label_id <tab> label_name`, for instance, in our case of sentiment detection `0 <tab> negative`, `1 <tab> neutral` and `2 <tab> positive`

The tweets in the file `pre-token-{train/val/test}.csv` are version of the same in `{train/val/test}_text.txt` data, pre-processed (using techniques such as stemming and contraction expansion) to be better suited to the NLP task (after model-specific tokenization, which will be applied later in the model-pipeline).

2.2 Model Training and Evaluation

2.2.1 BERT and RoBERTa Fine-tuning

The core of our analysis involves fine-tuning the pre-trained BERT-base and RoBERTa-base models tailored to our sentiment classification needs. Fine-tuning adjusts the models' pre-trained weights to better accommodate the specificities of sentiment analysis in tweets. These models are assessed on metrics such as accuracy, precision, recall, and F1-score to determine their effectiveness.

2.2.2 Comparative Analysis with Other Models

In addition to our BERT-based models, we use Naive Bayes and an LLM model (using Microsoft's Phi-3-Mini-128k-instruct[5] containing ~4 billion parameters) with few-shot prompt engineering to gauge their comparative performance in the sentiment classification task. This lended us further insights into the potential of other methods in approaching the task of classifying the sentiment of tweets.

2.3 Implementation

We utilize Python along with libraries such as `Numpy, Pandas, Transformers, Tensorflow, Torch, NLTK, Scikit-Learn, Matplotlib, Seaborn, LangChain, Auto-GPTq, Accelerate, Huggingface_Hub, URLLib,OS` and GitHub Copilot for implementing our entire codebase from data preprocessing to model training and testing. Our codebase, accessible via GitHub, includes detailed scripts, organized data, Jupyter Notebooks, and instructions for reproducibility that outline the model training, validation, and testing phases along with documentation including the final presentation and this paper itself as well.

3. Results and Discussion

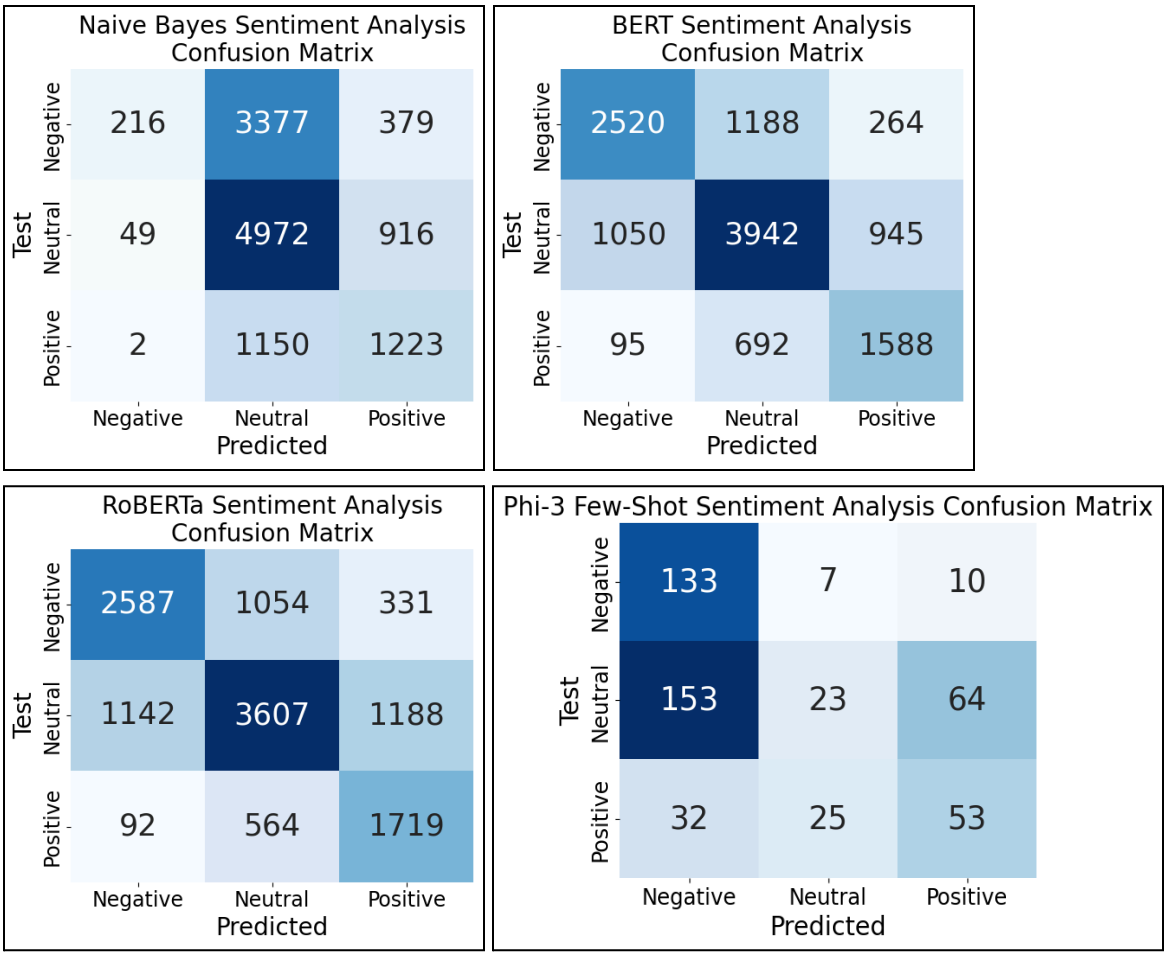
Our preliminary results indicate that the fine-tuned BERT model outperforms traditional models significantly in capturing the contextual nuances of tweet sentiment. The BERT model exhibits superior understanding of complex sentence structures and sarcasm, which are prevalent in social media texts. Furthermore, when compared to other advanced models like RoBERTa, BERT shows competitive performance, often excelling in precision and recall metrics.

In training these models, we were limited by computational resources – the size of our training set was significant, and access to GPU/TPU runtimes and RAM through Google Colaboratory was limited. Due to these limitations, we were only able to train the RoBERTa and BERT models for 1 and 2 epochs, respectively. We speculate that the RoBERTa model could be made more effective with greater training time. Additionally, testing on Phi-3-Mini was done using a small subset of our test set, using just 500 tweets.

Model	**Sentiment Classification Accuracy on the Test set**
BERT	0.655
RoBERTa	0.644
Naive Bayes	0.52
Phi-3-Mini (LLM)	0.418

Comparative Metrics for Sentiment Classification on the Test Set: BERT / RoBERTa			
	Precision	Recall	F1
Negative	0.69 / 0.59	0.61 / 0.80	0.65 / 0.68
Neutral	0.65 / 0.71	0.73 / 0.55	0.69 / 0.62
Positive	0.64 / 0.63	0.57 / 0.60	0.61 / 0.61

The following confusion matrices help us further visualize the performance of various methods by aiding our ability to compare them for misclassification rates and the like:



4. Conclusion and Future Work

Our work demonstrates the efficacy of using BERT-based models for tweet sentiment analysis, specifically the `bert-base-uncased` model. The project not only enhances our insights into usage of various models for sentiment analysis tasks but also opens avenues for real-time analysis applications. Future expansions could include using GANs, bigger LLMs / more training time, or exploring the comparative efficacy of zero/one/few-shot LLM prompting. Future work could also explore the integration of multilingual models to cater to diverse demographics on Twitter by having multiple agents trained to specialize in classification of a subset of tasks with one orchestrator, which classifies the tasks into buckets based on the other agents' specialities. This might improve the predictions, and this is a great theory to test as part of the future work!

5. References

[1] TweetEval Benchmark and Associated Papers: https://github.com/cardiffnlp/tweeteval
[2] BERTweet Paper: https://paperswithcode.com/paper/bertweet-a-pre-trained-language-model-for
[3] Hugging Face's FacebookAI/RoBERTa-base Model: https://huggingface.co/FacebookAI/roberta-base
[4] V. Mahalakshmi, P. Shenbagavalli, S. Raguvaran, V. Rajakumareswaran, E. Sivaraman, Twitter sentiment analysis using conditional generative adversarial network, International Journal of Cognitive Computing in Engineering, Volume 5, 2024, Pages 161-169, ISSN 2666-3074, https://doi.org/10.1016/j.ijcce.2024.03.002.

[5] Hugging Face’s Microsoft/Phi-3-mini-128k-instruct :
https://huggingface.co/microsoft/Phi-3-mini-128k-instruct