Evaluating BERT-Based Sentiment Analysis Across Different Classifcation Methods and Data Types

Bijan Eshghi
University of Lausanne
bijan.eshghi@unil.ch

***Abstract*- The proliferation of user-generated content on social media platforms and review websites has allowed for the development of robust sentiment analysis models to interpret public opinion and consumer feedback. This paper presents a comprehensive approach to sentiment analysis by using the Bidirectional Encoder Representations from Transformers (BERT), a state-of-the-art natural language processing model, as a foundation to inquire on efficient model specifications across different data types and sizes. The methodology involves fine-tuning pre-trained BERT models with different classification methods on diverse datasets comprising social media posts and a variety of reviews. The classification models presented are a Neural Network, XGBoost and the Random Forest.**

Index Terms - Natural Language Processing, Bidirectional Encoder Representations from Transforms, Twitter, IMDB, Reddit, Yelp, XGBoost, Neural Network, Random Forest.

I. Introduction

Natural Language Processing (NLP) has seen remarkable advancements in recent years, significantly driven by the development of sophisticated machine learning models. Among these, the Bidirectional Encoder Representations from Transforms (BERT) model, introduced by Google in 2019, represents a substantial leap forward (Chang, Lee, Toutanova 2019). BERT's innovative architecture, grounded in the Transformer model, enables it to understand the context of words in a sentence by examining them in both directions- forward and backward- simultaneously. This bidirectional approach contrasts sharply with previous models, which typically processed text in a unidirectional manner.

BERT's introduction has revolutionized various NLP tasks, including question answering, sentiment analysis, and named entity recognition (to name a few), by providing a pre-trained model that can be fine-tuned for specific tasks with minimal data and computational resources. Its ability to capture nuances of human language, such as polysemy and syntactic ambiguity, has set new benchmarks in NLP performance (Kacprzak 2024). This paper is interested in the evaluation of BERT's performance across different classifiers and data types.

.

II. Research Question

This paper delves into the application of the BERT model in the context of sentiment analysis, specifically by evaluating its performance across four datasets, each comprising a different online media platform, and using three different classification methods to answer the following research question: *In the context of BERT, does a single classification method consistently perform best across various data types, or do different classification methods exhibit superior performance with specific data types?*

The exploration of this question is vital for understanding the robustness and adaptability of BERT embeddings when applied to different types of classification tasks. While BERT excels at capturing semantic nuances in text data, the choice of classification method used to leverage these embeddings can greatly influence the overall performance of the model. Thus, identifying the most effective classification strategies is essential for maximizing BERT's potential across diverse applications

As an additional dimension of interest, it would be valuable to see how varying the size of the datasets will impact the performance of the models as it can provide some indications on scalability.

.

III. Data description

When searching for labeled datasets pertaining to sentiment analysis, there appear to be two prevailing data types for which quality datasets are available: social media posts and reviews.
This is likely due to the intrinsic subjective content, structured format, public accessibility and abundance of data that can be found on the relevant platforms where these data types can be found. As such, to answer the research question of this paper, four different datasets are used. The first, found on the platform HuggingFace, is the vastly popular *yelp_reviews_full* dataset from 2015, which consists of 700K Yelp reviews labeled with a star rating ranging from 1 to 5. The other three datasets can all be found on the data science platform Kaggle. The *Emotions* data set is the largest of these, consisting of close to 400K tweets, all meticulously labeled with one of six emotions: sadness, joy, love, anger, fear, and surprise. The *IMDB Dataset of 50K Movie Reviews* is also included, each review labeled as either positive or negative. Finally, the *Twitter and Reddit Sentimental Analysis Dataset* is used to obtain a dataset for Reddit comments, all labeled as either positive, neutral, or negative, totalling . With two datasets for each data type, social media posts and reviews, we are able to control for possible platform specific classification accuracy, at least to a certain extent. Ideally, one would like to obtain more datasets from different platforms to control this further, however for the purposes of this paper, four datasets will be

sufficient to compare the performance of our classifiers across reviews and social media posts.

In terms of dataset size, it is apparent that datasets containing hundreds of thousands of observations (Yelp & Twitter) are unnecessarily large since BERT is a pre-trained model that doesn't need that much data to be fine tuned. At the same time, the Reddit dataset contains a total of 36801 comments, and after dropping all non-english comments we are left with 31634 comments. After balancing the dataset by undersampling, we are left with 23556 comments and as such 7852 comments for each label. In order to make comparisons fair across platforms, all datasets are reduced to have a maximum of 7852 observations per label and are balanced. From here, in order to explore varying dataset sizes, we further shrink the datasets to medium (3500/label) and small (500/label) sizes. While it was attempted to make these datasets as comparable as possible, there are still some issues, such as the different lengths of text between datasets. To take the two most extreme cases as an example, after dropping non-english texts and balancing, the Twitter dataset had an average text length of 98 words while the IMDB dataset's average length was 1309 words. Finally, for all datasets, we conduct a train-validation-test split as 80/10/10. This split was chosen instead of a 60/20/20 as it was thought that this would reduce the size of training data by too much.

IV. Methodology

The classification methods chosen for this paper are a Neural Network, Random Forest and XGBoost as they are all powerful classification methods that are not overly complicated to understand or to implement. Furthermore, Random Forest is a strong baseline model upon which we can compare the Neural Network classification and XGBoost. The XGBoost model is renowned for its performance in structured/tabular data and excels at optimizing loss function through boosting, while Neural Networks can model complex patterns and interactions in data and are especially effective in capturing intricate features learned by BERT. With each of these models having their own advantages we can explore performance across the two different data types.

V. Code Implementation

A. Downloading Data

As previously mentioned, the first step was to download all datasets from Kaggle, except for yelp_reviews_full which could be found on HuggingFace. Once this was completed, a simple script was written (cleaning.py) which contained a function (convert) that would take a file location and name as input and would convert it to a pandas dataframe as long as it was a .csv, .tsv or .parquet file.

B. Data Cleaning

Once all datasets were converted to pandas dataframes, the cleaning process began for each dataset separately. The first step was to drop all non-essential columns from the dataset, keeping the text and label column (and naming them as such if necessary). Next, any string labels were turned into integers to allow for future processing. For the Yelp and Reddit datasets, it was necessary to import a package which allowed for language detection that permitted for all non-english texts to get dropped from the dataframes. Finally, all data frames were balanced by undersampling and some final summary statistics were measured. The cleaning process was completed once all cleaned data frames were exported as a full version containing the full cleaned dataframes, large, medium and small as mentioned in section III in a .csv format.

C. Model Implementation

Neural Network -

The first classification method that was implemented was a BERT-based Neural Network using PyTorch. The workflow starts by taking the pandas dataframes and splitting them into training, validation and test sets. The BERT tokenizer was used to pre-process the text data, transforming it into token IDs and attention masks suitable for input into the BERT model. The core of this script is a custom BertClassifier class, which fine-tunes a pre-trained BERT model and uses a feed-forward Neural Network for classification. Training involves setting up a data loader, defining a loss function, and optimizing using AdamW with a learning rate scheduler. Next, the model's performance is evaluated on validation data after each epoch, using metrics such as loss, accuracy and confusion matrices, and the trained model is saved using the dill package for future use in the Advanced Programming project. Finally, a function was included to test the model on unseen test data, generating a classification report and confusion matrix for comprehensive performance evaluation. Almost all computation was performed on the GPU. The design of the Neural Network and much of the programming was inspired and taken from Tran's 2019 website. It is worth noting that the Neural Network parameters used in this project are the parameters that come built-in with the BERT classifier.

XGBoost -

In this section, a text classification pipeline is developed using BERT embeddings and an XGBoost classifier. The process begins with splitting the dataframe into training, validation and test sets. The text data is then preprocessed and tokenized using a pre-trained BERT tokenizer, and embeddings are generated for each text using a BERT model, transforming the texas into fixed-size vectors, depending on the number of labels, an XGBoost classifier is initialized with appropriate parameters for binary or multi-class classification. Hyperparameters are tuned via GridSearchCV. The model is trained on the training set, evaluated on the validation set, and, if specified, tested on the test set. Performance metrics are the same as those in the model mentioned above and the trained model was saved using pickle. Almost lal the computation was performed on the GPU. The design of the XGBoost and much of the programming was inspired and taken from Gosh's 2024 online post.

Random Forest -

In this section, a text classification pipeline was developed using BERT embeddings and a Random Forest classifier. The process is almost identical as that for XGBoost, as such it won't be all re-iterated again. However, obviously the embeddings are used to train a Random Forest classifier in this case, and once again the hyperparameters are tuned via GridSearchCV. All computation was performed on the CPU as using a GPU for the task introduced complications. This however was not much of a hindrance on performance as Random Forest classification did not prove to be computationally demanding.

As a final remark to conclude this section of the paper, for all processes involving a random sampling of some sort, a seed was set to ensure replicability of all results.

VI. RESULTS

After having run all the training models and observing their performance, we are now able to evaluate the models across the three classifiers and four media platforms. The complete set of all results can be found on the GitHub repository. These include classification reports as well as confusion matrices. As such, this paper presents only the most essential findings in summary and will refer to the graphs by name. Firstly, uniform across all datasets, platforms, and sizes, the Neural Network performed best. While its high performance didn't come as a shock, considering that it is the built-in classifier for BERT, the uniformity across which it outperforms the other classifiers is remarkable, as there is not one instance where the other classifiers outperform the Neural Network. Between XGBoost and the Random Forest classifiers, the better all-round performing model is not quite as obvious. For the small dataset size, XGBoost performs better for IMDB and Twitter while they perform similarly for the other two datasets. Moving to medium sized datasets, XGBoost outperforms Random Forest for Twitter and Yelp while once again they perform roughly the same for the other two datasets. Finally, for the largest dataset size, XGboost outperforms Random Forest slightly for all datasets except for Twitter, where the difference is noticeable.

It is worth noting that of all datasets, the Yelp dataset was the hardest to predict by a large margin, even for the Neural Network. There are likely a few reasons for this, as star ratings are hard to predict even for humans (possibly leading to inconsistent labeling throughout the dataset), but also due to the long length of the reviews. While it is true that the BERT pads the length of all text to a maximum of 512 tokens, long texts are still difficult to analyze as there may be multiple emotions displayed throughout it. As such, it might prove difficult for the algorithm to label the entire corpus with one label.

.

VII. CONCLUSION

In this paper, we explored the efficacy of the BERT model in performing sentiment analysis across diverse datasets, including social media posts and review platforms. By fine-tuning pre-trained BERT models and employing three classification methods - Neural Network, XGBoost, and Random Forest - we aimed to determine whether a single classification approach consistently outperforms others across various data types or if performance varies based on the dataset's characteristics.

Our findings reveal that the Neural Network classifier consistently outperformed both XGBoost and Random Forest across all datasets and dataset sizes. This superiority is attributed to the Neural Network's capability to capture complex patterns and interactions within the data, which aligns well with BERT's sophisticated embeddings. Despite the varying nature of the datasets, the Neural Network's performance remained robust, highlighting its adaptability and effectiveness in leveraging BERT embeddings for sentiment analysis.

When considering XGBoost and Random Forest, the former generally performed better than the latter, particularly for larger datasets. However, the performance gap between these two classifiers was not as significant as that observed with the Neural Network, suggesting that while traditional machine learning models like XGBoost and Random Forest are viable options, they may not fully harness the depth of BERT's contextual embeddings compared to a Neural Network based approach

Additionally, our results indicated that the Yelp dataset posed the greatest challenge across all classification methods. This difficulty likely stems from the inherent variability and subjectivity in star ratings and the complexity of analyzing lengthy reviews

In conclusion, our comprehensive evaluation finds that the Neural Network exhibits superior performance in leveraging BERT embeddings for sentiment analysis across diverse datasets. The findings highlight the potential for further advancements in NLP by refining Neural Network architectures and fine-tuning strategies.

VIII. REFERENCES

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019, May 24). Bert: Pre-training of deep bidirectional Transformers for language understanding. arXiv.org. https://arxiv.org/abs/1810.04805

Kacprzak, K. (2024, January 26). Roberta vs. Bert: Exploring the evolution of Transformer models. Data Engineering, MlOps and Databricks services. https://dsstream.com/roberta-vs-bert-exploring-the-evolution-of-transformer-models/#:~:text=BERT's%20pre%2Dtraining%20involves%20two,language%20inference%2C%20and%20sentiment%20analysis.

IX. APPENDIX

ChatGPT was used to assist in writing this paper in a few sections. It was not, however, used to program anything anywhere in this project.